Distributed Intelligent Object Based Control System for Accelerators

By

RISHI PAL YADAV

(Enrollment No. : ENGG03200704001)

Raja Ramanna Centre for Advanced Technology, Indore

A thesis submitted to the

Board of Studies in Engineering Sciences

In partial fulfillment of requirements
For the Degree of

DOCTOR OF PHILOSOPHY

of

HOMI BHABHA NATIONAL INSTITUTE

April 2015

Homi Bhabha National Institute

Recommendations of the Viva Voce Committee

As members of the Viva Voce Committee, we certify that we have read the dissertation prepared by **Rishi Pal Yadav** entitled **"Distributed Intelligent Object Based Control System for Accelerators"** and recommend that it may be accepted as fulfilling the thesis requirement for the award of Degree of Doctor of Philosophy.

Date: April 24, 2015
Chairman – Prof. A. P. Tiwari

Date: April 24, 2015
Guide/ Convener – Prof. P. V. Varde

Date: April 24, 2015
Co-Guide – Prof. P. S.V. Nataraj

Date: April 24, 2015
Examiner – Dr. Ajith Kumar B.P.

Date: April 24, 2015
Member 1 – Prof. S. V. G. Ravindranath

Date: April 24, 2015
Member 2 – Prof. A. K. Bhattacharjee

Date: April 24, 2015
Technology Adviser – Mr. P. Fatnani

Final approval and acceptance of this thesis is contingent upon the candidate's submission of the final copies of the thesis to HBNI.

I hereby certify that I have read this thesis prepared under my direction and I recommend that it may be accepted as fulfilling the thesis requirement.

Date: April 24, 2015
Place: RRCAT. Indore.

P. S.V. Nataraj
Co-Guide

P. V. Varde
Guide

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at Homi Bhabha National Institute (HBNI) and is deposited in the Library to be made available to borrowers under rules of the HBNI.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the Competent Authority of HBNI when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

Rishi Pal Yadav

DECLARATION

I, hereby declare that the investigation presented in this thesis has been carried out by me. The work is original and has not been submitted earlier as a whole or in part for a degree/diploma at this or any other Institution / University.

Rishi Pal Yadav

List of Publications arising from the thesis

Journal

1. "Relay Feedback-Based Critical Parameter Estimation for First Order Plus Dead Time Type Plant in Networked Control System Configuration", R.P. Yadav, P.V. Varde, A. Chauhan, and P. Fatnani, *International Journal of Modelling, Simulation, and Scientific Computing*, **2011**, *Vol.2, No.3*, pp.375-391.

2. "A multi-agent based control scheme for accelerator pre-injector and transport line for enhancement of accelerator operations", R.P. Yadav, P. Fatnani, P.V. Varde, and P.S.V. Nataraj, *Online journal Elixir Comp. Sci. & Engg.*, **2012**, *Vol.44*, pp.7405-7410.

3. "Model-based Tracking for Agent-based Control Systems in the Case of Sensor Failures", R.P. Yadav, P.V. Varde, P.S.V. Nataraj, P. Fatnani, C.P. Navathe, *International Journal of Automation and Computing*, **2012**, *Vol.9, No.6*, pp.561-569.

4. "Intelligent agent based operator support and beam orbit control scheme for synchrotron radiation sources", R.P. Yadav, P.V. Varde, P.S.V. Nataraj, P. Fatnani, *International Journal of Advanced Science and Technology*, **2013**, *Vol.52*, pp.11-34.

Conferences

1. "Simulation and modelling of emission process in Microtron with type II cavity", R.P. Yadav, P. Fatnani, P.V. Varde, *4th Indian Particle Accelerator Conference (InPAC 2009)*, February 10-13, **2009**, RRCAT, Indore, India.

2. "Intelligent agent based control of INDUS TL-1", R.P. Yadav, P. Fatnani, P.V. Varde, *5th Indian particle accelerator conference (InPAC2011)*, February 15-18, **2011**, Delhi, India.

Rishi Pal Yadav

DEDICATIONS

I dedicate this thesis to my parents

ACKNOWLEDGEMENTS

I would like to express my deep sense of gratitude towards my guides Dr P.V. Varde and Dr P.S.V. Nataraj for their valuable guidance, keen interest and encouragement during the course of this work. I am extremely grateful to Dr P.D. Gupta, Dr A.P. Tiwari and Dr S.V.G. Ravindranath for their constant encouragement during this work. I am indebted to Mr P. Fatnani, Mr C.P. Navathe, Mr P.R. Hannurkar, Mr G. Singh, Mr P. Shrivastava, Dr. S.B. Roy and Dr. A. Banerjee for their support and detailed discussions on various aspects of the process.

My sincere thanks to my colleagues, Mr A. Sharma, Mr R.S. Saini, Mr R. Husain and Mr A. Chauhan for their valuable suggestions and cooperation.

I thank my wife (Mrs Sunita Yadav) for the inspiration and support she provided.

Contents

List of Abreviations

AC	Alternating Current
ABC	Agent Based Control
ADC	Analogue to Digital Converter
AGC	Automated Guided Vehicles
AI	Artificial Intelligence
AIM	Agent Information Management module
AL	ARCHON Layer
API	Application Program Interface
ATM	Asynchronous Transfer Mode networks
AT	Accelerator Toolbox, a computer simulation code
ARCHON	Architecture for Cooperative Heterogeneous On-Line systems
AR	Autoregressive
ARMA	Autoregressive moving-average model
ASP	Acceleration Start Point
BDI	Belief-Desire-Intention Model for Agents
BEDES	Beam Diagnosis Expert system
BL	Beam line
BPM	Beam Position Monitor
BPI	Beam position indicator
BR	Booster Ring
CAN	Controller area network

CIRCA	Cooperative Intelligent Real-time Control Architecture
COD	Closed orbit distortion
CRO	Cathode Ray Oscilloscope
CODES	Control System Diagnostics Expert System
CPU	Central processing Unit
DAC	Digital to analogue converter
DAI	Distributed Artificial Intelligence
DC	Direct Current
DCCT	Direct Current Transformer
DE	Differential Evolution
DP	Dipole Magnet
ELITERA	Name of SRS Facility at Trieste, Italy
ELEGANT	Computer code for accelerator simulation
EPICS	It is a SCADA Package
FCT	Fast Current Transformer
FIFO	First in first out queue
FOPDT	First order plus dead time type plant
FPGA	Field Programmable Gate Array
GA	Genetic Algorithm
GeV	Gega electron volt
GUI	Graphical User Interface
GPIB	General Purpose Interface Bus

HLCM	High Level Communication Module
HSC	Horizontal Steering Coil
ID	Insertion device
INDUS-1	450MeV electron accelerator in INDIA
INDUS-2	2.5GeV electron accelerator in INDIA
INTERRAP	It is a hybrid agent architecture
IS	Intelligent System
LabVIEW	It is a graphical programming language
LHC	Left Hand side Coil
MA	Model Agent
MAD	Methodical accelerator design, a computer simulation code
MAGAP	Model Assisted GA based Plans
MAS	Multi Agent System
MBT	Model Based Tracking
MeV	Mega electron volt
MU	Monitoring Unit
MPS	Machine Protection System
NCS	Networked control systems
NDP	New data packet
OOP	Object Oriented Programming
OP	Operating Point
PAL	Physical Access Layer

PC	Personal Computer
PCM	Planing and Coordination module
PLC	Programmable Logic Controller
PVSS	It is a SCADA Package
P/S	It is used for Power supply
PROFIBUS	one of the field buses
QD	Quadrupole Defocusing
QF	Quadrupole Focusing
RACETRACK	Computer code for accelerator simulation
RAM	Random Access Memory
RF	Radio Frequency
RHC	Right Hand side Coil
RL	Reinforcement Learning
RLS	recursive least square algorithm
RP	Reflected power
SCADA	Supervisory Control And Data Acquisition
SEM	Secondary Emission Monitor
SEWM	Secondary Emission Wire Monitor
SR	Storage Ring
SR	Synchrotron Radiation
SRS	Synchrotron Radiation Sources
SS	Safty shutter of beam lines

SVD	Singular value decomposition
TANGO	It is a SCADA Package
TCP/IP	Transmission Control Protocol/Internet Protocol
TD	Temporal Difference
TF	Transfer Function
TL-1	Transport Line-1
TL1CA	TL-1 Control Agent
TR	Teleo-Reactive
UDP	User datagram protocol
UPS	Uninterruptible Power Supply
VSC	Vertical Steering Coil
VI	Virtual Instrument, LabVIEW^{TM} code file are called as VI

List of Symbols

a_i	Used as variables
$\hat{a}_i$	Used to represent action
$a_{11}, a_{12}, ..., a_{ij}$	Used for Matrix elements
A	Set of actions = $\{\hat{a}_1, \hat{a}_2, ..., \hat{a}_n\}$
$\hat{A}$	Area in vector space
b_i	Used as variables / voltage of BPI electrodes
B	Belief set
$B(s)$	Local Magnetic field
$\breve{B}_{i,stor}$	Stored value of i^{th} BPM
$\breve{B}_{i,mes}$	Measured value of i^{th} BPM
$\breve{B}_{i,MSO}$	Value at i^{th} BPM in most probable offspring
c	Used to represent condition
C	Condition set =$\{c_1, c_2, ..., c_n\}$
$\breve{C}_{MIC}$	Control Input set =$\{i_{ca}, rf_{fre}\}$
$\mathring{C}$	Used to represent corrector magnet
d_1, d_2	Used to represent search distance
D	Desire set
D_{xx}	Drift space, where $_{xx}$ represents the drift space number
$\hat{D}$	Limiting distance
e	Beam emittance / error / charge of electron
e_x	Emittance in horizontal plane

e_y	Emittance in vertical plane
e_{noise}	Noise in system
$\hat{e}$	Used for representing Emission
$\mathring{e}$	Error component
E	Emission signal vector
$\mathring{E}$	Total error
f	Used for function
f^{ϑ}_{MIC}	State mapping function with Microtron tuned as perϑ
f_{MIC}	State mapping function with Microtron tuned as per last applied settings
f^{ϑ}_{TL1}	State mapping function with TL-1 tuned as perϑ
f_{TL1}	State mapping function with TL-1 tuned as per last applied settings
$\breve{f}$	Used to represent FCT signal
$\breve{F}$	Used to represent FCT vector
$\breve{F}_E$	FCT signal contribution due to Emission
g	Used to represent reflected power signal
G	Used to represent reflected power vector
h	Function representing possible history of actions
H_{Bump}	Used to represent bump height
i_{ca}	Cathode current of Microtron
i_{tl1}	TL-1 current(normalised)

I	Intention set
I_{BR}	Booster current
I_{MIC}	Beam current at Microtron output
I_{TL1}	Beam current at TL-1 output
I_{ca}	Cathode current of Microtron
$I_{inj,mes}$	Measured value of injection
$I_{inj,sim}$	Simulated value of injection
I_{lim}	Injection current limit
I_{ca}^{ASP}	Acceleration start point cathode current, Amperes
$\widetilde{I}$	Predicted injection current vector $= [I_{inj}(1), I_{inj}(2), ..., I_{inj}(n)]$
$J, J_i, \mathbf{J_1}, \mathbf{J_2}$	Used for representing the optimisation function
k_i	Used as variables
K	Focussing strength
$K(s)$	Local focussing strength
l	Used for representing length of magnets and drift space
$ln(x)$	Used for representing the natural logarithm of x
L_{BH}	Bump height Limit
m_i	Used as variables
$\widetilde{M}$	Magnet settings vector $= [I_1, I_2, ..., I_n]$
N	The set of Natural Numbers
p	Used for momentum, also used for position vector
$\breve{p}_{value}$	Used to represent performance value

P	Percept function
$\boldsymbol{P}$	Used for set of position vectors
$\boldsymbol{P}$	Used for set of position vectors
$\hat{P}$	Used for set of $\mathbf{P}$
$\breve{P}_{TL1}$	TL-1 operating point(operation state vector)
$\breve{P}_{MIC}$	Microtron operating point (operation state vector)
$\widetilde{P}$	Predicted start position vector
	$= \left[\widetilde{X}_{start}(1), \widetilde{X}_{start}(2), ..., \widetilde{X}_{start}(n)\right]$
q	Used for charge
$\widetilde{Q}$	Predicted position at BPM vector
	$= \left[\widetilde{X}_{BPM(i)}(1), \widetilde{X}_{BPM(i)}(2), ..., \widetilde{X}_{BPM(i)}(n)\right]$
r	Used to represent rule
$\breve{r}_{cav-fre}$	Cavity resonant frequency
$\breve{r}_{fre}$	RF frequency
R	Behavioural rule set $= \{r_1, r_2, ..., r_n\}$
$\breve{R}_{fre}$	Set of RF frequency
s	Used to represent state
S	Set of States $= \{s_1, s_2, ..., s_n\}$
$\mathbf{S_{MIC}}$	Set of Microtron states
$\mathbf{S_{TL1}}$	Set of TL-1 states
T_s	Sampling time
V	Volts

w_i	Used to represent weights
x	Coordinate in x direction, otherwise as defined
$x\prime$	Beam angle in x direction
$\hat{x}$	optimised horizontal beam position
x_d	desired horizontal beam position
$\widetilde{X}$	Beam parameter vector = (x, y, x', y')
X_E	Horizontal beam position movement due to Emission
$X_{\delta f}$	Horizontal beam position movement due to RF frequency deviation
$\widetilde{X}_{start}$	Beam parameter vector at TL-1 start
$\widetilde{X}_{end}$	Beam parameter vector at TL-1 end
$\widetilde{X}_{a,b}$	Beam parameter vector at the location a obtained by the method b
y	Coordinate in y direction, otherwise as defined
$y\prime$	Beam angle in y direction
$\hat{y}$	optimised vertical beam position
y_d	desired vertical beam position
Y_E	Vertical beam position movement due to Emission
$Y_{\delta f}$	Vertical beam position movement due to RF frequency deviation
α	Twiss parameter That gives the correlation established by optics between β and γ , $\beta\gamma = 1 + \alpha^2$)

α_x	α in horizontal plane
α_y	α in vertical plane
β	Twiss parameter (for beam size)
β_x	β in horizontal plane
β_y	β in vertical plane
δ	Dispersion
δ_x	Dispersion in horizontal plane
δ_y	Dispersion in vertical plane
δf	$= (RF_{fre} - CAV_{fre})$, frequency deviation
γ	Twiss parameter(for beam divergence)
γ_x	γ in horizontal plane
γ_y	γ in vertical plane
ρ_s	Local radius of curvature
σ	Beam size
$\psi_{a,b}$	Phase advance from the location $'a'$ to the location $'b'$
θ	Used to represent kick angles
$\pm\Delta\theta$	Used to represent angle deviation in beam lines
$\pm\Delta P$	Used to represent position deviation in beam lines
Δp	Used for momentum deviation
Δ_{start}	Iteration start limit
Δ_{stop}	Iteration stop limit
Δ_{const}	Used for representing some constant value

ϵ	Used for representing beam emittance
$\vdash$	Used for logical inference (rule deduction) for example
	$x \vdash y$ means, y is derivable from x
$\nvdash$	Used for logical negative inference ($= \neg \vdash$), for example
	$x \nvdash y$ means, y is not derivable from x
$\wp$	used for set operation
$\neg$	Logical NOT operator
$\wedge$	Logical AND operator
$\vee$	Logical OR operator
$\forall$	Universal quantification (for all), for example
	$\forall\ x : P(x)$ means, $P(x)$ is true for all x
μ	Used to represent Microtron state
$\hat{\mu}$	Used to represent optimised Microtron state
λ	Used to represent TL-1 state
$\hat{\lambda}$	Used to represent optimised TL-1 state
ϱ	Predicted beam position vector
$\boldsymbol{\xi}$	Set of predicted beam positions
ϑ	Optimised coalition state
$\hat{\varsigma}$	Set of optimised coalition states

Synopsis

Introduction

Particle accelerators are machines used for increasing the energy of charged particles. They find application in many fields of fundamental physics from elementary particles, astrophysics and cosmology to solid state, nuclear and atomic physics. In the last three decades, the synchrotron radiation (produced at synchrotron radiation source accelerator facilities) has evolved as an important tool in chemistry and biology to study molecules, atoms and nuclei structures. The control system of accelerator machines are of distributed layered architecture with interconnected software and hardware components of different layers spread over a large area and covering different subsystems of accelerators like injector, transport lines, storage ring and beam lines. The important control aspect is that all the sub-system operations are to be performed in a synchronized and sequential manner with interlocking and pre-qualification of actions at different machine operation states.

Intelligent objects are software programs called as agents that are situated in a given environment and are capable of acting in this dynamic environment towards achieving their goal. An agent program is modelled in terms of mentalistic notions such as beliefs, desires and intentions so that the developed software entity can pose some of the basic properties such as autonomy, social ability, reactivity, and proactivity, generally exhibited by humans. With these attributes, agents provide a high abstraction level for developing software and thereby potentially simplify the control system design for complex systems.

I propose to develop a method of real-time control for particle accelerators based on the intelligent agents that would enhance the system's operability, robustness and overall system performance.

Literature Survey

Software agents can be viewed as software entities designed to perform a designated function on behalf of a user/ its human counterpart. The idea of the software agent was first conceived by John McCarthy and the term was coined by Oliver G Selfridge in 1950 [1]. Since then the study on agent based systems has started and is now gaining importance. With the availability of low cost high computation intensive hardware this field is now emerging as a separate branch in artificial intelligence studies. Jennings and Wooldridge [2] define an intelligent agent as "a computer system that is capable of flexible autonomous action in order to meet its design objectives". Nwana *et. al.* [3] define an agent as "a component of software and/or hardware that is capable of acting exactingly in order to accomplish tasks on behalf of its user". Bradshaw [4] elaborates attributes that agents might possess, such as reactivity, autonomy, collaborative behaviour, knowledge-level communication ability, inference capability, persistence of identity and state over long periods of time, adaptivity (being able to learn and improve with experience), and mobility. For realization Rzevski [5] and Shen *et. al.* [6] summarized the minimum set of functional modules needed for realizing the intelligent agents as perception, cognition (or reasoning) and execution (action). The agent formulation is achieved through different agent architectures.

Architecture of an agent refers to the internal organization and interconnection of the constituent modules required to implement the agent behaviour. Based on the kind of representation and reasoning used, many different type of agent architecture descriptions appear in the literature. Subsumption architecture was first introduced by Brooks [7] for reactive agents that merely react to situations and do not reason about the world. The advantage of this approach is a faster agent response to changing environmental conditions so long as they have a predefined stimulus-response-pairing [8]. Genesereth and Nilsson [9] and others [10-12] realized agents with logic based architecture. They showed that intelligent behaviour can be generated in a system by giving that system a symbolic representation of its environment and its desired behaviour and syntactically manipulating this representation. This syntactic manipulation corresponds to logical deduction. Inspired by the human practical reasoning and decision process A.S.Rao and M.P.Georgeff [13] proposed the deliberative (beliefs-desire-intention) architecture where agent decides moment by moment which action to perform in the furtherance of desired goals. To get advantages of both reactive and deliberative architectures researchers [14, 15] proposed the hybrid architectures. With the added capabilities such as communication, coordination, cooperation and collaboration many agents of similar/different types can cooperatively work together in a multi-agent environment to achieve the goals of individuals and of the system as a whole.

In accelerator community the first use of agent application was demonstrated by Jennings [16]. Jennings connected the two diagnostic expert systems in the accelerator control environment namely CODES (control system diagnosis expert system) and BEDES (beam diagnosis expert system) by using the ARCHON (architecture for cooperative heterogeneous on-line systems) framework. Thus converting them to two distinct intelligent agents, that work together to fulfil common goal of diagnosing the accelerator faults. Another effort towards the intelligent control of accelerator was done by Klein *et.al.* [17-19]. They proposed a control system architecture based on hierarchical distributed knowledge-based controllers, each of which is an expert in controlling some section of the environment or in performing some function over that environment. These knowledge-based controllers were provided with the capabilities for planning, diagnosis, and learning, as well as knowledge acquired from human domain experts to select, sequence, and configure control actions. Nilsson proposed a new control approach based on combination of feedback-based control and traditional discrete action planning and called it as teleo-reactive [20] control. In teleo-reactive unlike traditional planning environments, no assumption is made that actions are discrete and un-interruptible and that an action's effects are completely predictable. On the contrary teleo-actions are executed as long as the action's preconditions hold and its goal has not yet been achieved (unless some other action closer to the final goal becomes activated). A short sense-react cycle ensures that when the environment changes, the control action changes to fit the new state. The strongest point of this approach is that the teleo-reactive tree execution is adaptive in that, if some unanticipated event in the environment reverses the effects of previous actions, teleo-reactive execution will typically fall back to some lower level condition and restart its work towards the top level goal. On the other hand, if something good happens, teleo-reactive execution is opportunistic i.e. when some higher condition unexpectedly becomes true, execution shifts to the action associated with that condition. Pugliese *et.al.* [21] proposed a model reference based architecture by combining the hierarchical and subsumption architecture to automatic beamline alignment problem. The strength of this approach is the coupling of an artificial intelligence and a real-time sub-system as parallel, cooperating components. Recently Schiemer *et.al.* [22] proposed the transport line tuning with artificial intelligence based scheme using neural networks and genetic algorithms. His finding has strengthened the idea of artificial intelligence based methods towards real-time control of accelerator for automated tuning.

Scope of Work

The agent based control approach has been studied for large complex systems like large industrial systems, power distribution systems, power plants and shipboard automation, and lists the advantages and limitation of the approach. The artificial intelligence based methods and agent based approach in accelerator environment has been studied towards fault diagnosis and tuning. The current state is that the present accelerator control systems are of supervisory control and data acquisition type and the philosophy of the overall control system is to provide the operators with the mechanism to monitor and control the elementary individual system parameters through sensors and actuators, like the current settings applied to the individual magnet power supplies and their read-back values. The aim of this research is to design and develop the intelligent agents for real-time control of accelerator subsystems that work in a collaborative manner towards achieving the common goal of accelerator tuning while adapting to external environment dynamically. The scope of the research work includes formulating the model of different accelerator subsystems, developing improved methods towards identification and validation of system model, system parameter measurement, simulation program development for simulating the accelerator facility. Framework development for representation and development of different accelerator sub-system control agents. Agent development and simulation studies to analyse enhancement in accelerator performance with agent based control. Agent plan formulation for cooperative accelerator tuning by distributed multiple agents. The research illustrates through simulation results that the proposed agent based methodology can successfully increase the systems robustness and operational performance under different system states.

Problem Formulation

The objective of the research is to develop a framework that allows the formulation of various agents for control of different accelerator subsystems to collectively achieve the task of cooperative accelerator tuning and gain thorough theoretical understanding of the approach. Cooperative accelerator tuning refers to the process in which agents autonomously exploit local sensed information for tuning task in cooperation with other agents and to cope with dynamic conditions. For this, first, the control system model of the accelerator facility is developed. Specifically, the basic model of the typical synchrotron radiation source facility comprising of electron source, transport line, high energy booster and storage ring is developed. For this model, the control system interface similar to the actual machine interface is built. Depending on the subsystem type, the methods of modelling through first principle or model identification through experiments have been utilized. Then, in an incremental manner for each of the sub-systems namely electron source (pre-injector), transport line, booster and storage ring, the intelligent agents are formulated. Simulation program is developed for analysing the effectiveness of the agent based control of individual subsystems and the multi-agent based scheme for the overall system. Then the scheme is extended by formulating the distributed multi-agent based control for tuning of beam orbit in the synchrotron radiation source.

An agent based control for beam transport lines: In accelerator, the transport line is used for transport of electrons/charge particle beams from source to destination. During transport, they modify the beam characteristics to match with the beam acceptance characteristics of the destination. For the scheme of agent based control for beam transport lines, the abstract architecture for beam transport line control agent has been formulated. Then, the framework for implementing the goal based agent with modular architecture has been developed. For analysing the performance of this agent as a case study, the development of the accelerator machine model comprising of electron source(Beam delivered by Microtron accelerator), transport line (Transport line-1 of INDUS-1 complex) and destination accelerator (Booster ring of INDUS-1 accelerator) has been carried out and validated through experiments. The simulation program for agent based control of transport line of this accelerator machine model is then developed. Simulations are then performed to analyse the performance enhancement in beam injection ability under dynamic behaviour of electron beam delivered from source. Then the goals of the formulated agent are extended through inclusion of model based tracking plans and the conditions for successful operation of this agent based control scheme are derived. Simulation results show that this agent based control can successfully improve the over-all system ability to deal with limited sensor/actuator failure cases.

Modelling of Microtron accelerator control system and formulation of agent based control for Microtron accelerators: Microtron is used as the pre-injector accelerator for the INDUS-1 accelerator machine, thus it acts as the initial electron source. For developing the agent based control for the source subsystem of accelerator machine as a case study, the requirements for the Microtron control agent are formulated. Then, experiments are conducted to identify the interdependence of dependent parameters emission, reflected power, FCT(represents the accelerated electron current) and beam position(X,Y) on independent parameters like cathode current, RF frequency and dipole current. Using this identified static and dynamic Microtron response, the model of Microtron accelerator is developed. With the assumption that there exists mechanism for on-line identification for networked control system and there exists an adaptive controller that can successfully drive the Microtron at desired operating point, the abstract architecture of Microtron control agent is formulated.

Extension of multi-agent control to cooperative accelerator tuning: The multi-agent based accelerator control framework is extended by formulating the plans based on cooperative tuning of accelerator subsystems. As a case study, this multi-agent control scheme is applied over the accelerator model comprised of source accelerator (Microtron), transport line(TL-1 of INDUS-1 complex), destination accelerator (Booster ring of INDUS-1 complex). Simulations are then performed to analyse the enhancement in overall injection current obtained with this scheme. This scheme is then extended by adding the system dynamics learning based cooperative optimisation plans. Simulation results show that for the dynamic environment, this scheme can produce more stable injection current with reduced number of operating point changes for transport lines.

Extension of multi-agent control to beam orbit control of synchrotron radiation sources: To control beam orbit in synchrotron radiation sources, the distributed multi-agent based accelerator tuning methodology is proposed. Based on the commonly employed multilayer control system architectures for synchrotron radiation sources, the orbit control job is distributed to multiple, low complexity reactive agents that work simultaneously and control the local orbit for individual beam lines and insertion devices in an optimized manner. For each type of agent in the proposed multi-agent methodology, agent architecture and design are formulated. This methodology is then extended by adding the constraint gradient based reinforcement learning capability to the agents. Simulation results show that with this type of scheme the accelerator on-line learning can be achieved with minimum disturbance to the nearby beam lines thereby improving the overall system performance.

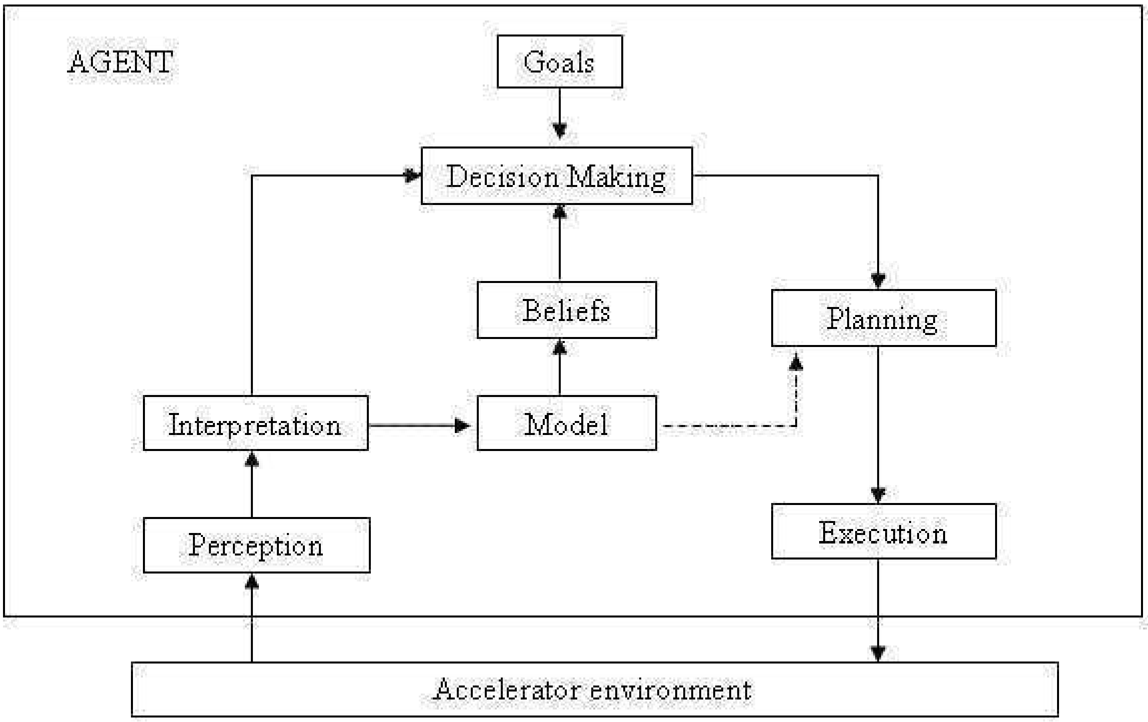


Figure 1: Transport line control agent architecture.

Agent Based Control for Beam Transport Lines

Formulation of transport line agent architecture

Taking advantage of modular control system architecture of accelerators, the model-based goal-based modular architecture of transport line agent is formulated (Figure 1). Considering transport line-1 of INDUS complex as a case study, the agent goals are formulated with associated plans as shown in Table 1. For each of these plans, the preconditions and action sequences are formulated along with the required capabilities for the agents model block.

Development of agent implementation framework

For implementation of this agent, a three layered framework is developed in graphical programming language (Figure 2) with postman-postoffice based communication feature for communication between distributed inter-agent / intra-agents modules.

Table 1: Goals and associated plans of TL-1 agent.

<table>
<tr><td>Goal</td><td colspan="3">Plan</td></tr>
<tr><td rowspan="6">Tune beam position and angle</td><td>No.</td><td>Name</td><td>Plan applicability</td></tr>
<tr><td>1</td><td>Measure Beam Position at all BPMs</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && optimise allowed</td></tr>
<tr><td>2</td><td>Correct Position and angle using two correctors without BPM data</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && (the suggested corrector values of two correctors are within constrain limits) && (without BPM correction is allowed) && (correction applicability > setlimit) && (system not in starting state)</td></tr>
<tr><td>3</td><td>Correct Position and angle using four correctors without BPM data</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && (the suggested corrector values of four correctors are within constrain limits) && (without BPM correction is allowed) && (correction applicability > setlimit) && (system not in starting state)</td></tr>
<tr><td>4</td><td>Correct Position and angle using two correctors with BPM data</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && (the suggested corrector values of two correctors are within constrain limits) && (system not in starting state)</td></tr>
<tr><td>5</td><td>Correct Position and angle using four correctors with BPM data</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && (the suggested corrector values of four correctors are within constrain limits) && (system not in starting state)</td></tr>
<tr><td rowspan="3">Cooperatively optimise injection current</td><td>1</td><td>Suggest nearest desired beam position and angle to Microtron agent for constrained TL-1 settings</td><td>(The corrector values of any of the four correctors exceeds constrain limits)&& (system not in starting state) || Event⟨suggest new-OP⟩</td></tr>
<tr><td>2</td><td>Evaluate injection current improvement for suggested beam position, angle and beam current</td><td>Event⟨Evaluate OP⟩</td></tr>
<tr><td>3</td><td>Cooperatively correct position and angle using four correctors</td><td>($inj < inj_{LO}^{lim}$) && (The corrector values of any of the four correctors exceeds constrain limits) && (system not in starting state) || Event⟨ invoke CO⟩</td></tr>
<tr><td rowspan="3">Auto Start/Stop TL-1</td><td>1</td><td>Start TL-1 normal</td><td>Event⟨Start TL-1 normal⟩</td></tr>
<tr><td>2</td><td>Start TL-1 optimised</td><td>Event⟨Start TL-1 normal fail⟩</td></tr>
<tr><td>3</td><td>Stop TL-1</td><td>Event⟨Stop TL-1⟩</td></tr>
</table>

Accelerator machine model development and validation

The overall model of accelerator machine comprising of beam source, transport line-1 and booster ring is developed for simulation studies by combining the developed individual models of these subsystems. The Microtron model can produce the electron beam with

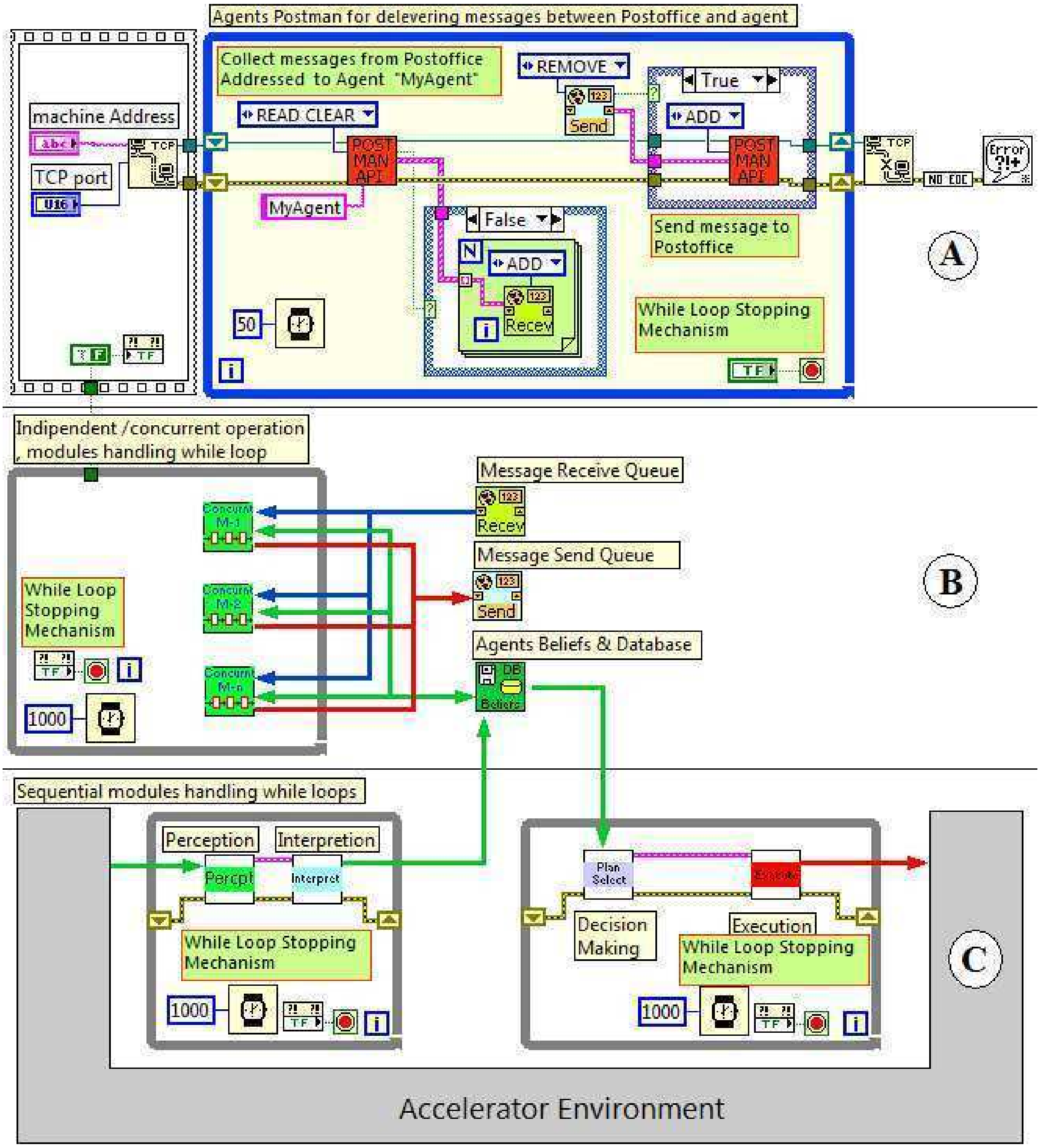


Figure 2: Framework for Agent representation, layer (A) implements the agents communication with the postoffice, layer (B) implements the concurrent process modules, layer (C) implements the agents interaction with the accelerator environment.

the design parameters ($e_x = 8.0 \times 10^{-7}$, $e_y = 3.0 \times 10^{-6}$) and beam size ($\alpha_x = -0.67$, $\beta_x = 0.96$, $\alpha_y = -0.50$, $\beta_y = 0.99$) comprises of given number of macro particles and capable of providing this beam with desired beam position and angle attributes (x, y, x´, y´). TL-1 and Booster models are developed in MAD[23]. TL-1 model accepts beam comprising of macro particles with different beam parameters. Depending upon the set

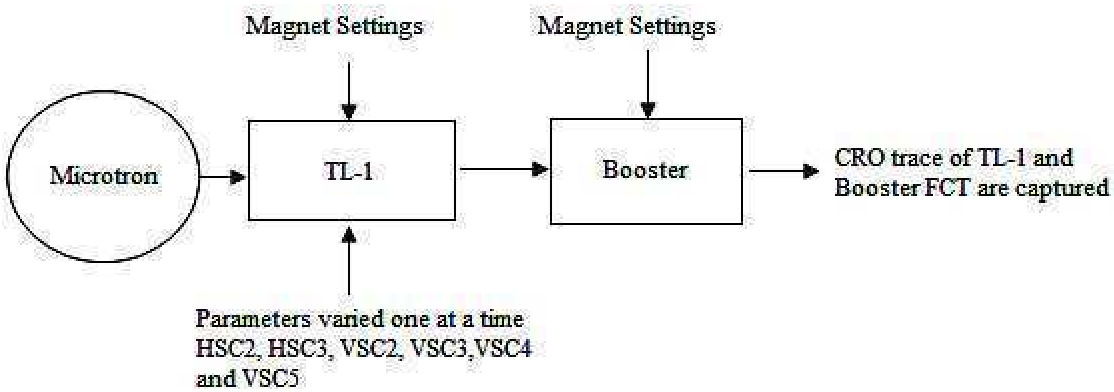


Figure 3: Setup for validation experiments.

values of all the power supplies, it tracks all the macro particles in the beam along the TL-1. It provides beam image at all beam position indicators and the beam parameters at TL-1 end. The booster model accepts the macro particle beam with beam parameters at TL-1 end. Depending upon the magnet settings, it constructs the machine lattice. Using this lattice, it then tracks the particle for the defined number of turns and produces the survived/lost attribute for each macro particle. Depending upon the survived/lost condition of macro particles, it provides the normalized beam current injected into the booster.

Table 2: Goodness of fit for TL-1 and Booster model.

Name	BR current at Inj			BR current at @3.3μs		
	SSE	R-square	RMSE	SSE	R-square	RMSE
HSC2	0.10	0.96	0.07	0.03	0.98	0.04
HSC3	0.04	0.97	0.05	0.02	0.98	0.04
HSC4	0.31	0.76	0.15	0.31	0.76	0.15
VSC2	0.05	0.98	0.05	0.07	0.97	0.06
VSC3	0.04	0.98	0.05	0.01	0.99	0.02
VSC4	0.01	0.99	0.03	0.01	0.99	0.03
VSC5	0.18	0.89	0.10	0.22	0.85	0.11

This developed model is validated through experimental results for the validation experiment setup configuration shown in Figure 3. The results show good agreement of the developed model with the experimentally observed parameter variations (Table 2).

Agent based accelerator control simulations

Combining the developed accelerator machine models and the TL-1 control agent, the simulation program is developed with block diagram shown in Figure 4. For monitoring

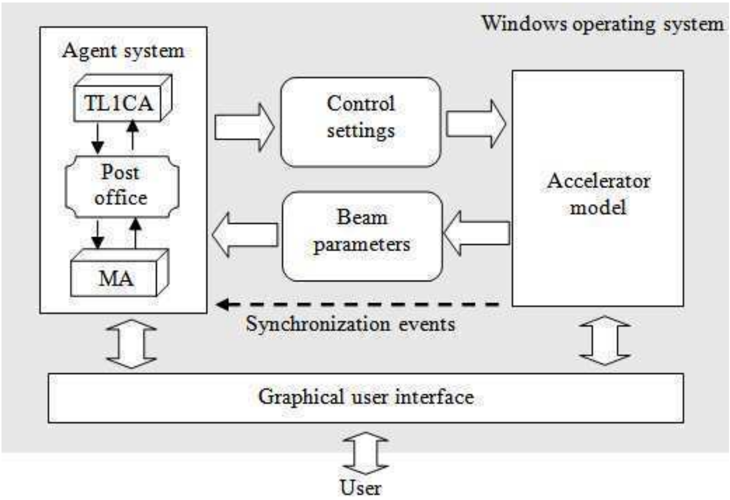


Figure 4: Simulation program Organization.

and control of simulation, a separate GUI is developed as shown in Figure 5. Where separate windows are provided for defining beam, defining beam movement, configuring accelerator simulation parameters, configuring agent parameters, and configuring agent based simulation parameters. Results of simulation can be viewed in graphic control panel and can also be stored in text file. For debugging, agent communication analyser is given in tab control. Using this GUI different fault conditions of devices can be introduced manually as well as can be defined through configuration file. Simulation results obtained for the accelerator control with and without agent based control system towards the

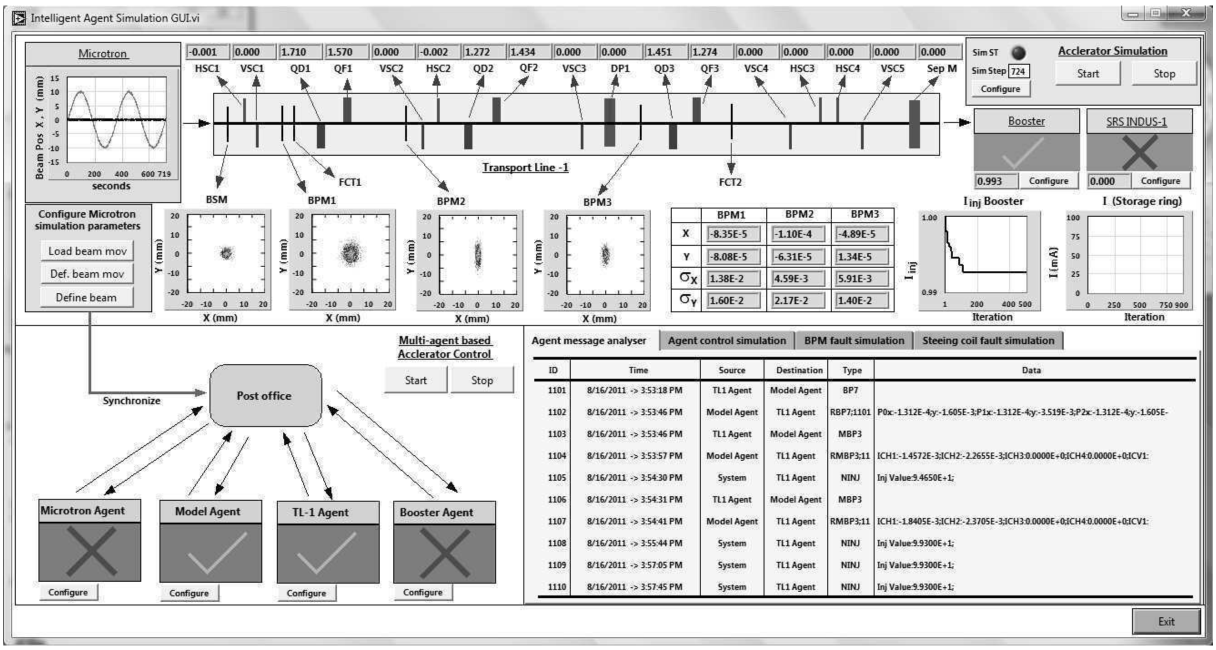


Figure 5: GUI for the agent based accelerator control simulation.

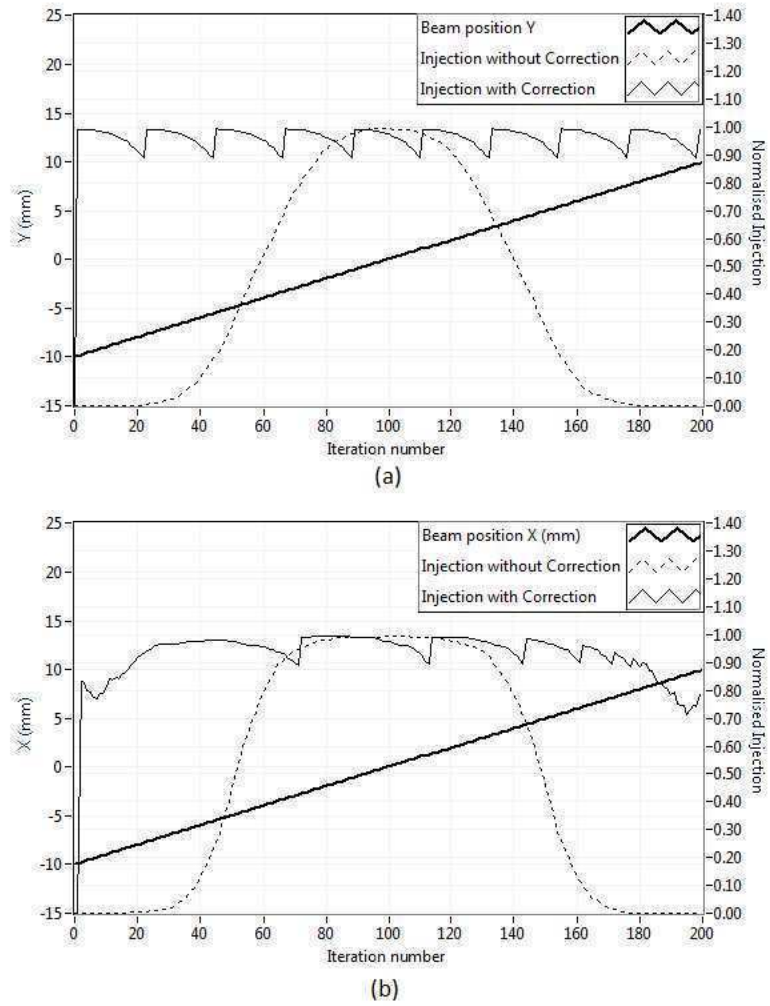


Figure 6: Injection with respect to beam position variation in (a) vertical and (b) horizontal plane with and without correction.

problem of beam position variations normally observed at the Microtron output shows that this method can effectively increase the systems operating range under such conditions while maintaining the required level of injection current (Figure 6).

TL-1 tuning using model based tracking principles

The agent goals and plans are added with model based tracking methods to estimate the beam position at the faulty BPI locations. This estimated data, along with the measured beam position data, is used by the agent for computing the system's present state, based on that it decides the next actions to be performed towards tuning of the transport lines (Figure 7). This control system scheme for different BPI failure scenarios is formulated and the condition for successful tuning of transport line by it is derived. The simulation results show that this scheme can successfully handle the limited BPI and corrector failure conditions while tuning the transport lines (Figure 8).

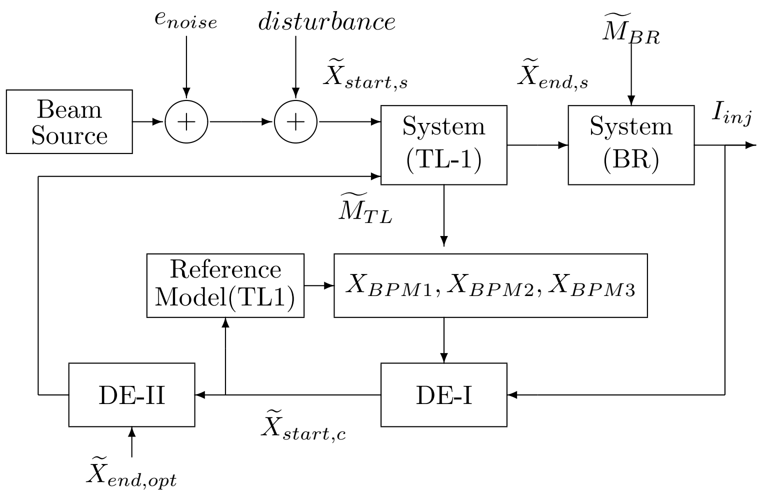


Figure 7: Block diagram for the agent based control system.

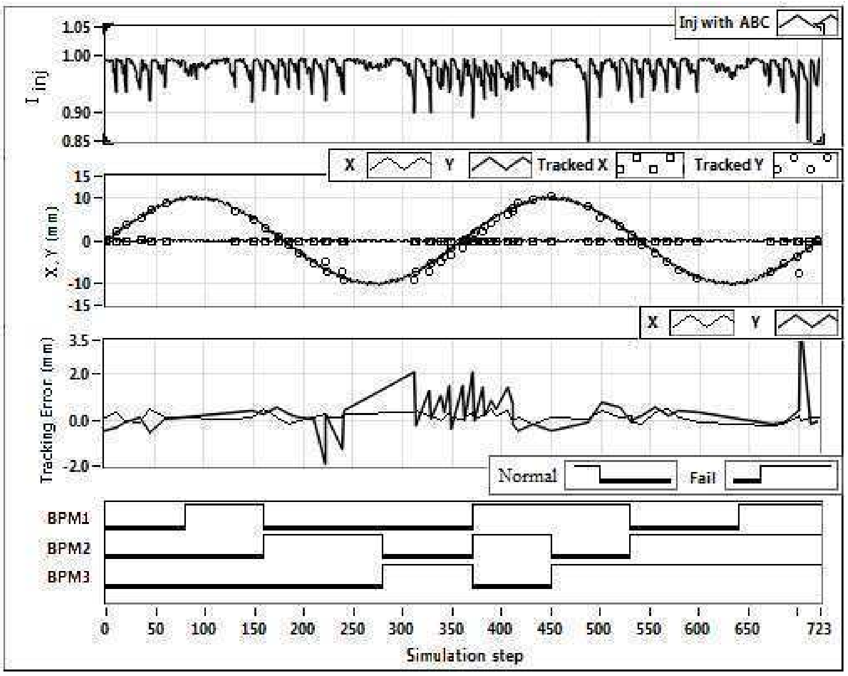


Figure 8: Injection current, tracked beam position, and tracking error for different BPI failure conditions.

Microtron Agent Development

Microtron system model identification

In the absence of first principle models of each component of Microtron system that we can easily interconnect and parameterize to create a complete model, the black-box identification method to generate a coarse time invariant control-oriented model of the system

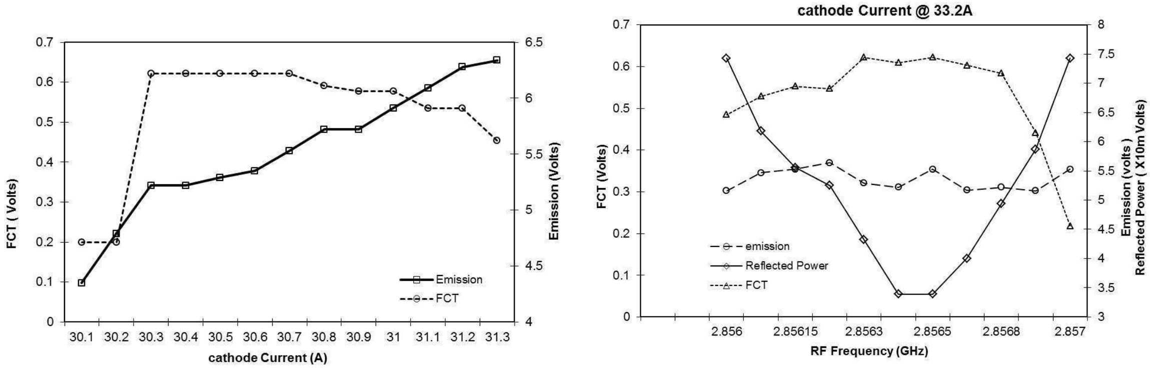


Figure 9: (a) Variation of FCT and Emission with respect to cathode current, (b) variation of FCT, emission and reflected power with respect to RF frequency.

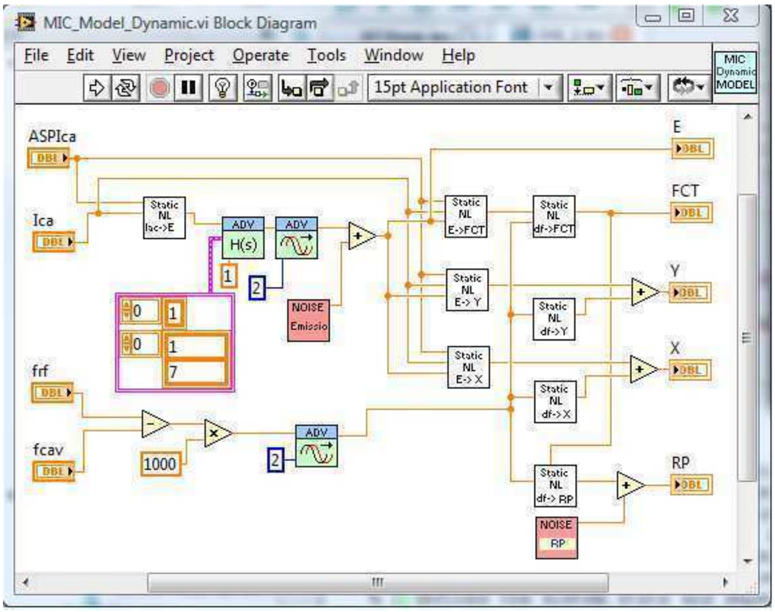


Figure 10: The Microtron Model VI block diagram.

is adopted. This model is formulated in two parts first the static part i.e. the relationships between different input output parameters at steady state and second dynamic part where the linear time invariant system transfer functions are identified. The experiments are performed on the system to identify the dependence of emission, reflected power, FCT and beam position over control parameters cathode current and RF frequency. With RF level, dipole current, RHC, LHC and mid-plane current values kept constant. From the experimentally collected data (for example Figure 9) the dependence of beam position variations on cathode current and dependence of emission current on cathode current are calculated using method of linear list squares, similarly the dependence of FCT on

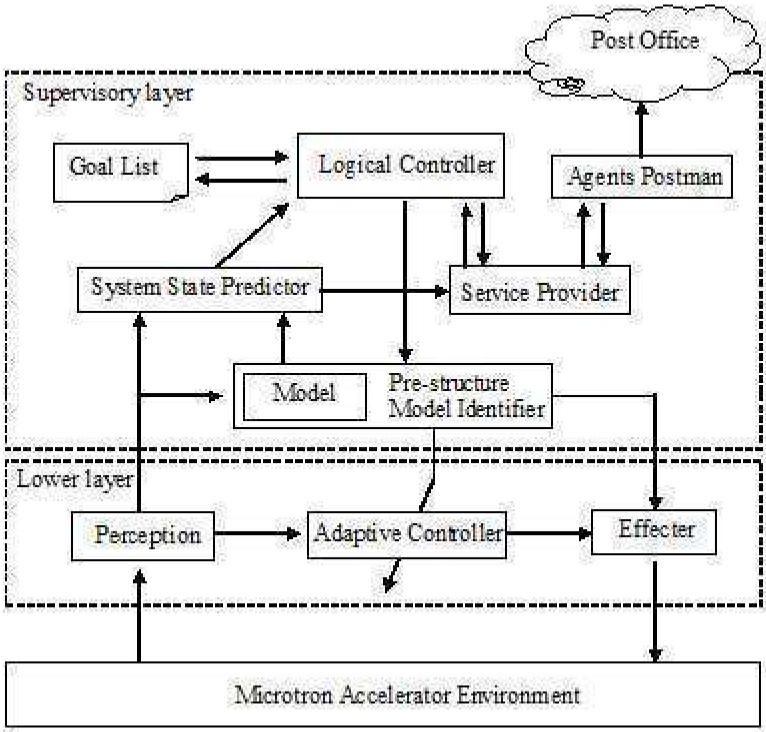


Figure 11: Microtron Agent Architecture.

cathode current and dependence of FCT on RF frequency detuning are calculated using least squares method for polynomial fitting. The non-linear least square fitting with Trust-Region Reflective Newton algorithm is used for calculating the dependence of beam position and reflected power on RF frequency detuning. Using these computed models the four-input five-output non-linear model of the Microtron system is coded in graphical programming language(Figure 10).

Microtron agent architecture

The abstract architecture of Microtron control agent is formulated with architecture shown in Figure 11. The agent architecture comprises of two loops, the first loop: comprised of perception, adaptive controller and effector blocks. This loop is responsible for continuously maintaining the machine operating point under dynamic conditions. The loop works on the principle of sense-think-control cycle where the accelerator environment is continuously sensed and if some drift in the operating point is observed the corresponding corrective action is calculated by the adaptive controller and applied to the accelerator environment through effector. The second loop is the supervisory loop responsible for autonomously controlling the agent actions and the interaction with other agents. The pre-structure model identifier block when required / asked by the logical controller identifies the plant model in the pre-structured model form by directly taking the control of effector and perception and using the predefined action recipe. This block also provides this identified model to other blocks like, system state predictor block, adaptive controller block and logical controller block for their functions. The system state predictor block continuously tries to learn the system dynamics and predicts the system dynamics for future `n´steps using the auto-regressive moving average algorithm. This block also provides the functionalities of predicting the future machine states/parameters under the influence of dynamics using the currently identified Microtron model. Service provider block is the communication interface of the agent with the other agents. It is responsible for serving the requests obtained from different agents and from logic controller which requires some data from other agents. The postman is the communication medium between the agent and the post office for exchange of messages between different agents. The logic controller is the brain of the agent and is responsible for managing and synchronizing all the activities of the agents towards the achievement of goals.

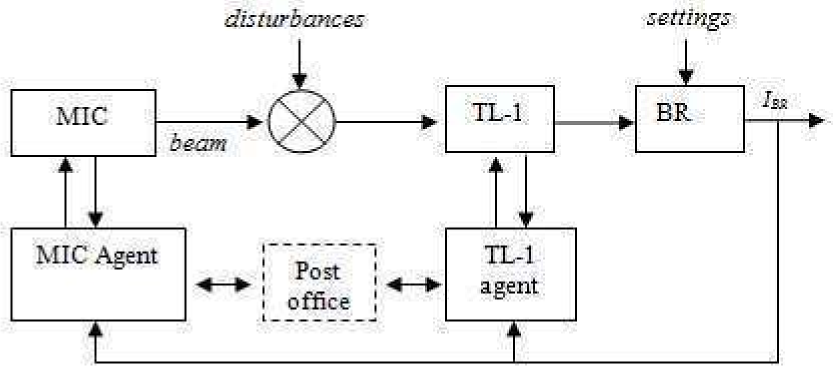


Figure 12: Block diagram for multi-agent based control of Microtron and TL-1.

Multi-agent Based Control for Accelerators

Cooperative tuning of accelerator by TL-1 and Microtron control agents

To analyse the developed methodology for controlling the pre-injector and transport line operations through multi-agent based control scheme, the developed accelerator models and agents are connected as shown in Figure 12.

The multi-agent based control of Microtron and TL-1 towards the cooperative tuning requires that both of the agents should try to maintain their individual operation to the optimum according to their local priorities on one hand and cooperatively decides their operating points such that their joint goal of increasing the overall injection current in the booster is achieved. As the beam current in booster depends on two factors first: the current produced by Microtron, and the second: the transfer efficiency (normalized beam current in TL-1) in TL-1, therefore the cost function J_1 is formulated to maximize the product of these two currents as:

$$maxJ_1 = I_{MIC}(x_t, x_t', y_t, y_t', OP_{MIC}) \times I_{TL1}(x_t, x_t', y_t, y_t', OP_{TL1}) \quad (1)$$

where the I_{MIC} and I_{TL1} are the beam current at Microtron output(which is a function of beam parameter and Microtron operating point) and normalized beam current at TL-1 output (which is a function of beam parameter and TL-1 operating point). For achieving the cooperative accelerator tuning, the agents jointly identify the operating point that maximizes the cost function J_1 given by Eq.(1) for the measured beam parameters. For the case of cooperative optimization based on the dynamics learning and with additional condition that the demand of change in the TL-1 magnet settings is to be reduced while maintaining the required level of injection current in booster. The Microtron agent at the time of deciding the new operating point for optimization cooperatively maximizes the cost function J_2 given by Eq.(2) considering the `n´steps ahead future disturbances based on the past movement history provided by the system state predictor block.

$$maxJ_2 = \sum_{i=1}^{n} I_{MIC}(x_i, x_i', y_i, y_i', OP_{MIC}) \times I_{TL1}(x_i, x_i', y_i, y_i', OP_{TL1}) \quad (2)$$

The cost function J_2 is formulated based on the fact that optimized operating point for the two agents subject to the above said criteria will be the one that not only maximizes the injection current in booster but also maintains this maximized level of current in booster for future durations under the influence of external disturbances. Thus J_2 is formulated to maximize the sum of product of I_{MIC} and I_{TL1} for the predicted future beam position movements. The simulation results for controlling the pre-injector accelerator Microtron and transport line under the influence of disturbance on beam in this scheme shows that this scheme can be used successfully for their optimal control without operator interventions.

Multi-agent based control of beam orbit in synchrotron radiation sources

In synchrotron radiation sources the users perform experiments at experimental stations attached to beam lines. In these beam lines the synchrotron radiation position is highly dependent on the electron beam position and angle at the source point. The tuning of accelerator for getting the desired electron beam position and angle at the source point is a time consuming and regular job done during commissioning of new beam lines or when accelerator is operated at new operating point. For accelerator control and tuning a novel intelligent agent based operator support and beam orbit control methodology has been developed. The developed multi-agent based scheme (Figure 13) is well suited for the multilayer control system architectures of synchrotron radiation sources. The scheme successfully distributes the orbit control job to multiple low complexity reactive agents that work simultaneously and control the local orbit for individual beam lines and insertion devices in an optimized manner. The performance of the beam line control agents and insertion device control agents is evaluated on periodic basis by monitoring agents. The gradient based constrained trainer agents are formulated that coordinates

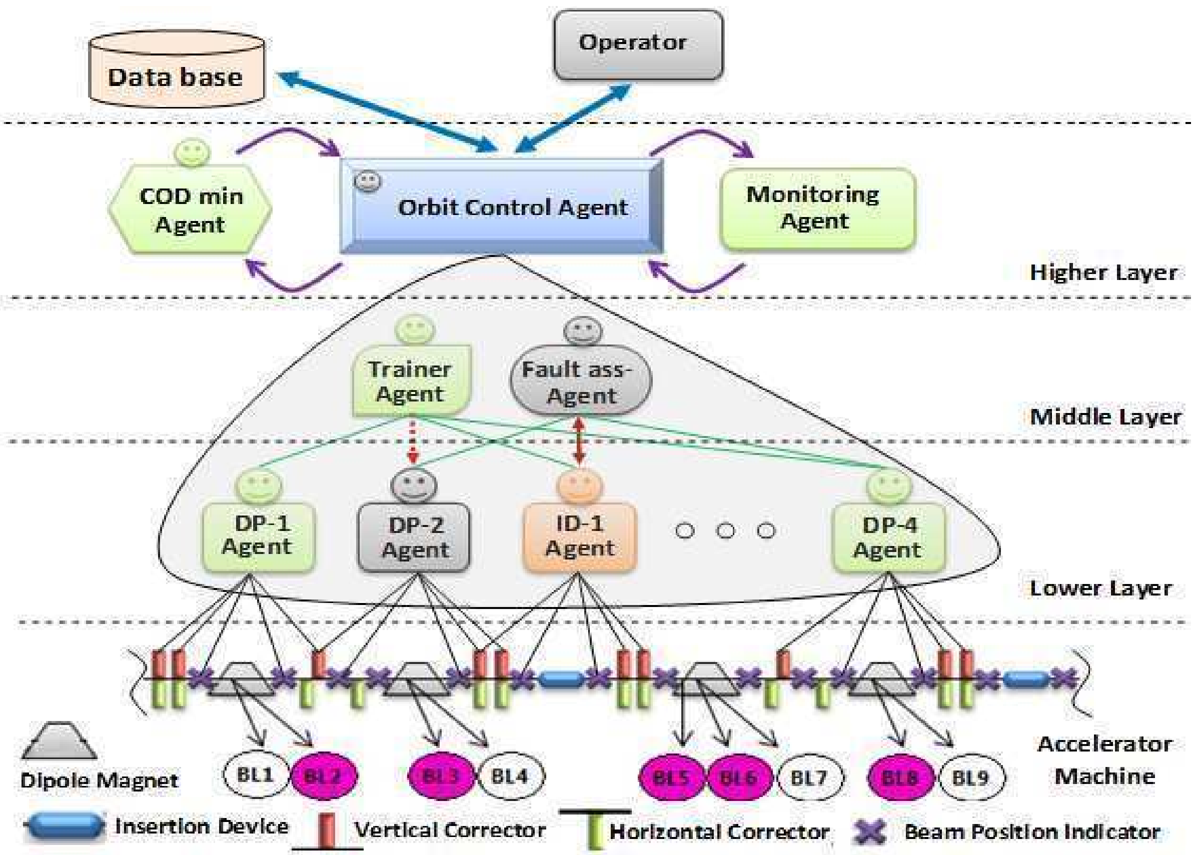


Figure 13: Organization of agents for multi-agent based beam orbit control system.

the agents training to achieve the accelerator tuning goal in this multi-agent environment. The logic based fault assistance agents are proposed for providing the fault identification and isolation. At top of all these agents the model-based goal-based orbit control agent is proposed that interacts with the user and database to generate the lower layer agents and control their activities in a synchronized and systematic manner. The proposed scheme of beam orbit control in particular is very useful for new synchrotron radiation sources, where new beam lines are in the process of commissioning as this scheme reduces the operator efforts and accelerator tuning time for providing beam to new beam lines. Further it extends the beam availability to other beam lines (already installed and in use beam lines) as the agent tunes the accelerator in systematic way and under constraints on local orbit bump leakage thereby enabling the use of other beam lines for routine experiments which otherwise was not possible.

Conclusion

The basic model of the typical synchrotron radiation source facility comprising of electron source, transport line, high energy booster and storage ring is developed using the methods of modelling through first principle and system model identification through experiments. Then for this model the control system interface similar to the actual machine interface is built and in an incremental manner for each of the subsystems namely electron source, transport line, booster and storage ring the intelligent agents are formulated. Finally combining all the models and agents the overall simulation program is developed for analysing the effectiveness of the agent based control of individual subsystems and the multi-agent based scheme. This scheme is extended by formulating the distributed multi-agent based control for tuning of beam orbit in the synchrotron radiation source where the agents of different complexity are distributed through different accelerator control system layers.

The simulation results show that the agent based control can be successfully utilized for automatic tuning of individual accelerator subsystems to improve the overall injection efficiency under the dynamic behaviour of beam at source. Further, with the integration of model based tracking concepts with the agent based control the overall system availability is improved for different sensor / actuator failure conditions. Also, it is found that the proposed multi-agent based scheme can successfully optimize the injection current by cooperatively tuning of individual subsystems and at the same time can satisfy their local tuning goals and constraints under dynamic system conditions by adding the system dynamics learning capabilities to the agent. Taking advantage of accelerator control system layered architecture the proposed multi layered distributed multi-agent framework for orbit control in synchrotron radiation sources can improve the system reliability and beam availability through systematic accelerator tuning by keeping the local orbit bump leakage under constraints.

References

[1]. Alan C. Kay, Computer Software, Scientific American, Vol. 251(3): pp.53–59, (1984).

[2]. Jennings N. R. and Wooldridge M., Agent Technology: Foundations, Applications and Markets, Springer, (1998).

[3]. Nwana H. S., Lee L. and Jennings N. R., Co-ordination in multi-agent systems, Software Agents and Soft Computing, New York, Springer, pp.42–58, (1997).

[4]. Bradshaw J. M., An introduction to software agents, in Software Agents, AAAI Press/MIT Press, pp. 3–46, (1997).

[5]. RZEVSKI G., Designing Intelligent Machines Vol. 1, Perception, Cognition and Execution, Butterworth-Heinemann, (1994).

[6]. Shen W., Norrie D. H. and Barthes J.P., Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing, Taylor and Francis, (2001).

[7]. Brooks R.A., A Robust Layered Control System for a Mobile Robot, In IEEE Journal of Robotics and Automation, Vol. 2(1), pp.14–23, (1986).

[8]. Bonasso R. P., Kortenkamp D., Miller D. P., and Slack M., Experiences with an Architecture for Intelligent Reactive Agents, in Intelligent Agents II (LNAI Vol. 1037), pp. 187–202, Springer-Verlag, Berlin, Germany, (1996).

[9]. Genesereth M. R. and Nilsson. N., Logical Foundations of Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, CA, (1987).

[10]. Konolige K., A Deduction Model of Belief, Pitman Publishing: London and Morgan Kaufmann, San Mateo, CA, (1986).

[11]. Lesperance Y., et. al., Foundations of a logical approach to agent programming, in Intelligent Agents II (LNAI Vol. 1037), pp. 331–346, Springer-Verlag, Berlin, Germany, (1996).

[12]. Rosenschein S. J. and Kaelbling L. P., A situated view of represent-action and control, in Computational Theories of Interaction and Agency, pp 515–540. The MIT Press, Cambridge, MA, (1996).

[13]. Rao A. and Georgeff M. BDI Agents: from theory to practice, in Proceedings of the First International Conference on Multi-Agent Systems (ICMAS95), pp. 312-319, The MIT Press, Cambridge, MA, USA, (1995).

[14]. Ferguson I. A., Towards an architecture for adaptive, rational, mobile agents, in Decentralized AI 3 – Proceedings of the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91), pp. 249–262. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, (1992).

[15]. Muller J. P., Pischel M., Modelling Interacting Agents in Dynamic Environments, in Proc. of Eleventh European Conf. On Artificial Intelligence, ECAI-94, pp.709–713, (1994).

[16]. Jennings, N. R., Varga L. Z., Aarnts R., Fuchs J. and Skarek P., Transforming Standalone Expert Systems into a Community of Cooperating Agents, Int. Journal of Engineering Applications of Artificial Intelligence, Vol. 6(4), pp.317–331, (1993).

[17]. Klein W.B., Westervelt R.T., and Luger G.F., An Architecture for intelligent control of particle accelerators, available from `http://www-bd.fnal.gov/icalepcs/abstracts/PDF/wpo63.pdf`, (2011).

[18]. Klein W.B., Stern C.R., Luger G.F., and Olsson E.T., An Intelligent Control Architecture for Accelerator Beamline Tuning, in Proceedings of AAAI / IAAI, pp.1019–1025, (1997).

[19]. Klein W.B., Westervelt R.T., and Luger G.F., A general purpose intelligent control system for particle accelerators. in Journal of Intelligent and Fuzzy Systems, New York: John Wiley, (1999).

[20]. Nilsson N.J., Teleo-Reactive Programs for Agent Control, Journal of Artificial Intelligence Research, Vol. 1, pp.139–158, (1994).

[21]. Pugliese R., Bille F., Abrami A. and Svensson O., Applying Intelligent System Concepts to Automatic Beamline Alignment, International Conference on Accelerator and Large Experimental Physics Control Systems 1997, 3–7 Nov 1997, Beijing, China, (1997).

[22]. Schirmer D. Hartmann P., Buning T. and Muller D., Electron Transport Line Optimization using Neuronal Networks and Genetic Algorithms, Proceedings of 9th European Particle Accelerator Conference EPAC2006, Edinburgh, Scotland, 26–30 June, (2006).

[23]. MAD web page `http://mad.web.cern.ch/mad/`, (2011).

List of Figures

List of Tables

Chapter 1

Introduction

Particle accelerators are machines used for accelerating the energy of charged particles. They find application in many fields of fundamental physics from elementary particles, astrophysics and cosmology to solid state, nuclear and atomic physics. In the last three decades, the synchrotron radiation (produced at synchrotron radiation source accelerator facilities) has evolved as an important tool in chemistry and biology to study molecules, atoms and nuclei structures. Recently, with the appearance of spallation sources to produce neutrons, the field of condensed matter and material science studies have also been added to the accelerator's scientific application list.

With the demand for increase in the energy of charged particles, the size and complexity of these machines have increased by many times with a large number of subsystems spread over large geographical areas. Thus, much more complex instruments and machines are needed to be run in a synchronized and sequential manner for their operation. Also, the number of parameters to be controlled and monitored during the experiments have increased to such an extent that the manual control and observation is beyond the scope of human capabilities. This has led to the development of accelerator control systems in accelerator facilities. In today's accelerator facilities, almost all the operations are exercised remotely and centrally with operators sitting in control room and working from operator console computers with the help of dedicated Graphical User Interfaces (GUI) of each subsystem. The need to enhance the plant operational reliability, availability and safety has encouraged control system engineers to develop improved methods of monitoring and control for such large complex machines. This has attracted the researchers' attention towards the use of artificial intelligence (AI) methods with accelerator control systems [56, 57, 58, 59, 60, 90, 102, 113]. A more detailed review of this is presented in chapter 2. Almost all of these methods are based on distributed hierarchical architectures for control; combining heuristic, knowledge-based and conventional control methods with reasoning, search and pattern recognition methodologies from artificial intelligence research. Although these methodologies successfully integrate the artificial intelligence concepts with the accelerator controls thereby extending the benefits of AI to accelerator control system. But, none of them actually does this through the concept of software agent based methodology.

Unlike the conventional methods of system control, in agent based systems, the elementary entity are software agents. An agent program is modelled in terms of mentalistic notions such as beliefs, desires and intentions so that the developed software entity can pose some of the basic properties such as autonomy, social ability, reactivity, and proactivity, generally exhibited by humans. With these attributes, agents provide a high abstraction level for developing software and thereby potentially simplify the design of complex systems. From designers perspective, the agent based programming approach extends altogether new possibilities to system control by providing mechanisms for capability encapsulation rather than function encapsulation where the agents being autonomous are dedicated towards fulfilling their design goal on one side but are also flexible in deciding the ways to do so at run time by choosing the appropriate alternate plan on other side.

Further to this, their lively ability to interact with the environment to decide their future course of action strongly advocates their use for poorly modelled, partially observable and dynamic systems where conventional methodologies fail to work. Particularly for the complex distributed systems comprised of many subsystems like modern accelerator facilities, large industrial systems [80, 63], power distribution systems [106], power plants [35], multimedia retrieval system [32], mobile e-commerce [107], robotics [16] and shipboard automation [79] the agent's social ability for cooperative working towards common goal justifies their use as loosely coupled distributed multi-agent based control thus enhancing the overall system operability, diagnosablity and reliability.

Inspired by the distributed and hierarchical architecture of accelerator control systems and the advantages of agent based methods, Jennings *et.al.*[41] first introduced the intelligent agent based methods to accelerator control environment by transforming the two separate standalone expert systems namely CODES(control system diagnosis expert system) and BEDES (beam diagnosis expert system) to two cooperative agents that work together to fulfil the common goal of diagnosing the accelerator fault. He proposed and formulated the underlying generic distributed artificial intelligence framework ARCHON that aided the development process by providing a language, a set of structures, and tools with which the necessary infrastructure and support mechanism for interacting agents can be instantiated. Since then this framework has been applied in different areas like electricity distribution management, electricity transport management, cement factory control and many more and showed the strength of multi-agent based approach towards the control of large distributed complex systems but not much work has been done towards the real-time control of particle accelerators based on the agent based concepts. Thus, it leaves the scope open for challenging work of design and development of intelligent agents for real-time control of accelerator subsystems that work in a collaborative manner towards achieving the common goal of accelerator tuning while adapting to external environment dynamically. In this context, the traditional technique of accelerator control usually cannot be applied as they are based on the supervisory control of the subsystems rather than the notion of role played by the subsystem in the overall broader perspective of the system. However, at the same time, the supervisory control and data acquisition infrastructure of accelerator control system can be utilised for building the basic system interface between the agents and the accelerator machine.

1.1 Motivation and objective of the thesis

The present accelerator control systems are of supervisory control and data acquisition type comprised of mainly two parts.The first part is the hardware that directly interfaces with the sensors and actuators. Normally this hardware part is spread over one or two layers with many small computers / controllers connected over field buses constituting the lower layer and higher computing capacity based computers constituting the upper layer. One or more upper layer computers logically combine to make the control system for one sub-system. The control systems of various sub-systems together make the overall hardware part of control system for the complete facility. The second part is the software part that comprises of the lower layer subroutines running at the lower layer controllers, the control scripts and GUI algorithm running at the operator computer console. The philosophy of the overall control system here mainly is to provide the operators with the mechanism to monitor and control the elementary individual system parameters through sensors and actuators, like the current settings applied to the individual magnet power supplies and their read-back values. A team of operators headed by the shift-in-charge and the physicist manages the overall operation of the facility, where the operator specialised for a subsystem ensures that the settings issued to him are applied over the machine in the systematic manner as per the different system operation states. Whenever the system is needed to operate over some other operating point under dynamic situations the physicist being the system expert reformulates the new system settings according to the then observed system state and his/her accelerator domain knowledge. For this, he/she usually takes help of advanced tools like accelerator models and accelerator physics related design codes. He/She then distributes the new formulated work plan to the subsystem operators for accelerator tuning. The agent based control for accelerator system will enhance this overall accelerator control philosophy by distributing some of the accelerator physicist responsibilities directly to the machine subsystems by assigning goals to the subsystem intelligent agents. With the agent based control system, the emphasis will change from working out the suitable control parameter values for accelerator subsystem tuning to the direct goal assignment to subsystem control agent. These assigned goals will be in terms of accelerator physics parameters at much higher level of abstraction, such as, in place of deciding the current needed by the magnet to correct the beam position at some location, the direct intended beam location(X and Y) will be given to the agent. Further, the agents being autonomous, will relax the burden of operators through independent decisions and deliberated provident planning concerning their actions, based on the nature of the environment they are situated in. Another major advantage of this agent based concept in accelerator control will be the enhanced overall system operability as the agents being fast and flexible could incorporate the run-time learning abilities, thus relaxing the accurate modelling pre-requirement as well as improving the system robustness. Moreover, the agents' social ability of inter-agent interaction could be further utilised for on-line prioritised optimisation and cooperative tuning of accelerator subsystems where the constituent agents decide their future course of actions through mutual consensus. Finally by carefully distributing these intelligent agents to different hardware layers in multi-layer accelerator control system architecture, the effective resource utilization could be increased.

The topic of this thesis is multi-disciplinary and touches many fields; artificial intelligence, agent technologies, distributed multi-agent based problem solution, accelerator control and software engineering all of which have now become distinct subjects, studied to the varying depths and thus cannot be covered in this thesis in totality. Therefore, the aim of this thesis is to work out the accelerator control system model based on the distributed intelligent agents and analyse it for some of the critical performance gaining parameters like- operational repeatability, parameter optimisation, fast tuning of system, system reliability and robustness as compared to existing control system. more specifically, the thesis addresses the questions:

- Which agent architectures are more suited for transforming the existing accelerator subsystems into intelligent agents using the existing control system infrastructure?

- How can we represent such intelligent agents suitable for integration with the accelerator control system?

- How to distribute different agents over the multi-layer accelerator control system framework for exercising the multi-agent control approach?

The above questions have been addressed in the context of synchrotron radiation source accelerator facilities which are comprised of many subsystems. It is demonstrated that different agent architectures are suited for formulations of different subsystem agents. The formulated agent architecture successfully controls the subsystems in cooperative manner for improving the overall robustness and system performance.

1.2 Thesis accomplishment and approach

The overall goal of the thesis is to develop a framework that allows the formulation of various agents for control of different accelerator subsystems to collectively achieve the task of cooperative accelerator tuning and gain thorough theoretical understanding of the approach. Cooperative accelerator tuning refers to the process in which agents autonomously exploit local sensed information for tuning task in cooperation with other agents and to cope with dynamic conditions.

Towards this goal, first, the control system model of the accelerator facility is developed. Specifically, the basic model of the typical synchrotron radiation source facility comprising of electron source, transport line, high energy booster and storage ring is developed. For this model, the control system interface similar to the actual machine interface is built. Depending on the subsystem type, the methods of modelling through first principle or model identification through experiments have been utilised. Then in an incremental manner for each of the subsystems namely electron source(pre-injector), transport line, booster and storage ring, the intelligent agents are formulated. Simulation program is developed for analysing the effectiveness of the agent based control of individual subsystems and the multi-agent based scheme for the overall system. Then the scheme is extended by formulating the distributed multi-agent based control for tuning of beam orbit in the synchrotron radiation source.

1.3 Contribution of the thesis

In this thesis, original contributions in the areas of multi-agent systems and particle accelerator control have been made. More specifically, the contributions of this thesis are as follows.

- **An agent based control for beam transport lines:** In accelerator, the transport line is used for transport of electrons/charge particle beams from source to destination. During transport, they modify the beam characteristics to match with the beam acceptance characteristics of the destination. For the scheme of agent based control for beam transport lines, the abstract architecture for beam transport line control agent has been formulated. Then, the framework for implementing the goal based agent with modular architecture has been developed. For analysing the performance of this agent as a case study, the development of the accelerator machine model comprising of electron source(Beam delivered by Microtron accelerator) , transport line (Transport line-1 (TL-1) of INDUS-1 complex) and destination accelerator (Booster ring of INDUS-1 accelerator) has been carried out and validated through experiments. The simulation program for agent based control of transport line of this accelerator machine model is then developed. Simulations are then performed to analyse the performance enhancement in beam injection ability under dynamic behaviour of electron beam delivered from source. Then the goals of the formulated agent are extended through inclusion of model based tracking plans and the conditions for successful operation of this agent based control scheme are derived. Simulation results show that this agent based control can successfully improve the over-all system ability to deal with limited sensor/actuator failure cases.

- **Modelling of Microtron accelerator control system and formulation of agent based control for Microtron accelerators:** Microtron is used as the pre-injector accelerator for the INDUS-1 accelerator machine, thus it acts as the initial electron source. For developing the agent based control for the source subsystem of accelerator machine as a case study, the requirements for the Microtron control agent are formulated. Then, experiments are conducted to identify the interdependence of dependent parameters Emission, Reflected power, Forward power, FCT(represents the accelerated electron current) and beam position(X,Y) on independent parameters Cathode current, RF frequency and Dipole current. Using this identified static and dynamic Microtron response, the model of Microtron accelerator is developed. With the assumption that there exists mechanism for on-line identification for networked control system and there exists an adaptive controller that can successfully drive the Microtron at desired operating point, the abstract architecture of Microtron control agent is formulated.

- **Extension of multi-agent control to cooperative accelerator tuning:** The multi-agent based accelerator control framework is extended by formulating the plans based on cooperative tuning of accelerator subsystems. As a case study, this multi-agent control scheme is applied over the accelerator model comprised of source accelerator(Microtron), Transport line(TL-1 of INDUS-1 complex), destination accelerator (Booster ring of INDUS-1 complex). Simulations are then performed to analyse the enhancement in overall injection current obtained with this scheme. This scheme is then extended by adding the system dynamics learning based cooperative optimisation plans. Simulation results show that for the dynamic environment, this scheme can produce more stable injection current with reduced number of operating point changes for transport lines.

- **Extension of multi-agent control to beam orbit control of synchrotron radiation sources:** To control beam orbit in synchrotron radiation sources, the distributed multi-agent based accelerator tuning framework is proposed. Based on the commonly employed multilayer control system architectures for synchrotron radiation sources, the orbit control job is distributed to multiple, low complexity reactive agents that work simultaneously and control the local orbit for individual beam lines (BL) and insertion devices (ID) in an optimized manner. For each type of agent in the proposed multi-agent framework, agent architecture and design are formulated. This framework is then extended by adding the constraint gradient based reinforcement learning capability to the agents. Simulation results show that with this type of scheme the accelerator on-line learning can be achieved with minimum disturbance to the near by beam lines thereby improving the overall system performance.

1.4 Organisation of the thesis

The rest of this thesis is organised as following:

- **Chapter 2** describes the intelligent objects, agents from literature and presents the different agent architectures proposed and utilised by researchers towards achieving the complex goals. It further extends the agent based notion to the multi-agent based problem solution and discusses the main agent attributes involved in the multi-agent based systems. Finally, it discusses the efforts done in past towards utilising the agent based control in particle accelerator control environment.
- **Chapter 3** describes the accelerator physics, different particle accelerator components, their mathematical models and important accelerator physics parameters. It discusses different simulation codes utilised in accelerator simulation and presents the model of electron beam, Transport Line-1 and Booster utilised in this thesis work for simulating particle accelerator machine subsystems. Finally, it describes the validation of the formulated model with the experimental results carried over the Transport Line-1 and Booster of INDUS-1 complex.
- **Chapter 4** discusses the control systems of particle accelerators. In particular, the system requirements for synchrotron radiation source control systems are discussed

alongwith the history, evolution and future trends in synchrotron radiation source control systems.

- **Chapter 5** discusses the formulation of intelligent agents for different accelerator subsystems. Particularly, the intelligent agent architectures formulated for TL-1 intelligent agent, Microtron intelligent agent and different INDUS-2 subsystem agents are discussed. It, then, describes the simulation program design and presents the overall system simulation results of accelerator control with the formulated intelligent agents. It also discusses the model based tracking and system identification in networked condition concepts, augmented with the presented agent architectures. It, then, extends the agent based accelerator control approach by discussing the distributed, multi-agent based accelerator control where two different agents TL-1 control agent and Microtron agent work cooperatively towards tuning of both the accelerator subsystems to improve the overall accelerator performance. Finally, it discusses the distributed intelligent agent based beam orbit control for synchrotron radiation sources where seven different types of agents, distributed over three different accelerator control system layers work cooperatively for controlling the beam orbit in an optimised manner.
- **Chapter 6** summarises the contribution of this thesis. It then describes several potential applications in which this work can be further developed and some future promising directions.

Chapter 2

Intelligent agent

Software agents can be viewed as software entities designed to perform a designated function on behalf of a user/ its human counterpart. For example, an accelerator controlling agent can be viewed as the virtual operator that controls the accelerator operations on behalf of the actual operator.

2.1 Agents as intelligent objects

The idea of the agent was first conceived by John McCarthy and the term was coined by Oliver G Selfridge in 1950[52]. Their view of an agent was that of a system which, given a goal, could carry out the details of the appropriate computer operations and could ask for and receive advice, offered in human terms, when it was stuck. An agent would be a soft robot living and doing its business within the computer world [52]. Since then, the agent concept has been studied by many researchers from varied fields such as e-commerce [89], web based applications [75], distributed computing [86], medical diagnosis [37], supply chain distribution [71], production scheduling [1], industrial process control [80] and many more. To distinguish the intelligent agents from classical programs, different researchers have given different definitions.

Jennings and Wooldridge [45] define an intelligent agent as "a computer system that is capable of flexible autonomous action in order to meet its design objectives". By flexible, they mean that the system must be responsive, proactive, and social. Agents should be able to perceive their environment and respond in a timely fashion to changes (responsive). Agents should be opportunistic or goal-directed when they respond to changes (proactive). Agents should also cooperate with other agents or humans in order to solve their problems and achieve the system goal (social). By autonomy, they mean that the system should be able to act without the direct intervention of humans or other agents, and should have control over its own actions and internal state.

Nwana and Ndumu [84] define an agent as "a component of software and/or hardware that is capable of acting exactingly in order to accomplish tasks on behalf of its user". Parunak [85] defines intelligent agent as "an active object with initiative", the next extended step to object-oriented programming in software evolution. While objects are passive and gain control only when some external entity sends them a message, agents can initiate actions and maintain control over localized code and data.

Hayes [34] defines an agent as "an entity that is capable of carrying out goals, and is part of a larger community of agents that have mutual influence on each other". She emphasizes that partial autonomy and being part of a community are distinguishing properties of agent-based systems. Autonomy gives the system robustness and modularity and being part of a community makes it possible to build organizations of agents whose net effect is greater than the sum of their parts.

Bradshaw [14] enumerates attributes that agents might possess, such as reactivity, autonomy, collaborative behaviour, knowledge-level communication ability, inference capability, persistence of identity and state over long periods of time, adaptivity (being able to learn and improve with experience), and mobility.

From the above stated authors' views, a concise definition of an intelligent agent can be formulated as an agent being an encapsulated software system situated in some environment, and developed with the notion of rational decision making in choosing their actions in an autonomous manner in this environment in order to maximise its chances of success towards achieving its goals. Further, intelligent agents must posses some of the attributes such as

- capable of acting in an environment
- able to communicate directly with other agents
- is driven by a set of tendencies (in the form of individual objectives or of a satisfaction/survival function which it tries to optimise)
- has resources of its own
- capable of perceiving its environment (but to a limited extent)
- has only a partial representation of its environment (and perhaps none at all)
- possesses skills and can offer services
- can improve its skills through learning
- tends towards satisfying its objectives, taking account of the resources and skills available to it and depending on its perception, its representations and the communications it receives.
- capable of taking initiatives to adapt in dynamic environment

The realization of intelligent agents is mainly done by encapsulating the above stated attributes in modules and then these modules serve as the building block for the overall realisation of the agent structure.

Rzevski [97] and Shen *et. al.* [105] have summarised the minimum set of such modules needed for realising the intelligent agents as perception, cognition (or reasoning) and execution (action). Shen *et. al.* [105] proceeds further to identify additional modules that may be included within the internal structure of an agent:

- communication interface
- social knowledge
- self knowledge (self representation)
- domain knowledge (domain representation)
- knowledge management
- learning
- problem solving methods
- co-ordination
- planning and scheduling
- control
- conflict management
- application interfaces

For our intelligent agent representation, we will follow that the model of perception, cognition and execution is a valid generic representation for any agent since the items in the above list could quite easily be grouped under these three headings.

2.2 Intelligent agent architectures

From conceptual abstract notion of intelligent agent to concrete realization of the actual intelligent agent, agent architectures play important role. Architecture of an agent refers to the internal organisation and interconnection of the constituent modules required to implement the agent behaviour. Agent architectures are thus linked to agent type and may be classified by behaviour or alternatively by the type of organisation structure.

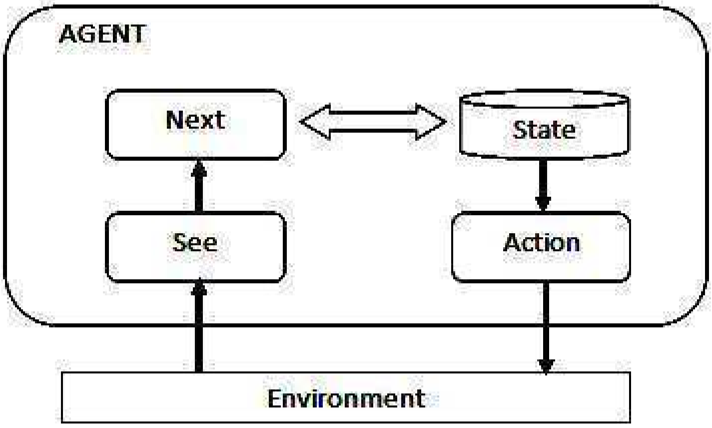


Figure 2.1: An abstract agent.

Based on the kind of representation and reasoning used, many different types of agent architecture descriptions appear in the literature. Some of the widely recognised and well studied architectures are briefly described below.

2.2.1 An abstract agent

Starting from the agent definitions that the agents are situated in an environment, are capable of observing this environment through sensors, can work on this environment through some actuators and they posses some minimum attributes like *perception*, *cognition* and *execution*, the block diagram for such an abstract agent is shown in figure 2.1. Here the block *see* performs the function of percept preparation i.e. the preparation of system variables from the sensors data that can serve as agent/environment state defining variables (or at-least assist in identifying the system states). The block *next* plays the role of providing the cognitive behaviour to the agent through action deliberations and block *action* represents the agent's mechanism for altering the environment through actuators.

Let S be the set of environment states, $S = \{s_1, s_2, ...\}$, A be the set of actions, $A = \{\hat{a}_1, \hat{a}_2, ...\}$. Now, if the sequence of agents interaction with the environment is provided as history, i.e, the sequence of state-action pairs.

$$h : s_0 \xrightarrow{\hat{a}_0} s_1 \xrightarrow{\hat{a}_1} s_2 \xrightarrow{\hat{a}_2} ...s_{i-1} \xrightarrow{\hat{a}_{i-1}} s_i \xrightarrow{\hat{a}_i} ...$$

where s_0 is the initial state and $\hat{a}_i$ is the action the agent performs when it is in the state s_i. Now for any h to be a possible history of an agent starting from initial state s_0, all actions are generated from state sequence.

$$\forall i \in N, \hat{a}_i = action((s_0, s_1, ..., s_i))$$

and every new state must belong to the set of possible environmental states reachable from the previous state by applying the selected action.

$$\forall i \in N, i > 0, s_i \in env((s_{u-1}, \hat{a}_{u-1}))$$

now for a purely reactive agent(i.e the agent which decides its next action to be performed without referring to its history) will be described by the set of actions situation pairs given as.

$action : S \longrightarrow A$

where there is at-least one action mapping present from each state in S to actions in A. In a more formal way, the function *see* will be a mapping of system states to percepts.

$see : S \longrightarrow P$

and the function *action* will be a mapping from current percept to the actions.

$action : P^* \longrightarrow A$

and the function *next* is the system state updating function

Algorithm 1: Execution steps of an abstract agent.

Data: observed environment state($s : S$), action set A
Result: perform action $\hat{a}$
initialise internal state to i_0 ;
repeat
    /*`observe environment for state` */
    READ s ;
    /*`and generate perception` */
    $p \leftarrow see(s)$;
    /* `select action according to internal state and percept` */
    $\hat{a} \leftarrow action(next(i_0, p))$;
    /*`perform selected action` */
    DO $\hat{a}$;
until *stop*;

$next : P^* \times P \longrightarrow P^*$

The execution cycle for such an abstract agent is shown in Algorithm 1

2.2.2 Subsumption agent architecture

Subsumption architecture falls in the category of reactive architectures where agents merely react to situations and do not reason about the world. Usually, both, the agents and the actions are relatively simple and global properties are seen as emerging from the interaction of behaviours [10, 39, 93]. The advantage of this approach is a faster agent response to changing environmental conditions so long as they have a predefined stimulus-response-pairing.

Subsumption architecture was first introduced by Brooks [11] for reactive agents and is mainly motivated by the following facts

- Intelligent agents can be designed without encapsulating the decision-making based on syntactic manipulation of symbolic representations of knowledge.
- The rational behaviour cannot be disembodied but is a product of the interaction the agent maintains with its environment.

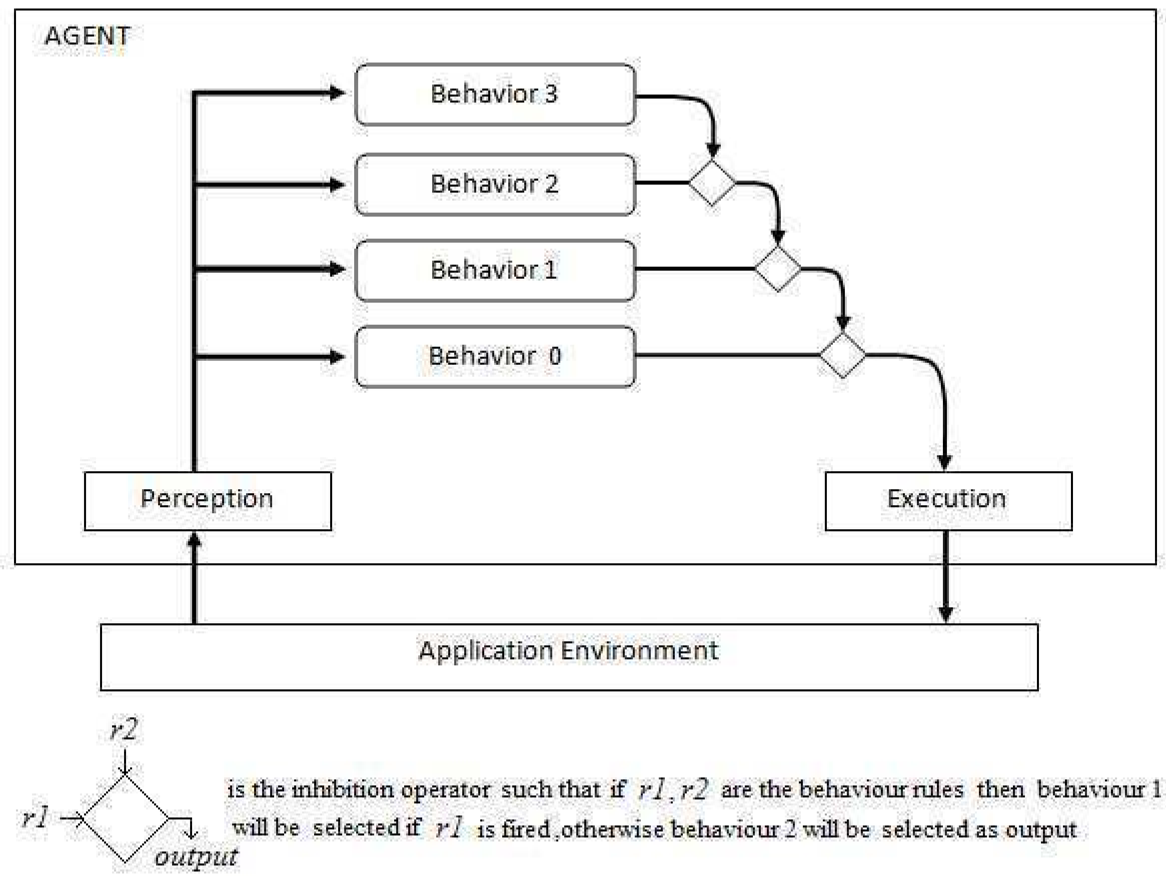


Figure 2.2: The Subsumption architecture.

- The complex intelligent behaviour emerges from the interaction of various simpler behaviours.

It is a modular architecture with horizontal linking between modules as shown in figure 2.2, here the modules are organised in vertical layers. The modules operate in parallel, with those higher up in the organisation having a dominance over those lower down. This means that the higher modules can inhibit the behaviour of lower level modules. As with the modular architecture, the designer defines the connections between modules and the dominance relationships that exist between them in the form of inhibit rules. Usually the implementations of inhibiting relationship is implemented by means of priority assignment for individual behaviours.

Algorithm 2: Pseudocode for action selection by a subsumption agent.

Data: percept $(p : P)$, action set A
Result: selected action $\hat{a}$
/*`get the list of all the rules for the percept` P */
initialise fired:$P(R)$;
/*`for the current percept` p `generate the list of fired rules` */
fired:=$\{(c, \hat{a})|(c, \hat{a}) \in R$ and $p \in c\}$;
/*`if fired rule list is not empty` */
if $|fired| > 0$ **then**
    /*`then select the highest priority rule` */
    find minimum $(c, \hat{a})$;
    /*`and return the action sequence associated with it` */
    return $\hat{a}$;
else
    /*`otherwise do nothing` */
    return *null*;
end

Here the agent's decision-making is realized through a set of task accomplishing behaviours. Each behaviour is like an individual action function, continually taking perceptual input and mapping it onto an action to perform. No complex symbolic representations and no symbolic reasoning are utilised at all. It mainly implements the rules to map *situation* $\rightarrow$ *action* relationship. The percepts block accepts input from sensors and produces a set of percepts P. The action is realised through a set of behaviour rules R, together with an inhibition relation $\prec$, over time where c is a set of percepts called that *condition* and $\hat{a}$ is an action.

$$R = \{(c, \hat{a})|C \subseteq P, \hat{a} \in A\} \tag{2.1}$$

A behaviour will fire in state s if some function $see(s) \in c$ (if the condition is satisfied by the percepts). The inhibition relation is a total ordering on the behaviour rules. If $r1$ inhibits $r2$, then the inhibit rule can be written as $r1 \prec r2$, i.e., $r1$ is lower in the hierarchy than $r2$ and hence will get priority over $r2$, where $r1, r2, ...$ are the rules (elements of R). The algorithm for action identification for such an agent is given in Algorithm 2.

The subsumption architecture has been successfully used in robotic applications e.g. AGVs (automated guided vehicles)[126] and in control of Intelligent Geometry Compressor [76, 77] and many more, proving its significance. Reactive architectures like subsumption, on one hand are advantageous for being modular, supportive towards iterative development and testing of real-time systems in their target domain, and emphasising on connecting limited, task-specific perception directly to the expressed actions that require it. But, on the other hand, they suffer from drawbacks of inability to have many layers, since the goals begin interfering with each other, the difficulty of designing action selection through highly distributed system of inhibition and suppression, thus providing rather low flexibility at runtime.

2.2.3 Logic based agent architectures

Logic based architecture discussed by Genesereth and Nilsson [30] and others[62, 69, 95] realises the agents using the traditional approach to building artificially intelligent systems. It is based on the notion that intelligent behaviour can be generated in a system by giving that system a symbolic representation of its environment and its desired behaviour and syntactically manipulating this representation. This syntactic manipulation corresponds to logical deduction. In logic based architectures, the system's symbolic representation is stored as a database of formulae of classical first-order predicate logic similar to a *Prolog* [20] database. This database mimics the information that agents have about their environment and plays a somewhat analogous role to that of belief in humans. The agent behaviour is mainly decided by the *deduction rules*.

Algorithm 3: Pseudocode for action selection by a logic based agent.

Data: system state Δ, logic sentences set L, L-formulae set D, action set A
Result: selected action $\hat{a}$

```
/*get the deduction rules ρ from the Knowledge-Base K */
initialise get ρ, ;
/*for each action â element of set A */
for each a ∈ A do
    /*if from the internal state Δ the formula ℘(â) can be proved
      using deduction rule ρ */
    if Δ⊢_ρ ℘(â) then
        /*then return action â as the valid action */
        return â ;
    end
end
/*for each action â element of set A */
for each a ∈ A do
    /*if from the internal state Δ the formula ¬℘(â) can not be
      proved using deduction rule ρ */
    if Δ⊬_ρ ¬℘(â) then
        /*then return action â as the valid action */
        return â ;
    end
end
/*otherwise do nothing */
return null ;
```

Let L be the set of sentences of classic first-order logic and $D = \wp(L)$ be the set of sets of L-formulae. Now if $\Delta, \Delta_1, \Delta_2, ...$ are the members of D representing the agent's internal state, then the agent's decision making process is modelled through a set of deduction rules denoted by ρ. The choice of action by the agent is decided through proving the formula $\wp(\hat{a})$ by theorem proving process based on the present database representing the system state. i.e. if $\Delta \vdash_\rho \wp(\hat{a})$ or $\Delta \nvdash_\rho \neg\wp(\hat{a})$ can be proven from an internal state Δ using only the deduction rule ρ then the action $\hat{a}$ is the valid action for applying to the system. The algorithm for action identification for such an agent is given in Algorithm 3.

Along with the logic based action selection mechanism, the agent will have the normal blocks for mapping the functions from sensors to the percepts in a more elaborative approach through symbolic representation. Thus the complete execution cycle will be given by.

$see : S \rightarrow P$

$next : D \times P \rightarrow D$

$action : D \rightarrow A$ where the functions *see*, *next*, *action* are the basic function of abstract agent.

Using logic based approach, the agent's behaviour can be guaranteed. This may be useful for safety-critical applications or applications of high priority such as tackling those parts of accelerator control which can directly result into beam killing by the immediate wrong agent actions rather than simply degrading the operation performance. The theorem proving process takes time and by the time the agent proves which action is optimal, the environment may have changed. This leads to the problem of *calculative rationality* (i.e. the decision making apparatus produces action that was optimal when decision making process began). Therefore this type of architecture is only suitable when environment doesn't change faster than the agent can make decisions. Also, many times, the representation of procedural knowledge and reasoning about temporal information is not easy for implementation point of view.

2.2.4 Deliberative (Beliefs-Desire-Intention) architecture

BDI agent model presents the abstraction of rational agent based on the notion of belief-desire-intention, inspired by the human practical reasoning and decision process i.e. the process of deciding, moment by moment which action to perform in the furtherance of desired goals. For this, the reasoning by humans involves two important phases: (1) what goals a human wants to achieve and (2) how he is going to achieve them. The first

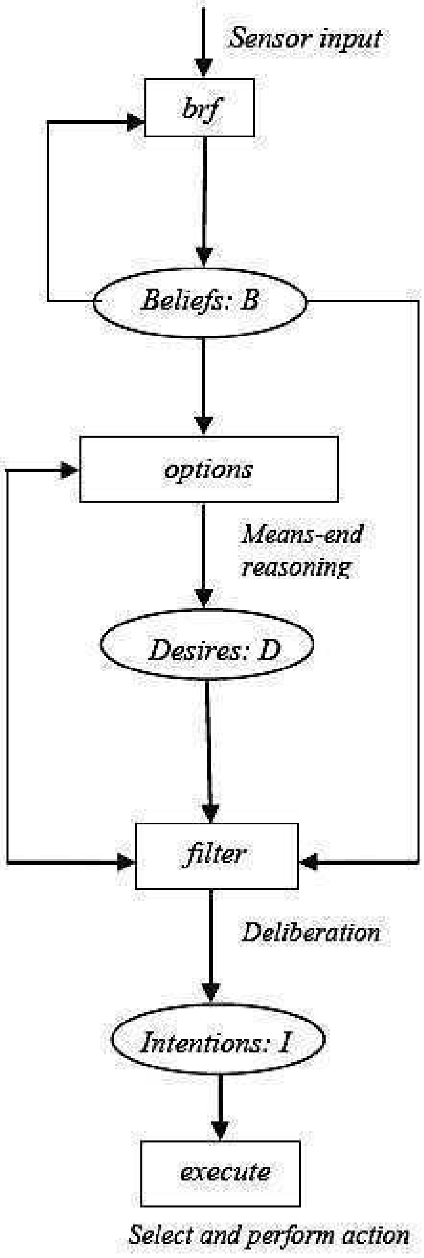


Figure 2.3: The BDI agent abstract architecture.

is also called the *deliberation* process and the latter is called $means - end\ reasoning$ [12, 13, 31, 98, 99]. The architecture of BDI agent is outlined in the block diagram in figure 2.3. There are seven main components in a BDI agent.

$Note: -$ Let Bel, Des and Int denote large abstract sets from which beliefs, desires and intentions can be taken. The state of a BDI agent is at any moment a triple $\langle B, D, I\rangle$ where $B \subseteq Bel$, $D \subseteq Des$ and $I \subseteq Int$.

1. A set of beliefs representing information the agent has about its current environment $\wp(Bel)$.

2. A belief revision function, (brf), which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs. Therefore it is a mapping from a belief set and percept into a new belief set

   $brf : \wp(Bel) \times P \longrightarrow \wp(Bel)$

3. An option generation function, $(options)$, which determines the options available to the agent $(its\ desires)$, on the basis of its current beliefs about its environment and its current intentions; thus it maps a set of beliefs and a set of intentions to a set of desires.

   $options : \wp(Bel) \times \wp(Iint) \longrightarrow \wp(Des)$

   Here, the main function of options is $means - end\ reasoning$, and this must be consistent with beliefs and current intentions as well as *opportunistic* to recognise when environmental circumstances change advantageously.

4. A set of current options, representing possible courses of actions available to the agent.

5. A filter function $(filter)$, which represents the agent's deliberation process, and determines the agent's intentions on the basis of its current beliefs, desires, and intentions.

   $filter : \wp(Bel) \times \wp(Des) \times \wp(Iint) \longrightarrow \wp(Int)$

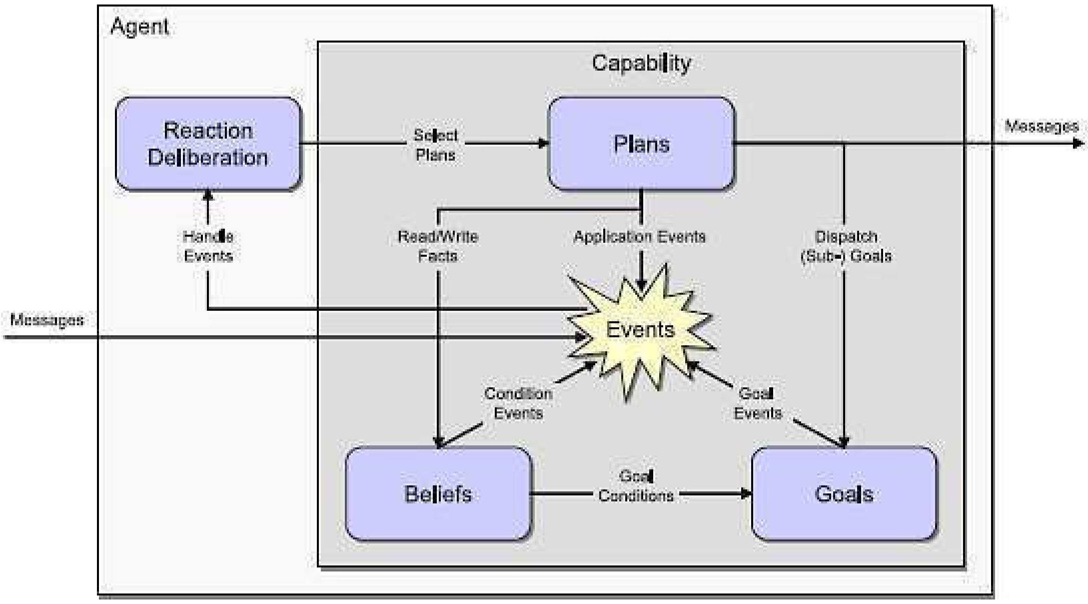


Figure 2.4: The BDI agent implementation block diagram [66].

It must drop intentions that are no longer achievable, retain intentions that are not yet achieved and it should adopt new intentions to achieve existing intentions or to exploit new opportunities. A constraint on filter is that it must satisfy $filter(B, D, I) \subseteq I$, i.e. current intentions must be either previously held intentions or newly adopted ones.

6. A set of current intentions, representing the agent's current focus i.e. the goals it has committed to trying to bring about.

7. An action selection function ($execute$), determines an action to perform on the basis of current intentions.

$$execute : \wp(Int) \longrightarrow A$$

Desire and intention are the mental attitudes concerned with the actions and beliefs representing the agent's information and knowledge about its environment.

Algorithm 4: Standard BDI interpreter.

```
Data: Sets beliefs B, Desires D, Intentions I
Result: perform action()
initialise internal state ;
repeat
    /*observe environment for events */
    READ event − queue ;
    /*and generate options */
    options ← OPTION-GENERATE(event − queue) ;
    /* select valid options */
    selected − options ← DELIBERATE(options);
    /* add the selected options to intentions list */
    UPDATE-INTENTIONS(selected − options);
    /* execute one action execution cycle */
    EXECUTE(intentions);
    /* get new external events */
    READ event − queue ;
    /* remove the sucessfull attitudes */
    DROP(successful-attitudes);
    /* remove the impossible attitudes */
    DROP(impossible-attitudes);
until stop;
```

In BDI agent implementations the abstraction of desire is represented by the set of goals for the agent. An active goal can be considered as the desire that has been adopted for active pursuit by the agent. For achieving the goals, the agent implementation construct provides mechanism for defining different plans. A plan is an ordered list of actions which must be performed for achieving the goal. The intentions represent the current plans the agent has chosen to execute or is currently executing. Figure 2.4 shows the block diagram of a BDI agent implementation [66]. The algorithm 4 shows the classical BDI interpreter proposed by A. S.Rao and M.P.Georgeff [91].

At the beginning of each cycle, the option generator reads the event queue and returns the list of options. Options are basically the goals that the agent perceives viable for consideration at an instant of time. From implementation point of view, options can be viewed as the list of all the alternate plans that are meant for achieving a particular goal. Each plan comprises of a body that describes the sequence of actions or sub-goals that have to be achieved for plan execution to be successful. The conditions under which a plan can be chosen as an option are specified by an initiation condition (triggering events) and a precondition, specifying the situation that must hold for the plan to be executable. Next, the deliberate procedure selects from the options list those plans which are found suitable for immediate execution and lists them as selected-options. The update-intentions procedure then updates the agent's beliefs for the immediately selected intentions. The execute procedure caries out the actual work of acting upon the environment towards achieving the intended goal by sequentially executing the recipe specified in the body of the selected plan. The next three procedures get-new-external-events, drop-successful-attitudes and drop-impossible-attitudes basically update the event queue with the system generated events and external events and filter out the un-relevant goals before proceeding for the next interpreter cycle.

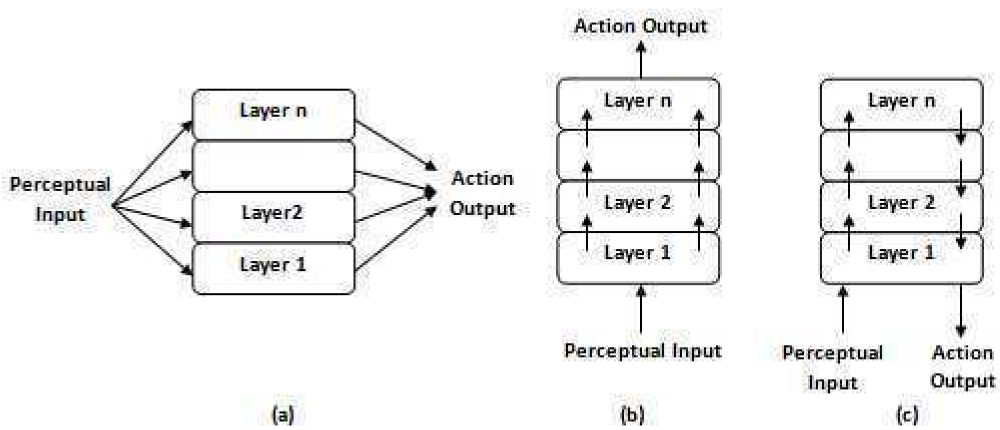


Figure 2.5: The Layered agent architecture (a) horizontal layering, (b) vertical layering with one pass and (c) vertical layering with two pass.

2.2.5 Hybrid architectures

Hybrid architectures also called as hierarchical architectures [28, 73] attempt to combine both reactive architecture and deliberation architecture in order to take advantages of both. As a result such architectures are capable of providing fast reactive behaviour as well as more intuitive cognitive behaviour by incorporating the learning and sustained, pro-active, goal directed behaviour. In the layered architecture, the various subsystems of the agent capable of providing reactive and pro-active behaviours are arranged in a layered structure.

Often, the reactive component is given some kind of precedence over the deliberative one, so that it can provide a rapid response to important environmental events. The Touring Machines [28], INTERRAP [73, 74], and CIRCA [72] are good examples of such architectures. Normally in an agent body subsystems are arranged into a hierarchy, with higher layers dealing with information at increasing levels of abstraction. Typically, there will be at least two layers, to deal with reactive and pro-active behaviours respectively.

Two types of control flow mechanisms are normally used within layered architectures

1. Horizontal Layering,

2. Vertical Layering.

Horizontal Layering

In horizontally layered architectures, the software layers are each directly connected to the sensory input and action output. This is shown in figure 2.5(a) This approach is simple and implements agent by implementing n layers, for n different types of behaviour. To overcome the problem of interfering of one layer's action with that of other layer, when layers compete with one another to generate action suggestions, central control is exercised through mediator function, where mediator decides about the layer to which the task of control is given. This forms a bottleneck when the number of layers increases as the mediator function has to consider all possible interactions between layers before deciding about the layer of the agent that should take control.

Vertical Layering

In vertically layered architectures, (figure 2.5(b),(c)), sensory input and action output are each dealt with by at most one layer each. This type of architectures are classified into one pass architectures and two pass architectures. In one-pass architectures, control flows sequentially through each layer, until the final layer generates action output. In two pass architectures, information flows up the architecture and control then flows back down. Both these architectures reduce the complexity of interaction between layers thus relaxing the control bottleneck problem faced in the horizontal layering but at the same time suffers from the problem of fault intolerance, as failure in any one layer can seriously affect the agent performance.

2.2.6 Modular architecture

This type of architecture is widely used in multi-agent systems and may range from very simple, comprising a few modules to complex organisations involving a large number of modules. It is sometimes referred to as a horizontal-module architecture since the modules are at the same level in the organisation. Also, in this type of architecture, all of the connections between the modules are typically fixed i.e. the information flow is pre-defined by the agent designer. The simple example shown below reveals the basic perception, cognition and execution structure. Figure 2.6 shows the different modules of such an agent system. A few examples of modular architecture are robotic applications[51], network diagnostics [3] and in service agents[78].

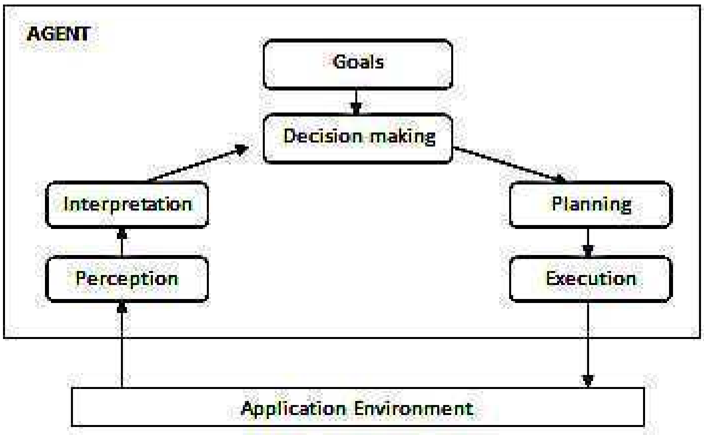


Figure 2.6: The modular agent architecture.

2.3 Multi agent systems

The agent based problem solution and exercising of system control is extended further by designing the system comprising of multiple agents where each agent is directed towards achieving its local goal and contributes its share in achieving the common goal of the community. The field is covered under the concept of Distributed Artificial Intelligence (DAI) where the agents are grouped together to form communities which cooperate and interact to achieve the goals of individuals and of the system as a whole. Further, it is assumed that each agent is capable of a range of useful problem-solving activities in its own right, has its own aims and objectives and can communicate with others [41]. The multi-agent based approach requires the enhancement of individual agent behaviours/capabilities irrespective of the way they are implemented and of the method/methodology adopted by underlying agent implementation for participating as an individual in an multi-agent system. Some of these capabilities are communication, coordination, cooperation and collaboration as discussed in the following.

Communication

In multi-agent systems the mechanism to communicate between different agents working as a team is a must for information exchange. This information exchange can be accomplished through environment in an indirect way in which one of the agent modifies the environment with an intention that following agents will capture the purposefully introduced disturbance in the environment and will infer the message to be conveyed. This type of primitive biologically inspired information exchange is of limited use but is an important trick of communication through behaviour and mainly employed for physical agents like robots with very simple structures and limited resources [29]. The direct communication mechanism is used when the agents are situated in an networked environment. The two main types of communication methods of this type are *shared memory* and *message passing*. The most widespread example of the former is the blackboard system [26, 27]. where the blackboard is a global database containing entries generated by the agents. The entries include intermediate results generated during problem solving and include both elements of the problem solution and information deemed important in generating solution elements. Message passing ideas have been drawn from conventional object-oriented programming and in particular from object-based concurrent programming and is the widely adopted approach for inter-agent communication [42, 53, 119]. Message passing has some advantages over the blackboard system. In particular, shared memory systems generally do not scale up well - a single blackboard can be a severe bottleneck and multiple blackboards have the same semantics as message passing systems.

At the extreme of sophistication is the use of formal languages involving extended exchange of series of messages to support a conversation between agents [117]. In this context there is much, and growing interest in the field of ontology as a possible mechanism for agents to share the meaning of exchanged symbols [105].

Co-ordination

Co-ordination may be regarded as the process by which an agent reasons about its local actions and the (anticipated) actions of others to try and ensure that the community acts in a coherent manner, to achieve the overall goals of the system. Without coordination, the benefit of decentralized problem solving vanishes and the community may quickly degenerate into a collection of chaotic, in-cohesive individuals. Further, the coordination process ensures that all necessary portions of the overall problem are included in the activities of at least one agent. Specific examples of coordination activities include supplying timely information to needy agents, ensuring the actions of multiple actors are synchronized and avoiding redundant problem solving [67]. Various co-ordination techniques have been devised like organisational structuring, subcontracting, negotiation and multi-agent planning. Basically all of these techniques use the following mechanisms for co-ordination at one stage or the other in their operation cycle.

- *Mutual adjustment* - agents share information and resources to achieve some common goal by adjusting their behaviour according to the behaviour of the other agents,

- *Direct supervision* - one agent has some degree of control over others which may have been arrived at through mutual adjustment,

- *Standardization* - supervisor agent establishes standard procedures for agents to follow in given situations,

- *Mediation* - one agent serves as a facilitator or broker to influence interaction between agents.

Collaboration

Collaboration arises when one agent is able to perform a task, which only it can do, and as a result enables another agent to achieve its own goal. Clearly, the need for collaboration is determined by the allocation of skills and resources to agents made when the system was designed. Except in simple cases, it is often necessary to coordinate collaboration in order to make effective use of skills and resources consistent with overall system goals. The process of deduction which agents have to collaborate for the given goal is governed by the coalition formation strategies. A number of coalition formation algorithms have been developed to determine which of the potential coalitions should actually be formed [23, 55, 83, 100]. All of these coalition formation processes include three main activities [101].

- *Coalition structure generation*- partitioning the set of agents into exhaustive and disjoint coalitions. Such a partition is called a coalition structure.
- *Optimizing the value of each coalition*- pooling the tasks and resources of the agents in every coalition, in the coalition structure, in order to maximize the coalition value.
- *Payoff distribution*- dividing the value of each coalition among its members so as to achieve stability or fairness.

Co-operation

Co-operation is about agent's actions being mutually supportive to their respective goals. Supportive action by one agent for another may be intentional or incidental. Ferber [29] states that, put simply, the problem of co-operation condenses down to determining who does what, when, by what means, in what way and with whom. Ferber summarises this in the formula:

Co-operation = collaboration + co-ordination of actions + resolution of conflicts

Ultimately, the co-operation strategy of a multi-agent system is critical in ensuring that actions by autonomous agents in pursuit of local goals have, at least, a beneficial, if not optimal, effect on overall system performance.

2.4 Intelligent control in accelerator scenario

In accelerator community the first use of agent application was demonstrated by Jennings [41]. Jennings connected the two diagnostic expert systems in the accelerator control environment namely CODES(control system diagnosis expert system) and BEDES (beam diagnosis expert system) by using the ARCHON framework, thus converting them to two distinct intelligent agents, which can work together to fulfil common goal of diagnosing the faults. The ARCHON frame work was composed of four main components : a high-level communication module (HLCM), which manages inter–agent communication; a planning and coordination module (PCM), which is essentially responsible for deciding what the agent will do; an agent information management module (AIM), which is responsible for maintaining the agent's model of the world and finally, an underlying intelligent system (IS), which represents the agent's domain expertise. The HLCM, PCM and AIM together constitute a kind of 'agent wrapper', which was used to encapsulate the existing intelligent system to turn them into agents. Strengths of ARCHON architecture were: use of a top down approach to look at the overall needs of the application and a bottom up approach to look at the capabilities of the existing system so that the development efforts can be minimised. Problem solution was achieved through pre-specified recipes called as plans. Plans were represented in the form of tree with nodes as tasks and arcs as conditions. The self model and the acquaintance model were provided for assistance. Apart from accelerator fault finding, the ARCHON architecture has been successfully applied to different other industrial applications [40, 44, 111, 119, 120] also primarily for assisting in the fault finding operation.

Another effort towards the intelligent control of accelerator was done by Klein et.al. [56, 57, 58, 59, 60, 113]. They proposed a control system architecture based on hierarchical, distributed, knowledge-based controllers, each of which is an expert in controlling some section of the environment or in performing some function over that environment. These Knowledge-based controllers were provided with the capabilities for planning, diagnosis, and learning, as well as knowledge acquired from human domain experts to select, sequence, and configure control actions. Controllers were also designated with the responsibility for reasoning about the system state, diagnosing errors in controller actions, decomposing goals into tasks and actions, and initiating human intervention as and when found necessary.

Means for implementing the general purpose optimization or control algorithms, such as hill-climbing optimization, fuzzy logic or neural network-based feedback control, conventional control loops, etc were provided through services called as solvers. More complex procedures were formulated by coordinated assembly of individual solvers. An object oriented physical access layer (PAL) was developed as an abstraction mechanism between controllers and the underlying control system to provide a mechanism for hiding unimportant implementation details about the domain hardware and provide a uniform interface for control access.

A new control approach called as teleo-reactive(TR)[81] control was utilised. Which is a combination of feedback-based control and traditional discrete action planning. TR programs sequence the execution of actions that have been assembled into a goal-related plan. Here, unlike traditional planning environments, no assumption is made that actions are discrete and un-interruptible and that an action's effects are completely predictable. On the contrary, teleo-actions are executed as long as the action's preconditions hold and its goal has not yet been achieved (unless some other action closer to the final goal becomes activated). A short sense-react cycle ensures that when the environment changes, the control action changes to fit the new state. The strongest point of this approach is that the TR tree execution is adaptive in that, if some unanticipated event in the environment reverses the effects of previous actions, TR execution will typically fall back to some lower level condition and restart its work towards the top level goal. On the other hand, if something good happens, TR execution is opportunistic i.e. when some higher condition unexpectedly becomes true, execution shifts to the action associated with that condition.

In addition to the above stated accelerator control efforts, there are two more examples which although do not come under intelligent agent based control but need their mentioning as these are related to the scope of this thesis. The first is related to the application of intelligent system concepts to automatic beamline alignment problem. Pugliese et.al.[90] proposed a model reference based architecture by combining the hierarchical and subsumption architecture. The strength of this approach is the coupling of an artificial intelligence and a real-time sub-system as parallel, cooperating components. The second is related to the electron transport line optimization using neural networks and genetic algorithms, where Schiemer et.al.[102] proposed the accelerator (particularly the transport line) tuning using the artificial intelligence based scheme.

ARCHON architecture although talks about the use of self model but lacks the ability of adaptation mechanism in it which may be required for achieving the close loop control objective. Further to this, the architecture addressed all the system parameter setting operations/system parameter reading operations through the AL-IS Interface (which is acceptable for the diagnostic applications and is a must to make the application platform independent) but for implementation of local closed loop control, the methodology for direct transactions with system inputs/outputs can serve as a better alternative.

The TR based architecture proposed by Klein et.al. solves the lack of adaptation and I/O operation through intermediate layer problem of ARCHON and is thus more suited for the accelerator tuning and control application. But it experiences a problem of execution failure of TR tree under certain conditions, for example, if the TR tree enters a state in which no node is active, or if the TR tree is stuck in a cycle involving two or more nodes, or if the tree is stuck in a cycle involving a single node. The first case arises because the control plan is incomplete and does not include an action responding to some unexpected state. In the second case some node repeatedly terminates execution before its parent node becomes active (because it makes its own pre-image condition false). In the third case a single node repeatedly executes without making its parent active. The solution to these requires automatic derivation of action models from accelerator models. Further to this, this approach requires the intelligence at the controller level which may be possible to achieve for new machines to be developed, or for machines which are under the process of up-gradation, whereas it may not always be possible to incorporate the changes at controller levels for the existing machines.

2.4.1 ARCHON framework

ARCHON stands for ARchitecture for Cooperative Heterogeneous ON-line systems. It devised a general-purpose architecture, software framework, and methodology which has been used to support the development of distributed artificial intelligence (DAI) systems in a number of real world industrial domains [40, 44, 111, 119, 120]. The ARCHON framework focusses on the real implementation issues rather than the symbolic representation and semantics of the agent based control problem. The methodology is mainly focussed on the distributed problem solving using multi agent scenarios and tries to address both types of real word systems, purpose-built and pre-existing systems, where ARCHON basically acts as the gluing platform which joins the distributed intelligent systems (IS) capable of exhibiting some semi-autonomous behaviour and extends their ability towards joint problem solving for common goal. In ARCHON, individual problem solving entities are called agents. These agents have the ability to control their own problem solving and to interact with other community members. The interactions typically involve agents cooperating and communicating with one another in order to enhance their individual problem solving and to better solve the overall application problem. Each agent consists of an ARCHON Layer (AL) and an application program (IS) as shown in the figure 2.7.

The ARCHON's modular and layered implementation architecture comprises of mainly four modules - Monitor, Planning and Coordination Module (PCM), Agent Information Management (AIM) module, and High Level Communication Module.

Monitor

The Monitor is responsible for controlling the local IS. Each IS task is represented in the Monitor by a monitoring unit (MU). MUs present a standard interface to the Monitor whatever the host programming language and hardware platform of the underlying IS could be. These MUs can send and receive messages (of type *directives*, *confirmations* and *requests*) to and from the IS. All messages pass through the AL-IS interface which performs the translation and interpretation required for the IS to understand the AL directives and for the AL to understand the IS messages. For the IS to be able to react to an AL directive, the interface translates the command into the corresponding local control action. However, the interpretation and implementation of commands at IS are

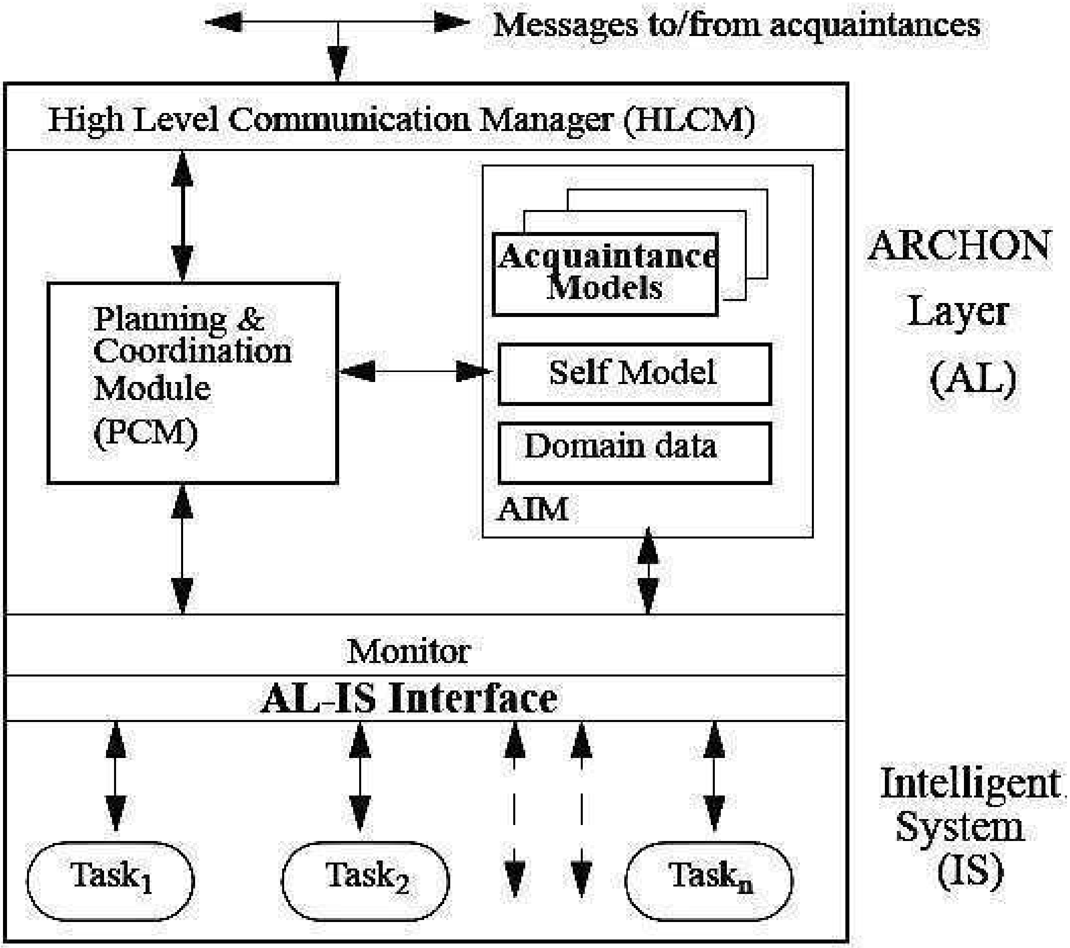


Figure 2.7: ARCHON agent architecture[44].

left to the IS implementer domain.

MUs represent the finest level of control in the AL, and at the next level of granularity there are plans. Plans are pre-specified, acyclic, OR-graphs, in which, the nodes are MUs and the arcs are conditions. These conditions can: be dependent on data already available from previously executed MUs in the plan, be dependent on data input to the plan when it started, make use of the locking mechanism for critical sections of the plan, or be used to return intermediate results before a plan has completed. The plan mechanism is provided with inbuilt backtracking facility which can be used to express preferences and deal with complex alternatives.

The highest level at which the IS's activities are represented is the *behaviour* level.

Behaviours contain a plan, a trigger condition for activating the behaviour, descriptions of the inputs needed by the activity and the results which will be produced, and any children of the behaviour. There are two types of behaviour: those that are visible to the PCM (and the other AL components) and those that are purely internal to the Monitor The former type are called *skills* and they may be triggered by new data (either arriving from other agents or which the agent has generated itself) or by direct requests from other agents.

Planning and Coordination module

The PCM is the reflective part of the AL, reasoning about the agent's role in terms of the wider cooperating community [43]. It is composed of generic rules about cooperation and situation assessment which are applicable in all industrial applications - all the domain specific information needed to define individual behaviour is stored in the self and acquaintance models. The former contains information about the local IS and the latter contains information about the other agents in the system with which the modelling agent will interact. The type of information contained in both models is approximately the same, although it varies in the level of detail, and includes the agent's skills, interests, current status, workload and so on. For example, in order to determine how to obtain information which is needed to execute a behaviour but which is not currently available, the PCM will make reference to its self model to see if the information can be provided locally by executing an appropriate skill. If the information cannot be provided locally then the acquaintance models are checked to see if another community member can provide it. When the Monitor gets some results from a behaviour, firstly, the PCM checks the self model to see if the data can be used locally and then it examines its acquaintance models to see if any other agents are believed to be interested in receiving the data. And finally the PCM deals with requests arriving from other agents. By reference to its self model, it will decide whether to honour the request and will then activate the necessary skill to provide the requested data. When the information is available it will ensure that a reply is directed to the source of the request.

Agent information management module

The AIM module is a distributed object management system which was designed to provide information management services to cooperating agents [116]. Within ARCHON, it is used to store both the agent models and the domain level data. As an illustration of the agent models, consider an agent, which is capable of producing information about ALARM-MESSAGES. The interest slots of its acquaintance models contain those agents who are interested in receiving this information and the conditions under which they are interested (a null condition signifies in all cases). The following portion of the acquaintance model specifies that an agent called BRS is interested in ALARM-MESSAGES which contain chronological information, that an agent called AAA is interested only in non-chronological alarm messages, and that an agent called BAI is only interested in non-chronological alarm messages which have the string INT within their ALARMS field:

```
INTEREST-DESCRIPTOR
    INFORMATION-NAME: ALARM-MESSAGES
    INFORMATION-CONDITION:
        ["BRS",   (CONTAIN    (ALARM-MESSAGES "CHRONOLOGICAL "YES""));
        ("AAA",   (CONTAIN    (ALARM-MESSAGES "CHRONOLOGICAL "NO""));
        ("BAI",   (AND   (CONTAIN (ALARM-MESSAGES "CHRONOLOGICAL "NO""))
                         (CONTAIN (ALARM-MESSAGES.ALARMS "INT"))));]
```

High Level Communication Module

The High Level Communication Module (HLCM) allows agents to communicate with one another using services based on the TCP/IP protocol. The HLCM incorporates the functionality of the ISO/OSI Session Layer which continuously checks communication links and provides automatic recovery of connection breaks when possible. Information can be sent to named agents or to relevant agents (decided by reference to interests registered in the acquaintance models).

The ARCHON architecture has been used to integrate a wide variety of application program types under the general assumption that the ensuing agents will be loosely coupled and semi-autonomous. The agents are loosely coupled since the number of interdependencies between their respective ISs are kept to a minimum; the agents are semi-autonomous since their control regime is decentralised (meaning each individual ultimately decides which tasks to execute in which order). The ISs themselves can be heterogeneous - in terms of their programming language, their algorithm, their problem solving paradigm, and their hardware platform. An AL views its IS in a purely functional manner, it expects to invoke functions (tasks) which return results, and there is a fixed language for managing this interaction.

There is no centrally located global authority and each agent controls its own IS and mediates its own interactions with other agents. The system's overall objectives are expressed in the separate local goals of each agent in the agent community. Because the agent's goals are often interrelated, social interactions are required to meet global constraints and to provide the necessary services and information. Such interactions are controlled by the agent's AL; relevant examples include:asking for information from acquaintances, requesting processing services from acquaintances, and spontaneously volunteering information which is believed to be relevant to others.

In more detail, an agent's AL controls tasks within its local IS, decide when to interact with other agents (for which it needs to model the capabilities of its own IS and the ISs of the other agents), and communicate with its acquaintances. It is these basic functionalities that ARCHON's modular and layered architecture provides to the designer to help him transform the existing systems to intelligent agent.

2.4.2 Teleo-reactive (TR)control

Nils Nilsson proposed the Teleo-reactive agents [81] based on the notion of teleo-reactive programming for planning and action representations. Teleo-reactive agents [9, 50, 82] react to their perceptions of the world by obeying an internal program (or policy) mapping perceptions to actions. The simplest policy structure is a set of mutually-exclusive production rules of the form *perception* → *action*, usually intended to control durative behaviour: given some current perception the agent performs the corresponding action until acquiring a new perception, whereupon it reacts likewise to that. Such an agent may or may not have sufficient perceptive capability to know, at any instant, the entire state of the world. An agent of this kind presumes that, it is capable of perceiving an intended goal state, whenever that state arises, and is accordingly designed with that capability in mind. Its program includes an explicit test for the goal state, whilst the nature and ordering of its rules are inferred by reductive analysis of that test. Its goal-orientedness is thus explicit in the program.

TR control occupies a region between feedback-based control and traditional discrete action planning. TR programs sequence the execution of actions that have been assembled into a certain kind of goal-related plan. Unlike traditional planning environments, no assumption is made that actions are discrete and uninterruptible and that an action's effects are completely predictable. On the contrary teleo-actions are typically durative and are executed as long as the action's preconditions hold and its goal has not yet been achieved (unless some other action closer to the final goal becomes activated)[41]. A short sense-react cycle ensures that when the environment changes, the control action changes to fit the new state.

TR action sequences or plans are represented in a data structure called a TR tree. A TR tree can be described as a set of condition-action pairs:

$$
\begin{aligned}
&c_0 \rightarrow \hat{a}_0 \\
&c_1 \rightarrow \hat{a}_1 \\
&\quad \ldots \\
&c_n \rightarrow \hat{a}_n
\end{aligned}
$$

where c_s are conditions and $\hat{a}_s$ are the associated actions. c_0 is typically the top level goal of the tree and $\hat{a}_0$ is the null action, i.e., do nothing if the goal is achieved. At each execution cycle the c_i are evaluated from top to bottom until the first true condition is found. The associated action $\hat{a}_i$ is then performed. The evaluation cycle is then repeated at a frequency that simulates the reactivity of circuit based control. TR trees are constructed so that each action $\hat{a}_k$, if continuously executed under normal conditions, will eventually make some condition higher in the tree true. This then ensures that under normal conditions the top level goal, c_0, will eventually become true. TR tree execution is adaptive in that if some unanticipated event in the environment reverses the effects of previous actions, TR execution will typically fall back to some lower level condition and restart its work towards the top level goal. On the other hand, if something good happens, TR execution is opportunistic: when some higher condition unexpectedly becomes true, execution shifts to the action associated with that condition. Figure 2.8 represents such a typical TR tree showing the relationships between goals, actions, and sub-goals. At each execution cycle, the action associated with the highest true condition in the tree is selected for execution. when the highest active level contains more than one action with satisfied preconditions, some arbitrary probabilistic method is used for choosing between possible actions [81].

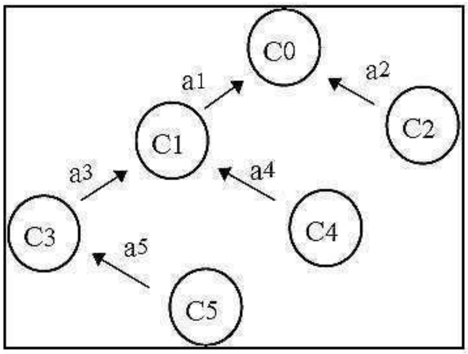


Figure 2.8: teleo-reactive tree graph[113].

Construction of TR trees can be accomplished through a planning algorithm. Starting from the top level goal, the planner searches over actions whose effect includes achievement of the goal. The preconditions of the action generate a new set of subgoals, and the procedure recurses. Termination is achieved when the preconditions of one of the leaf nodes of the tree is satisfied by the current state of the environment. That is, the planning algorithm regresses from the top level goal through goal reduction to the current state. Actions, of course, generally have side effects, and the planner must be careful to verify that an action at any level does not alter conditions that are required as preconditions of actions at a higher level. TR tree planning algorithms typically build plans whose leaf nodes are satisfied by the current state of the environment. They do not build complete plans, that is, plans that can start from any world state, because such plans would generally be too large to efficiently store and execute. This is an important point because sometimes an unexpected environmental event can shift the world to a state in which no action preconditions in a TR tree are satisfied. In such case, replanning is done.

2.5 Summary and conclusion

In this chapter the concept of intelligent object *agent* is introduced. From literature the important agent properties that distinguish agents from simple computer programs emphasised by researchers are discussed. Starting from the abstract agent representations, the important agent implementation architectures along with their merits and demerits are presented. Extending the agent based system scenario, the important parameters necessary for multi-agent based system implementation are introduced. The literature survey on the use of agent based technology and the intelligent control concepts in the context of particle accelerators is then presented with an extended discussion on the case study of ARCHON framework and teleo-reactive control.

In chapter 4 we will see that the accelerator control system is comprised of distributed and layered structure; thus for accelerator control environment the subsumption architecture being simple and fast, can serve as the best candidate for implementations at the lowermost layers, that, demands a fast reactive cycle but are having the limited computational power. Further, in the implementation of agents for such controllers, the subsumption architecture can be extended with the communication capability already available with the accelerator control environment, thus embedding the coordination mechanism to such agents that enhances their integrity with the multi-agent based overall control of the facility.

The modular nature of the accelerator control system can be effectively utilised by using the modular architectures for agents that are needed to be implemented at middle or uppermost layers of accelerator control system. These agents can be implemented with the modules that either directly embed the underlying control system interface into their perception and execution modules or can simply communicate with the subsystems through message passing thus decreasing the implementation and testing time. Further to this, the concepts from BDI agent architecture could be mixed with the modular architecture for agent implementations with multi-level plan library for handling the systems dynamics.

The logical agent architecture could be utilised for implementing the agents at the higher layers where all the accelerator parameter information from different subsystems is available. These agents could assist the operators / underlying low complexity agents by providing the diagnostic information or by providing the guidance for deciding their actions for example the logical agents can inform to the local beam line tuning agents about the faulty device which it is using to correct the beam or it can provide the information about the effective pair of correctors for the particular accelerator operating condition that the agent can use for beam correction.

For accelerator environment on broader perspective, the overall accelerator control job can be subdivided among different agents of varying architecture that can be implemented on different accelerator control system layers of varying complexity and computation power. These agents can be logically integrated with sufficient granularity of work division and responsibility delegation to work as a single, multi-agent based control system that reduces the operator efforts in accelerator tuning and control job.

Chapter 3

Accelerator modelling

3.1 Introduction

Accelerators, specifically particle accelerators are the machines developed for acceleration of charged particles. All of them are based on the phenomenon of interaction of electric charge with static and dynamic electromagnetic fields. Based on the principle of using the electrostatic or oscillating electric fields for acceleration of charged particles, accelerators are classified into DC type accelerators and AC type accelerators. The CockcroftWalton generator and the Van de Graaf generator are the two popular examples of The DC accelerators. The AC accelerator, use oscillating electric field inside of a radio frequency cavity for creation of necessary accelerating electric field required for acceleration of the charged particles thus the acceleration process in an AC accelerator forces the condition that the particle should arrive/reach at the accelerating cavity at the right time. This synchronism condition expressed in terms of accelerating electric field phase is called as synchronous phase (ϕ_s). The acceleration produced during single pass through single cavity is not sufficient. Therefore for achieving higher energy, two types of concepts are utilized in AC accelerators:(1) When the charged particle is made to pass through a number of accelerating cavities placed one after another with suitable phase advance from cavity to cavity that satisfies the synchronous phase condition. Such accelerators where the charged particle is made to pass only once through one accelerating cavity is called linear accelerators. (2) when the charged particle is made to pass repeatedly many times through the same cavity or a group of cavities they are called as circular accelerators. Circular accelerator utilize the magnetic optics for confining and focusing of charge particle beams in a circular / closed loop paths so that the charged particle that are leaving the cavity are made to follow a closed loop path (circular /nearly circular) through magnetic optics to reach again at the cavity inlet. Depending upon the accelerator type, there are special mechanisms called as injection mechanisms / extraction mechanisms [6, 7] that are used for injection/ extraction of the charged particles into/from the closed ring. These mechanisms works on the principle of modifying the local magnetic optics for a very short time in a synchronized manner so that the orbit path is modified locally for accepting the incoming charged particle bunch from the nearby path / deflection of the charged particle bunch to the nearby path.

The synchrotron radiation sources [24] developed worldwide, use the special circular accelerators called as synchrotrons [68, 118] for particle acceleration and storage rings [68, 118] for storing the accelerated charged particles for a long time. Synchrotron particle accelerators are specific cyclic particle accelerators in which the guiding magnetic field (the magnetic field that bends and confines the charged particles into a closed path) increases in synchronism with the increasing particle energy in such a way that the closed beam path remains constant. Sometimes the synchrotron accelerator in a synchrotron radiation source facility is also called as booster. The storage ring is a special type of synchrotron accelerator in which the particle energy does not change. The purpose of storage ring is to store the accelerated particles (mainly electrons) for long duration of time. During this storage time, the charged particles are made to bend in dipoles and in insertion devices like undulator and wigglers where they lose energy in the form of synchrotron radiation. The storage rings are equipped with the RF cavity to compensate this energy loss by particles during each revolution period by accelerating them back to the same energy. It is this acceleration process because of which the storage rings come under the accelerator category. Further to this, most of the synchrotron radiation sources (for example INDUS-2, ELETTRA) employ the common booster cum storage ring in their final stage of acceleration, where the same ring first acts as synchrotron accelerator and accelerates the charge particles from injection energy to the top energy/ full energy through the process of ramping [2] and then switches its role from booster to storage ring and stores the particles for the remaining time till the beam current decays to the limiting value. During this storage time, the synchrotron radiation produced by dipoles and insertion devices is used at experimental stations connected to beam lines.

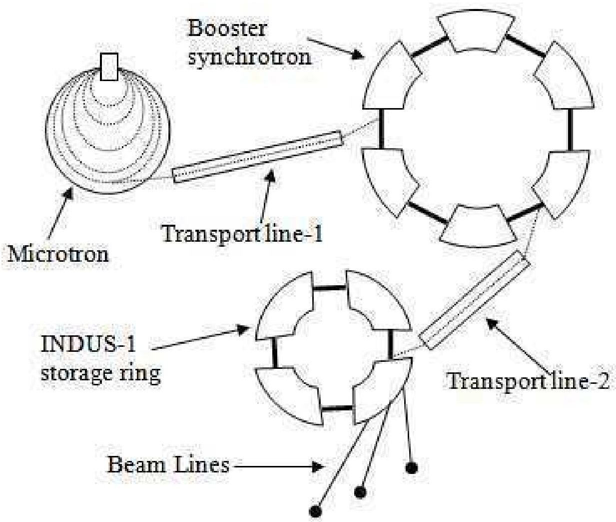


Figure 3.1: The Block diagram of INDUS-1 complex.

A typical synchrotron radiation source facility (like INDUS-1) comprises of many subsystems. Figure 3.1 shows the block diagram of INDUS-1 synchrotron radiation source facility highlighting different synchrotron radiation source components. The energy increase of order of few GeVs (0.45GeV in case of INDUS-1 and 2.5GeV in case of INDUS-2) for electron rings is achieved in many steps due to design constraints on physical devices like magnets, power supplies, RF cavities etc. The first accelerating unit which accelerates the electrons from ground energy level (zero energy) to some intermediate energy level (upto tens of MeV) is called as pre-injector. Linac and Microtron are the two accelerators mostly used as pre-injectors. In case of INDUS-1 and INDUS-2 Microtron accelerator is used as pre-injector. It accelerates the electrons from ground energy level to 20 MeV for injection to Booster. Booster is the synchrotron accelerator which accelerates the electrons further to next higher energy level (upto hundreds of MeV/GeV). In case of INDUS-1 and INDUS-2 facility, the booster is a synchrotron that accelerates the electron energy from 20MeV to 450/550 MeV. The final ring is either the booster-cum-storage ring or simple storage ring. In case of booster-cum-storage ring the electrons are accelerated to the final acceleration energy and then held at that energy for the duration of beam usage. In case of INDUS-1 the ring is a storage ring that stores the accelerated electrons at 450MeV. Whereas in the case of INDUS-2, the INDUS-2 ring is a booster-cum-storage ring and accelerates the electrons from 550MeV to final energy of 2.5GeV before entering to the storage mode of operation. The components named "Transport line-1"and "Transport line-2"in the figure 3.1 are the transport lines. The purpose of transport lines is to transport the charged particle beam from source to destination. Often the source and destination are the two different accelerators for accelerating the beam in different energy ranges. Since the transport lines join two accelerators for beam transport, they play the important role of matching the beam acceptance characteristics [68, 118] of destination to that of source.

The software simulation codes for accelerator design are used extensively by accelerator physicist at the time of designing the accelerator facility. The popular accelerator design and simulation codes are RACETRACK [122], MAD [70], ELEGANT [4] and AT [110]. Out of these, the MAD and AT are the two codes which can be used for accelerator component design (Transport Lines and synchrotron lattice) in a more flexible manner as the MAD is a command prompt based code that can accept input variables through files and can also output the result in the files in text mode. The AT is basically the toolbox extension to the MATLAB, thus providing the opportunity of using the MATLABs computing power and inbuilt function library to explore their use in simulation. In this thesis work, the accelerator component models using these codes are developed for building the overall integrated accelerator model that is used for analysing the scheme of intelligent object based control.

In the rest of this chapter the accelerator physics terminology and important accelerator physics parameters, accelerator components (lattice forming and measurement devices) for synchrotron, beam transport lines and beam storage rings from perspective of this thesis work are discussed. The individual transfer matrices of the individual magnetic components used to produce magnetic optics are discussed from accelerator simulation development point of view. Different accelerator simulation codes are discussed. The developed integrated models of different accelerator components using theses codes for use in simulation of accelerator control by intelligent objects are discussed.

3.2 Coordinate system

In circular accelerators the moving curvilinear coordinate system is used as shown in the figure 3.2. In this coordinate system the reference is taken from the moving particle such that the dimension x (horizontal) and dimension y (vertical) are perpendicular to each other and to the direction of motion of particle. In accelerator physics the motion of an arbitrary particle in this curvilinear coordinate system is described relative to the

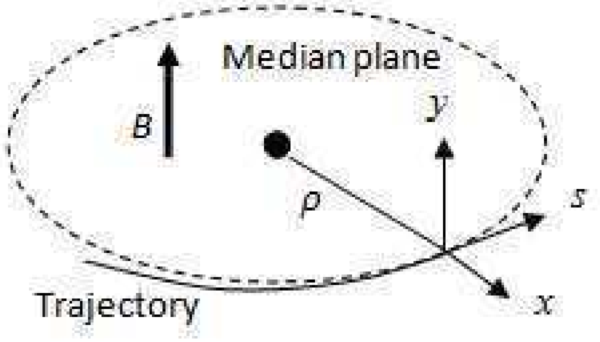


Figure 3.2: The moving curvilinear coordinate system used with circular accelerators.

ideal particle. An ideal particle is defined as the particle that has the ideal position, direction and energy at a certain instant of time. The path followed by an ideal particle is called the equilibrium orbit or the central orbit. In circular accelerators this orbit is normally confined to a plane, which is called the median plane. In this coordinate system an arbitrary particle has six degrees of freedom: $x, x'(=\frac{dx}{ds}), y, y'(=\frac{dy}{ds})), s, \dot{s}(=\frac{ds}{dt})$. The particle motion in the x (horizontal) direction and y (vertical) direction is called as the transverse motion and the motion in the s direction is called as longitudinal motion. The notion of phase space is normally used for representing particle motion in six degrees of freedom. The four-dimensional space defined by the transverse coordinates only is called as transverse phase space. In phase space the motion of an arbitrary particle is described by a flowline. The projection of this flowline on one of the three two dimensional phase plane shows the motion along the respective dimension. When the motion of a particle in any of the phase planes only depends on its coordinate and momentum in the same phase plane the motion is called as uncoupled and can be solved separately in every plane.

3.3 Transverse plane motion

An ideal particle on the design trajectory experiences only the dipolar field component thus will circulate and keep following the ideal orbit. Whereas a particle with transverse position deviation $x(s)$ and $y(s)$ will experience the quadrupolar field and will be re-focused towards the ideal orbit thus result in particle oscillation around the central orbit in transverse plane these oscillation are called as betatron oscillations. The equation of motion in transverse plane in linear approximation (x,$y << \rho$) are given by [118]:

$$\frac{d^2x}{ds^2} + (\frac{1}{\rho^2(s)} - K(s))x = \frac{1}{\rho(s)}(\frac{\Delta p}{p}) \tag{3.1}$$

$$\frac{d^2y}{ds^2} + K(s)y = 0 \tag{3.2}$$

where $K(s) = -(\frac{dB}{dx})/(\frac{p}{q}), \rho(s) = (\frac{p}{q})/B(s)$. Here $K(s)$ is the local focusing strength, $\rho(s)$ the local radius of curvature, p the particle's momentum,Δp the momentum deviation of the particle from the ideal particle and q the particle charge. Now for an on momentum particle(i.e. $\Delta p = 0$) the above equations can be expressed as following.

$$\frac{d^2z(s)}{ds^2} + \Upsilon(s)z(s) = 0 \tag{3.3}$$

Here $\Upsilon(s)$ is a function of independent variable s and z can be x or y. $\Upsilon(s)$ is $k - \frac{1}{\rho^2(s)}$ and for a quadrupole magnet, $\Upsilon(s) = k$. For a dipole magnet, k is non-zero only if magnet has some gradient in the field along transverse direction i.e. it has a quadrupole component and $\frac{1}{\rho^2(s)}$ represents the geometrical focusing term of a dipole magnet. $\Upsilon(s)$ is different for different magnets along s and is zero in a magnet free region i.e. drift space. Within a magnet also, it varies along s due to fringing of the magnetic field at edges of a magnet. For a single magnet, using concept of 'effective length', the strength can be made constant inside this magnet and the parameter Υ becomes a step function at edges of this magnet. This model in accelerator physics is known as 'hard edge model'and solution of the equation obtained using this approximation is known as 'piece-wise solution'. In each piece, Hill's equation becomes similar to that of a simple harmonic oscillator. Under this approximation the solutions of Hills equations for drift space, dipole magnets and quadrupole magnets are obtained which provides a mapping of particles position and angle from entry to the exit of the magnet. These maps in linear dynamics are often represented by transfer matrices and are used by accelerator simulation codes like MAD for simulating the particle trajectories. These matrices relates the final coordinates to initial coordinates as following.

$$\begin{pmatrix} y_1 \\ y_1' \end{pmatrix} = \begin{pmatrix} C & S \\ C' & S' \end{pmatrix} \begin{pmatrix} y_0 \\ y_0' \end{pmatrix} \tag{3.4}$$

here C and S are the cosine like and sine like principle solutions of Hills equation for the considered piece (element) of optics. The specific matrices of different elements are discussed in detail in later sections of this thesis.

3.4 Beam emittance

All ion and electron beams can be characterized by their properties in a 6-dimensional phase space. To an arbitrary particle, a point is associated in each of the three phase planes. Thus the representation of every beam particle on the phase plane will appear as a two-dimensional cloud of points of varying density. A parameter called emittance is defined as the area on a phase plane that encircles a certain percentage of the total number of particles. In literature, several different values used for this percentage for Gaussian beams are 39.3%, 86.4% and 95.6%. Generally, for transverse phase planes, an ellipse is taken that describes the boundary of this area. Generally, this six-dimensional space is arranged into three two-dimensional planes that roughly correspond to the three coordinates in everyday life i.e. horizontal (x), vertical (y), and longitudinal (z). The parameters plotted on the axes of these planes correspond to 1) horizontal position and angle with respect to the ideal particle trajectory, 2) vertical position and angle with respect to the ideal particle trajectory, and 3) difference from the average beam energy and phase relative to some reference frequency. The smaller value of emittance is desired for better quality of the beam because large emittance beams are difficult to transport and difficult to match into other devices. The emittance is measured in units of millimeter-milliradians ($mm - mrad$) or in meter-radians($m - rad$).

3.5 Acceptance

Acceptance is an important property of accelerators and transport lines. Consider the virtual phase space just before injection into a beam-transport system (i.e. at entry to an accelerator or transport line). Now assume that there is a particle at each position in this phase space. Now if the particle from this imaginary phase space will not be lost in the system then the corresponding particle position in the phase space is called a permissible position. Now all such permissible positions will form an area in each of the three phase planes. This area is called as the acceptance of the machine/ component. Thus, one can say that the acceptance is the area of the largest ellipse that can be drawn in phase space that encircles the permissible/ survived particles for a machine. For proper and efficient injection of beam, the beam emittance is required to be matched with the machine acceptance at injection point.

3.6 Twiss parameters

Solving the Hills equation also provides us with the different optical parameters known as Twiss parameters. In this case instead of using the piece wise solution a more general solution is used. For this the solution can be written down as:

$$x(s) = A\sqrt{\beta(s)}cos(\mu(s) - \mu_0) \tag{3.5}$$

Here due to s dependence of Υ the amplitude is also taken as a function of s where the amplitude $A\sqrt{\beta(s)}$ is s dependent and A is a constant. The function $\beta(s)$ is called 'beta function'and $\mu(s)$ is the 'betatron phase'. Differentiating twice the above equation (Eq.3.5) with respect to s and then putting these values in Hill's equation and using $\alpha(s) = -\frac{1}{2}\frac{d\beta(s)}{ds}$ and $\gamma(s) = \frac{1+\alpha(s)^2)}{\beta(s)}$ gives the equation

$$\gamma y^2 + 2\alpha yy' + \beta y'^2 = \pi\varepsilon \tag{3.6}$$

This expression gives the invariant of motion and represents the equation of an ellipse with an area of $\pi\varepsilon$. The area of the ellipse is invariant of motion and is known as 'Courent-Synder invariant'. The parameters $\alpha(s), \beta(s), \gamma(s)$ are called as Twiss parameters.

Now if we consider the area of an ellipse which encloses the ellipse for all the particles in the beam then this area is called as the emittance in that dimension and thus the parameters along with the emittance actually define the beam dimensions in phase space. Figure 3.3 shows the physical importance of the Twiss parameters in relation to the phase space ellipse. The importance of the Twiss parameters is that the Twiss parameters describe the beam shape and size during beam propagation through magnetic optics. The Eq.3.6 describes the invariant of motion is used to obtain the propagation of Twiss parameters through the magnetic optics. At initial location if we consider the Twiss parameters as $\alpha_0, \beta_0, \gamma_0$ then using the invariant of motion condition at any downstream location the following relation holds.

$$\pi\varepsilon = \gamma y^2 + 2\alpha yy' + \beta y'^2 = \gamma_0 y^2 + 2\alpha_0 yy' + \beta_0 y'^2 \tag{3.7}$$

Through transfer matrices (Eq. 3.3) writing final coordinates in terms of initial coordinates and substituting in above equation and rearranging provides the relation between

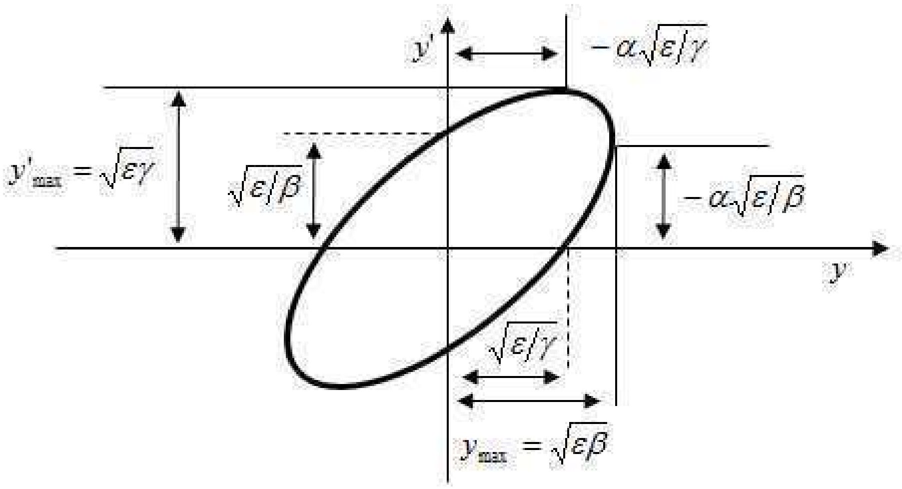


Figure 3.3: Relation of Twiss parameters with the phase space emittance ε. Here α, β and γ are the Twiss parameters describing the optical characteristics of beam.

the Twiss parameters for two locations as following.

$$\begin{pmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \end{pmatrix} = \begin{pmatrix} C^2 & -2SC & S^2 \\ -CC' & SC' + S'C & -SS' \\ C'^2 & -2S'C' & S'^2 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \beta_0 \\ \gamma_0 \end{pmatrix} \tag{3.8}$$

In this fashion for every location of optics the Twiss parameters are calculated using initial Twiss parameters. For each elements the matrix appeared in Eq.3.8 is discussed in detail in later sections.

3.7 Transport Lines

Transport lines are the elements of acceleration facilities which serve the purpose of transferring the electron beams from one physical location to another physical location and at the same time providing the means to match the beam delivery and acceptance characteristics of source and destination devices / sub-elements. Often the sources are the small accelerators called as pre-injectors (which accelerate the electrons to small energies ranging from few tens of MeV to few hundreds of MeV and usually are at low current) or boosters (which accelerate the electrons to relative high energies at higher currents) and the destination devices are the bigger accelerators or storage rings. In TL-1 case the source of electron beam for transfer line one is Microtron, an electron accelerator which accelerates the electrons to 20MeV and the destination device is the booster ring which further accelerates these electrons up to 450MeV/550MeV. To get the maximum transfer efficiency, transport line requires the matching of beam characteristic parameters (x, x', y, y') and optics characteristic parameters $(\alpha_x, \beta_x, \gamma_x, \alpha_y, \beta_y, \gamma_y)$ at both the ends with the source and destination devices emittance and acceptance parameters . This matching of parameters at both the ends is achieved by means of applying the appropriate settings to each constituent elements of transport line that results in the setting of suitable magnetic lattice. Transport line is comprised of, beam chamber that passing through different types of electromagnets and serves the purpose of maintaining the vacuum environment for beam propagation. Whereas electromagnets steer and focus the beam according to the specific requirement of the facility . For proper tuning of the beam line, beam lines are also provided with beam diagnostic devices. A brief description of the different beam line elements is presented below.

3.7.1 Magnets

The electromagnets used in accelerators for bending and focusing of the electron beams can be broadly classified in to two categories a) Dipole magnets: these are the magnets used for bending of beam b) Quadrupole magnets: these are the magnets used for focusing of the electron beams in order to confine the beams inside the beam chambers.

DIPOLE MAGNET:

A dipole magnet is a magnet constructed to create a homogeneous magnetic field over some distance. Particle motion in that field will be circular in a plane perpendicular to the

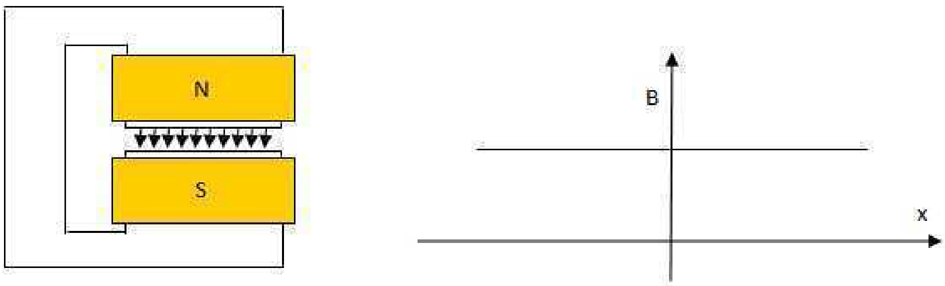


Figure 3.4: Uniform field profile for a dipole magnet.

field and collinear to the direction of particle motion and free in the direction orthogonal to it. Thus, a particle injected into a dipole magnet will travel on a circular or helical trajectory. In particle accelerators, dipole magnets are used to realize bends in the design trajectory (orbit) of the particles. Figure 3.4 shows the field profile for dipole magnets. Depending upon the shape, use and time related characteristics the Dipole magnets can be subdivided into a few other types also but their basic principle of operation remains the same i.e. to provide a uniform magnetic field over some distance called as a good field region.

A few types of dipole magnets are following.

- Sector type Dipole magnets: these are of sector type shape and are mainly used for bending of beams through large angles.
- Rectangular type Dipole magnets: these are rectangular shaped dipole magnets and used for bending of beams through large angles.
- Steering magnets: these are small dipole magnets used to provide a small angular kick to the beam.They are mainly used for correcting the beam path. Depending upon the plane in which they provide a kick, they are further classified as Vertical steering magnets or Horizontal steering magnets.
- Injection septum: this is a special type of dipole magnet which provides a dipolar field over a region for a very short interval of time in synchronous manner. From

analysis point of view these can be taken as rectangular type dipole magnets.

QUADRUPOLE MAGNET:

Quadrupole magnets consist of groups of four magnets laid out so that in the multipole expansion of the field the dipole terms cancel and where the lowest significant terms in the field equations are quadrupole. In these magnets the magnetic fields magnitude grows rapidly with the radial distance from its longitudinal axis. And thus provides a bending force on moving particles proportional to their distance from centre of magnet. i.e. the particle passing through the centre of the quadrupole will not experience any bending force and particle passing away from centre will experience a bending force thus providing a focusing action on the beam. Figure 3.5 shows the poles placement and the field profile for quadrupole. These are used for particle beam focusing.

3.7.2 Diagnostic devices

Diagnostic devices are the devices used for measurement of different beam parameters at different places in the beam line.

FLUORESCENT SCREEN TYPE BPM:

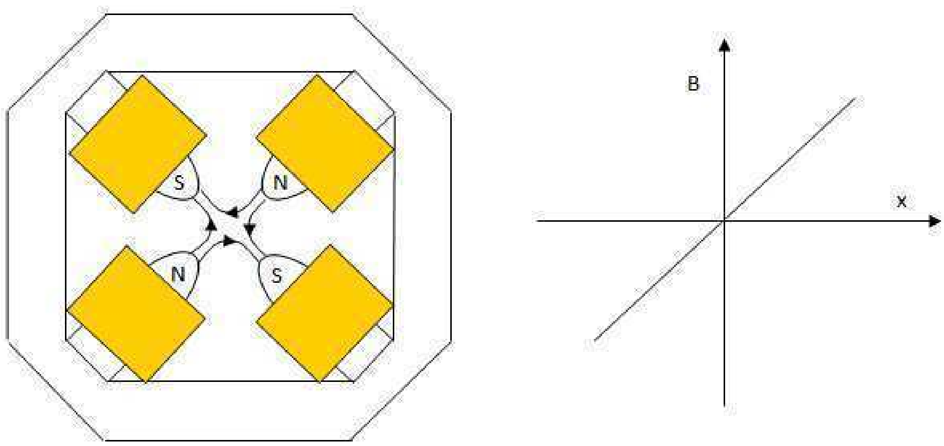


Figure 3.5: The Quadrupole magnet and its field profile.

These are the devices used to provide beam size and position and called as Beam position monitors. A BPM comprises of a fluorescent screen made up of Al_2O_3 doped with Chrome and called as 'choromex-screen' mounted on a movable support which when required can be inserted in the path of beam using stepper motor /pneumatic actuators. The electron beam striking on the screen produce light which is viewed through the window in the beam chamber with the help of CCD camera. These screens are provided with cross-hairs, for calibration and size and position measurements. Figure 3.6 shows the arrangement for the fluorescent screen type BPM. These BPM are destructive type device as the beam is destroyed in the process of measurement. In TL-1 the BPM of fluorescent screen type are used.

BUTTON ELECTRODE TYPE BPM:

The most popular non-interceptive type of beam position monitors are the four button electrode type BPMs. In four button electrode type BPMs there are four button electrodes arranged in the fashion as shown in the figure 3.7. When electron beam passes through the BPM it induces a charge on the buttons. The induced charge depends on the relative position of the beam from buttons. By measuring the voltage induced in the electrodes

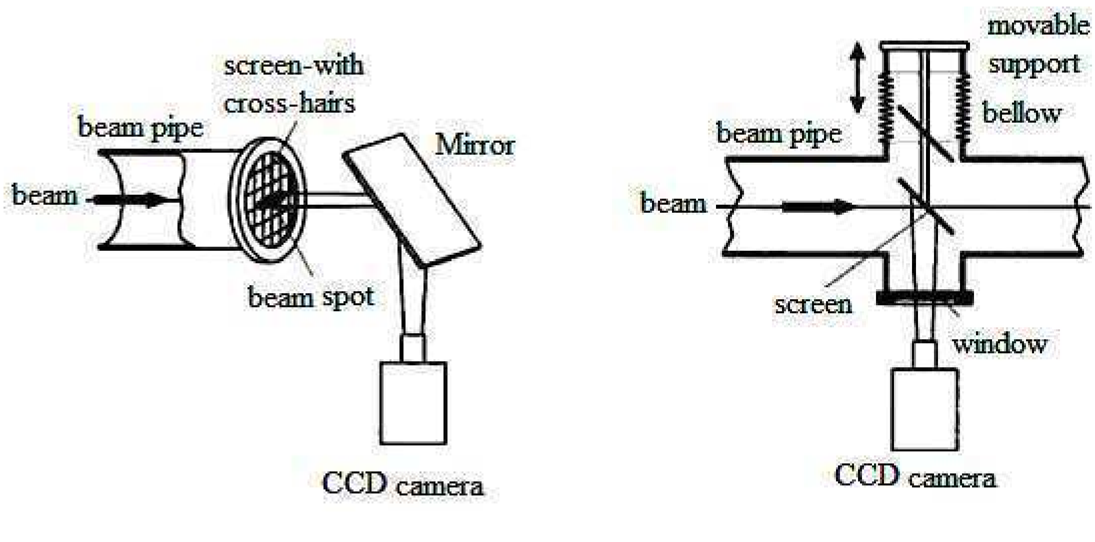


Figure 3.6: Fluorescent screen type BPM arrangement.

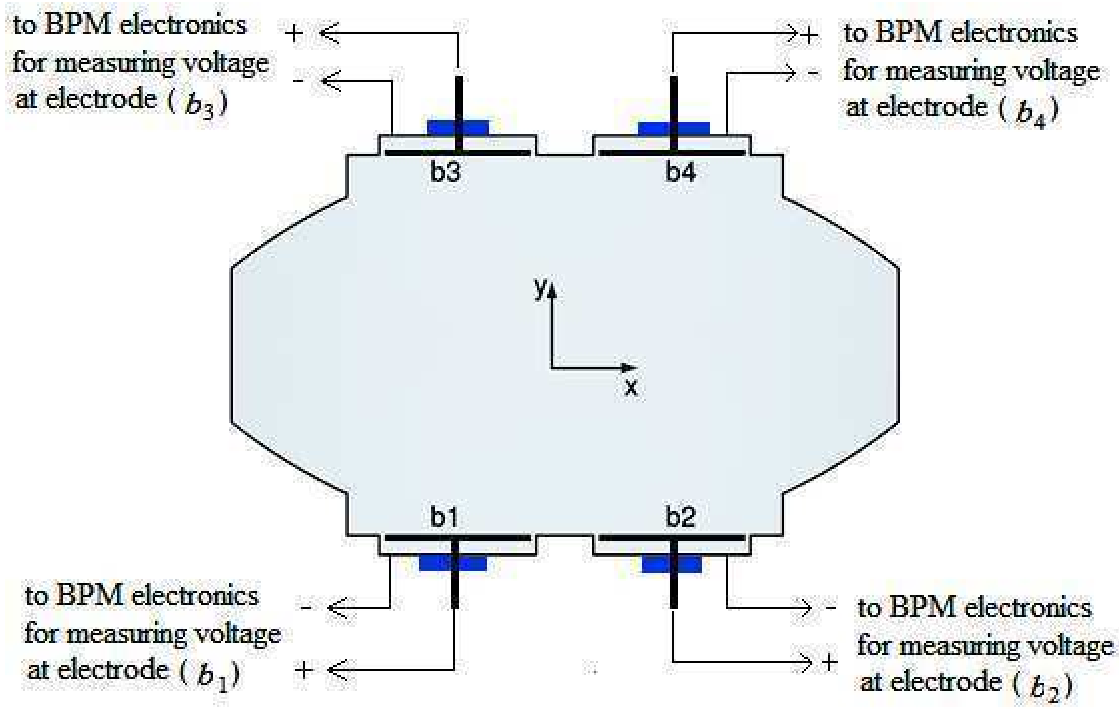


Figure 3.7: Four button electrode type BPM.

$V(b1, b2, b3, b4)$ beam position $P(x, y)$ is calculated using the following relations:

$$x = x_0 \frac{(b_2 + b_4) - (b_1 + b_3)}{b_1 + b_2 + b_3 + b_4} \tag{3.9}$$

$$y = y_0 \frac{(b_3 + b_4) - (b_1 + b_2)}{b_1 + b_2 + b_3 + b_4} \tag{3.10}$$

Where x_0 and y_0 are calibration factors set by geometry of BPM. These equations give good results when beam is near to the centre of the BPM, for the case of beam position far away from the centre of BPM, nonlinear BPM model based methods are used to increase measurement accuracy [94].

SEWM:

Secondary Emission (SEM) Grids, also known as harps, consist of ribbons or wires which are placed in the beam. As the beam intercepts the grid, secondary emission leads to a current in each strip which is proportional to the beam intensity at that location. By measuring this current for all strips a beam profile is obtained. These are mostly used to measure the density profile of beams in transfer lines. The sets of three such SEWM, properly spaced (i.e. with the right phase advance between monitors), can be used for determination of the emittance ellipse. Figure 3.8 shows the figure of a SEWM

FCT:

Fast Current transformers are the devices used to measure the beam current. Operation of FCT is based on the principle of current transformer. Beam passing through the centre of a ferrite core acts as primary and the wire wound on the core acts as the secondary pick up coil and provides the signal proportional to the beam current.

BEAM SLIT:

These are the sets of two blades placed parallel to form a slit. These are provided with the mechanical arrangement to move these slits collectively or individually. The purpose

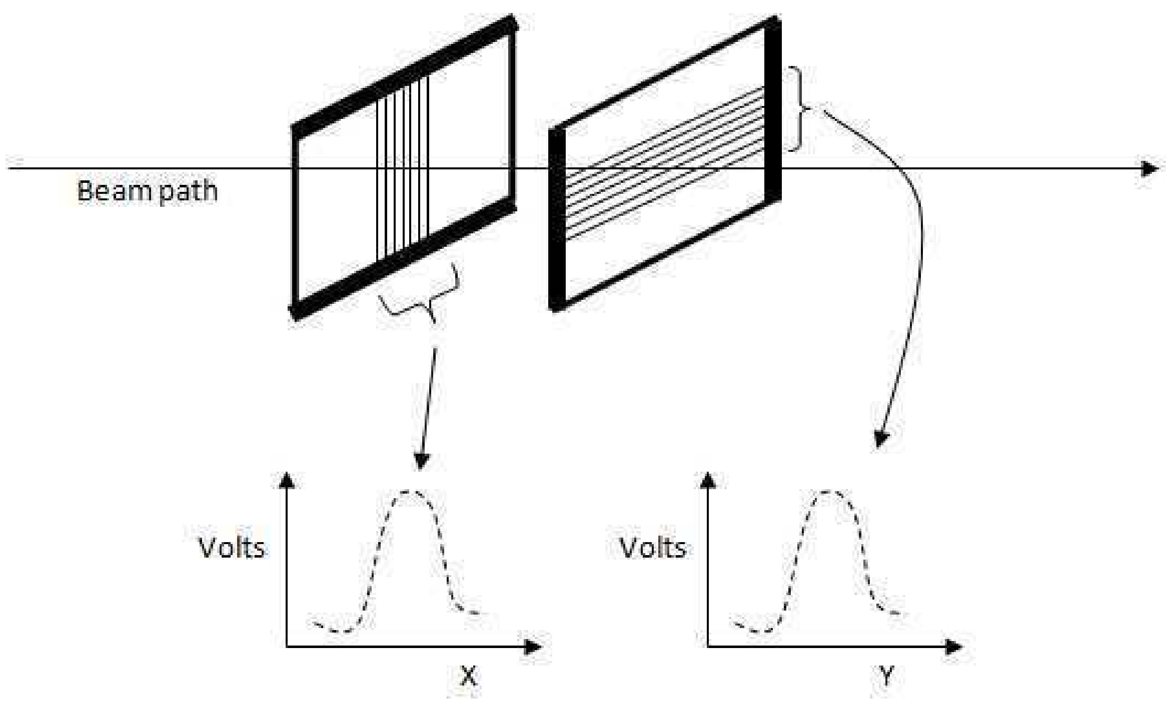


Figure 3.8: Secondary Emission Wire Monitor (SEWM).

of beam slits is to cut the unwanted part of beam. In TL-1 these slit blades are mounted with fluorescent screens so that these can also be used as BPM.

3.7.3 Modelling procedure

The standard procedure of modelling accelerator components based on the individual elements transfer matrix is used for modelling of TL-1. The transfer matrix of individual elements transforms the particle position (x, y), angle(x', y'), Twiss parameters (α, β, γ) and dispersion (δ) from input point to the output point. These transfer matrices are obtained by solving the equation of motion of charged particle for the corresponding elements [68, 118]. For simulating the particle loss behaviour by the element, a physical aperture at the input of element is simulated in the form of constraint function F_c given by

$$F_c = C_0 \bigwedge C_x \bigwedge C_y \tag{3.11}$$

$$C_0 = \text{ particle condition at the input of element} \tag{3.12}$$

$$C_x = (x_0^{HL} \geq x_0 \geq x_0^{LL}) \bigwedge (x_1^{HL} \geq x_1 \geq x_1^{LL}) \tag{3.13}$$

$$C_y = (y_0^{HL} \geq y_0 \geq y_0^{LL}) \bigwedge (y_1^{HL} \geq y_1 \geq y_1^{LL}) \tag{3.14}$$

where $\bigwedge$ represents the Logical AND operation. The symbols x_0^{HL}, y_0^{HL}, x_0^{LL} and y_0^{LL} represents the high limit and low limit values in horizontal and vertical plane at the element input. Similarly symbols x_1^{HL}, y_1^{HL}, x_1^{LL} and y_1^{LL} represents the high limit and low limit values in horizontal and vertical plane at the element output. These limits are calculated based on the physical constraints for the element such as mechanical dimensions of beam pipes and internal apertures of different elements.

THE TRANSFER MATRIXES OF DIFFERENT ACCELERATOR ELEMENTS:

Drift space: Drift space is the free space between two elements covered with beam pipe (to sustain low vacuum requirements). In drift space the particle moves freely without influence of any external force. The transfer matrices to calculate the particle position $(x, y)$and angle (x', y') for drift space are given below. The transfer matrices for calculating the Twiss parameters for drift space are also given below. L denotes the Length of the drift space.

$$M^x_{drift} = M^y_{drift} = \begin{pmatrix} 1 & L & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad T^x_{drift} = T^y_{drift} = \begin{pmatrix} 1 & -2L & L^2 \\ 0 & 1 & L \\ 0 & 0 & 1 \end{pmatrix} \tag{3.15}$$

The following equation gives how one can calculate the beam parameters $(x_1, x_1^{'}, , \delta_1^x, \alpha_1^x, \beta_1^x, \gamma_1^x)$ at output of the drift space from the parameters $(x_0, x_0^{'}, , \delta_0^x, \alpha_0^x, \beta_0^x, \gamma_0^x)$ provided at the input of drift space using the matrix M^x_{drift} and T^x_{drift} :

$$\begin{pmatrix} x_1 \\ x_1' \\ \delta_1^x \end{pmatrix} = \begin{pmatrix} 1 & L & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_0' \\ \delta_0^x \end{pmatrix} \quad and \quad \begin{pmatrix} \alpha_1^x \\ \beta_1^x \\ \gamma_1^x \end{pmatrix} = \begin{pmatrix} 1 & -2L & L^2 \\ 0 & 1 & L \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_0^x \\ \beta_0^x \\ \gamma_0^x \end{pmatrix} \tag{3.16}$$

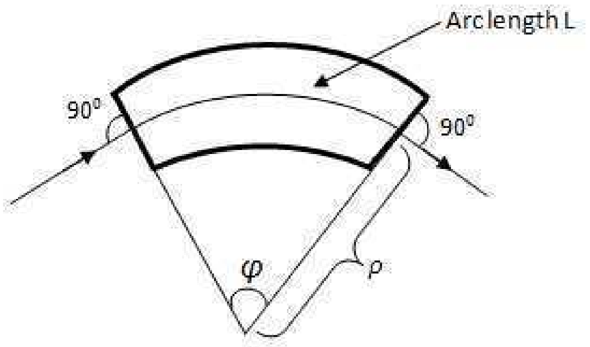


Figure 3.9: Dipole sector Magnet

Sector type Dipole magnet: For sector type dipole the transfer matrix are given by:

$$M^x_{SectorDipole} = \begin{pmatrix} \cos\varphi & \rho\sin\varphi & \rho(1-\cos\varphi) \\ -\frac{1}{\rho}\sin\varphi & \cos\varphi & \sin\varphi \\ 0 & 0 & 1 \end{pmatrix} \tag{3.17}$$

$$M^y_{SectorDipole} = \begin{pmatrix} 1 & L & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.18}$$

$$T^x_{SectorDipole} = \begin{pmatrix} \cos^2\varphi & -2\rho\sin\varphi\cos\varphi & \rho^2\sin^2\varphi \\ \frac{1}{\rho}\sin\varphi\cos\varphi & \cos^2\varphi - \sin^2\varphi & \rho\sin\varphi\cos\varphi \\ \frac{1}{\rho^2}\sin^2\varphi & \frac{2}{\rho}\sin\varphi\cos\varphi & \cos^2\varphi \end{pmatrix} \tag{3.19}$$

$$T^y_{SectorDipole} = \begin{pmatrix} 1 & -2L & L^2 \\ 0 & 1 & L \\ 0 & 0 & 1 \end{pmatrix} \tag{3.20}$$

where L is the arc length (path length travelled by the particle inside the dipole magnet),

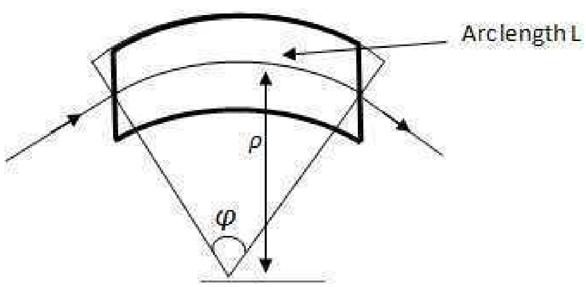


Figure 3.10: Dipole sector Magnet.

ρ is the radius of curvature and $\varphi = \frac{L}{\rho}$ is the dipole bending angle as shown in figure 3.9.

Rectangular type dipole magnet: For rectangular type dipole the transfer matrix are given by:

$$M^x_{RectDipole} = \begin{pmatrix} 1 & \rho\sin\varphi & \rho(1-\cos\varphi) \\ 0 & 1 & 2\tan\frac{\varphi}{2} \\ 0 & 0 & 1 \end{pmatrix} \tag{3.21}$$

$$M^y_{RectDipole} = \begin{pmatrix} \cos\varphi & \rho\sin\varphi & 0 \\ -\frac{1}{\rho}\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.22}$$

$$T^x_{RectDipole} = \begin{pmatrix} 1 & -2\rho\sin\varphi & \rho^2\sin^2\varphi \\ 0 & 1 & \rho\sin\varphi \\ 0 & 0 & 1 \end{pmatrix} \tag{3.23}$$

$$T^y_{RectDipole} = \begin{pmatrix} \cos^2\varphi & -2\rho\sin\varphi\cos\varphi & \rho^2\sin^2\varphi \\ \frac{1}{\rho}\sin\varphi\cos\varphi & \cos^2\varphi - \sin^2\varphi & \rho\sin\varphi\cos\varphi \\ \frac{1}{\rho^2}\sin^2\varphi & \frac{2}{\rho}\sin\varphi\cos\varphi & \cos^2\varphi \end{pmatrix} \tag{3.24}$$

Steering magnets: The steering magnets are dipole magnets of small size that provide very small amount of kick, thus practically for simulation purpose these do not affect the position and only add the kick to the angle part. Thus the following equations are used for exercising the steering coils.

For Horizontal steering magnets:

$$x_1 = x_0 \tag{3.25}$$

$$x_1^{'} = x_0^{'} + \phi \tag{3.26}$$

where ϕ is the kick given by steering magnet. For Vertical steering magnets:

$$y_1 = y_0 \tag{3.27}$$

$$y_1^{'} = y_0^{'} + \phi \tag{3.28}$$

It is assumed that the changes made by steering coils to the Twiss parameters is negligible.

Focusing Quadrupole magnets (horizontal): The transfer matrix for focusing quadrupole is given by:

$$M^x_{Quadrupole} = \begin{pmatrix} \cosh\varphi & \frac{1}{\sqrt{|K|}}\sinh\varphi & 0 \\ \sqrt{|K|}\sinh\varphi & \cosh\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.29}$$

$$M^y_{Quadrupole} = \begin{pmatrix} \cosh\varphi & \frac{1}{\sqrt{|K|}}\sinh\varphi & 0 \\ -\sqrt{|K|}\sinh\varphi & \cosh\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.30}$$

$$T^x_{Quadrupole} = \begin{pmatrix} \cosh^2\varphi & -\frac{2}{\sqrt{|K|}}\sinh\varphi\cosh\varphi & \frac{1}{|K|}\sin^2\varphi \\ -\sqrt{|K|}\sinh\varphi\cosh\varphi & \sin^2\varphi + \cosh^2\varphi & -\frac{1}{\sqrt{|K|}}\sinh\varphi\cosh\varphi \\ |K|\sin^2\varphi & -2\sqrt{|K|}\sinh\varphi\cosh\varphi & \cosh^2\varphi \end{pmatrix} \tag{3.31}$$

$$T^y_{Quadrupole} = \begin{pmatrix} \cosh^2\varphi & -\frac{2}{\sqrt{|K|}}\sinh\varphi\cosh\varphi & \frac{1}{|K|}\sin^2\varphi \\ \sqrt{|K|}\sinh\varphi\cosh\varphi & \cosh^2\varphi - \sin^2\varphi & -\frac{1}{\sqrt{|K|}}\sinh\varphi\cosh\varphi \\ |K|\sin^2\varphi & 2\sqrt{|K|}\sinh\varphi\cosh\varphi & \cosh^2\varphi \end{pmatrix} \tag{3.32}$$

For the case of defocussing quadrupole the horizontal and vertical plane transfer matrix interchanges rest all remains same.

3.8 Model of electron beam

The electron beam coming out of Microtron is modelled with the help of two dimensional array of structure representing the beam in the transverse plane i.e. the plane perpendicular to the beam motion as shown in figure 3.11. This approach is adopted because of ease in mapping the relative beam profile from the Image obtained from beam position monitors. Here each element of the array (the structure containing the I(Intensity in

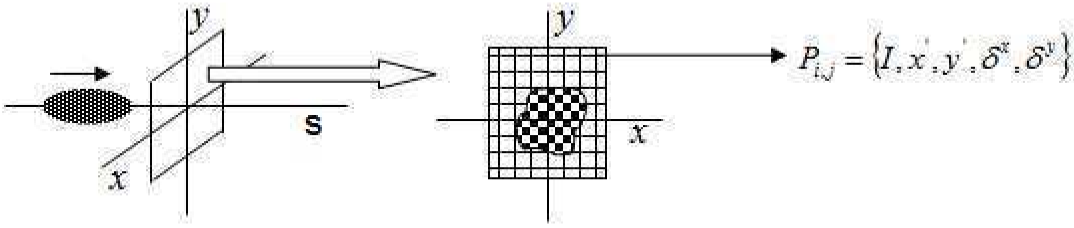


Figure 3.11: Modelling of electron beam.

relative units), x', y',δ_x , δ_y) and with the index values (i, j) representing the position of the element (x, y) stores the information of the unit part of beam at the respective point.

Modelling of the beam profile propagation through the TL-1 is achieved by transforming each element from the defined input beam to the calculated output beam parameter at the desired position s in TL-1 using the developed TL-1 model. As shown in the figure 3.12. The intensity of each point at the output is calculated by summing the contribution from each unit input point after transforming to the output coordinates.

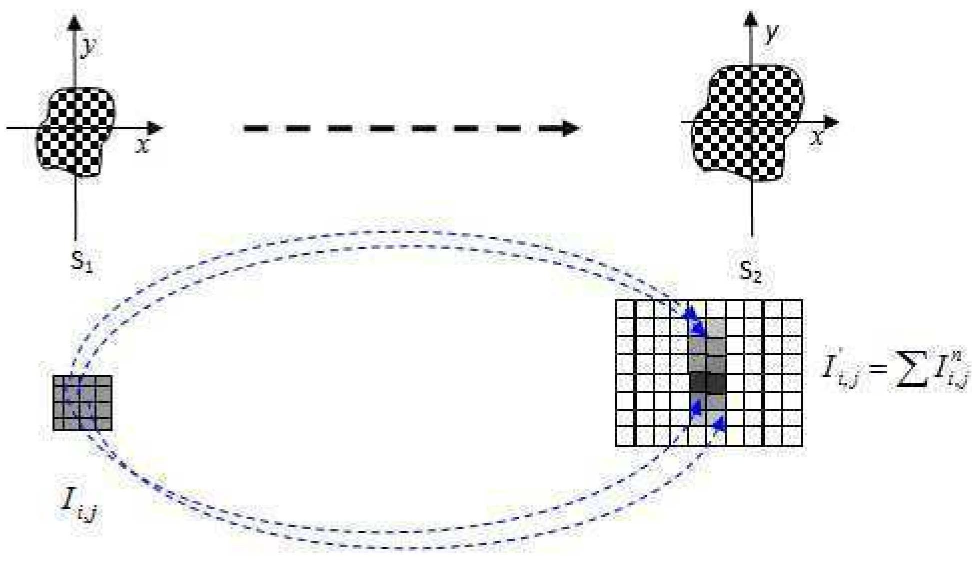


Figure 3.12: Beam profile propagation.

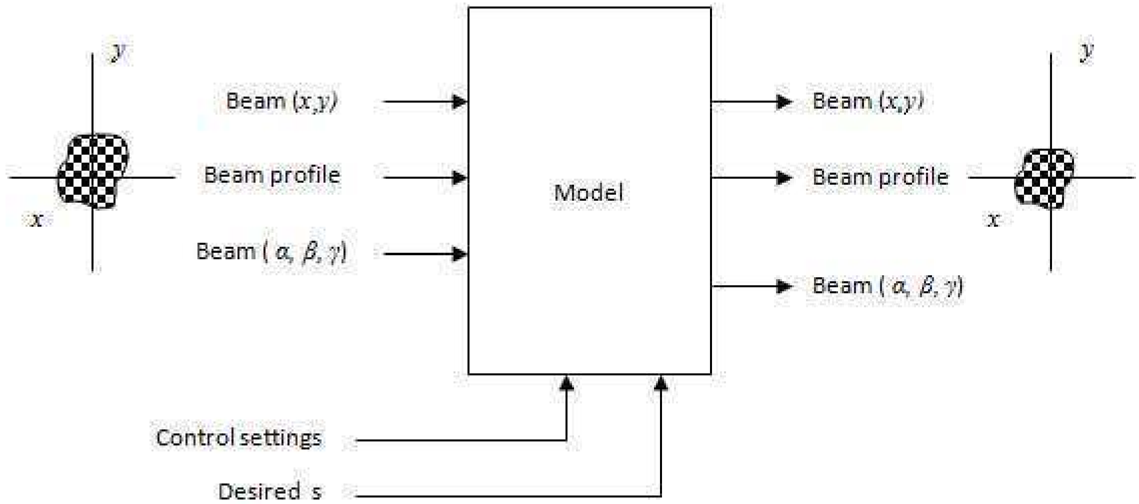


Figure 3.13: Requirements from TL-1 model.

3.9 Model of TL-1

The model of TL-1 is constructed by combining the models of individual components (discussed in previous section) of TL-1. It is constructed considering the following requirements from model.

- The model should accept the control setting values for different magnets and according to the control setting values it should provide relationship between the following different variables.
  - The beam position (x, y), beam angle (x', y') and profile throughout the TL-1 (specifically at BPM locations).
  - The model should provide the Twis parameters throughout the TL-1.
- The model should provide the beam current/beam profile simulating the particle loss through the beam transportation.

Figure 3.13 shows the black box representation of the model depicting the input and output requirements.

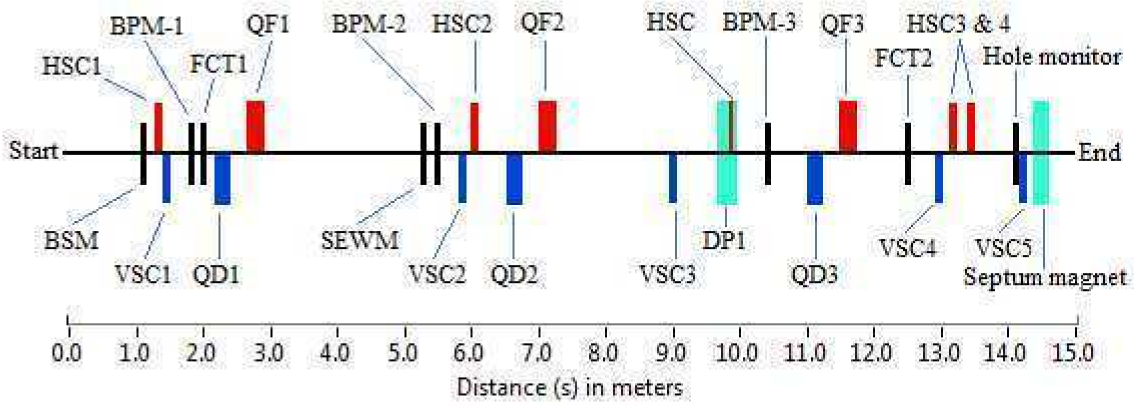


Figure 3.14: Schematic of the TL-1.

TL-1 LAYOUT:

Transport Line- 1 comprises of Microtron extraction tube, three pairs of dc-quadrupoles (QF1-QF3 and QD1-QD3), four horizontal steerers (HSC1-HSC4), five vertical steerers (VSC1-VSC5), one 15^0 dc-dipole (DP1) and one 15^0 injection septum (pulse width 200s). For measurement of beam position there are three fluorescent screen type beam position monitors (BPM1-BPM3). Figure 3.14 shows the layout of the transport line-1. The figure

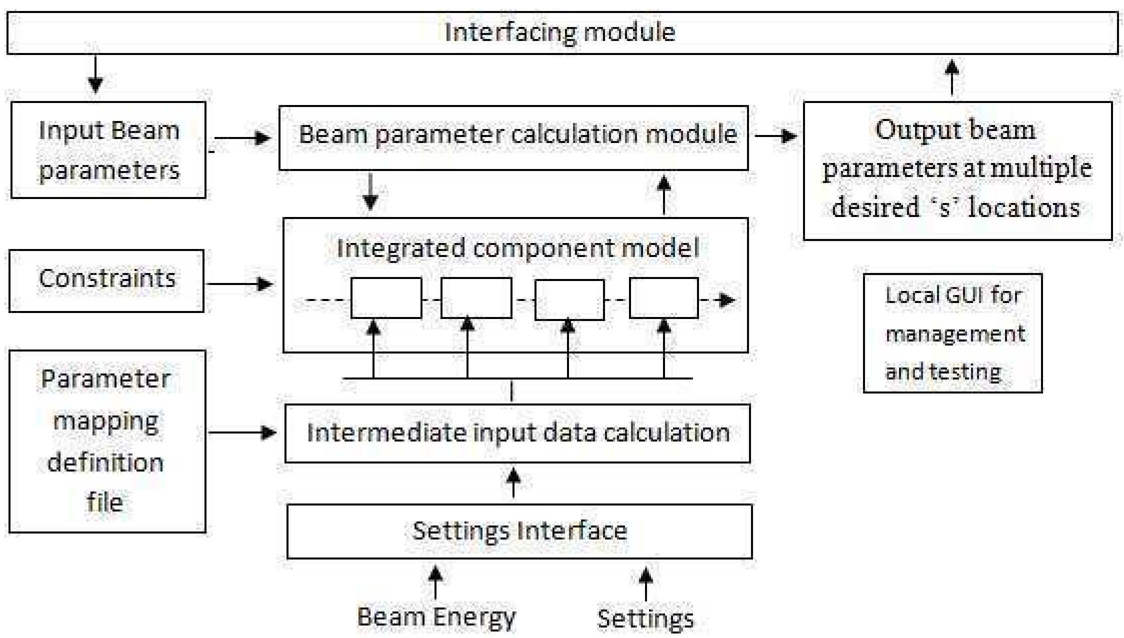


Figure 3.15: Block diagram for overall model of TL-1.

3.15 shows the block diagram of the formulated TL-1 model. The intermediate input data calculation block serves the purpose of calculating the value of various parameters which are used in the matrix as inputs from the current settings (in the form of currents in various coils in units of ampere) and the beam energy using the parameter mapping data definitions provided in file. The beam parameter calculation module calculates the different beam parameters by repeatedly using the integrated component model. The Table 3.1 and 3.2 shows the TL-1 design parameters used for making the TL-1 integrated component Model.

Table 3.1: List of TL-1 components.

Sr.No	Identifier	Type	Parameters	Comments
1	DP1	DIPOLE	L=0.222400 ANGLE=0.2617	This dipole is of sector type. HSC is placed midway between the dipole.
2	QF1 to QF3	QUADRUPOLE	L=0.2454 KF= calculated	Focussing Quadrupole
3	QD1 to QD3	QUADRUPOLE	L=0.2454 -KF= calculated	Defocussing Quadrupole
4	HSC1 to HSC4	HKICKER	L=0.00 KICK= calculated	Horizontal Corrector
5	VSC1 to VSC5	VKICKER	L=0.0000 KICK= calculated	Vertical Corrector
6	BPM1 to BPM3	BPI	L=0.00	Fluorescent screen type BPIs

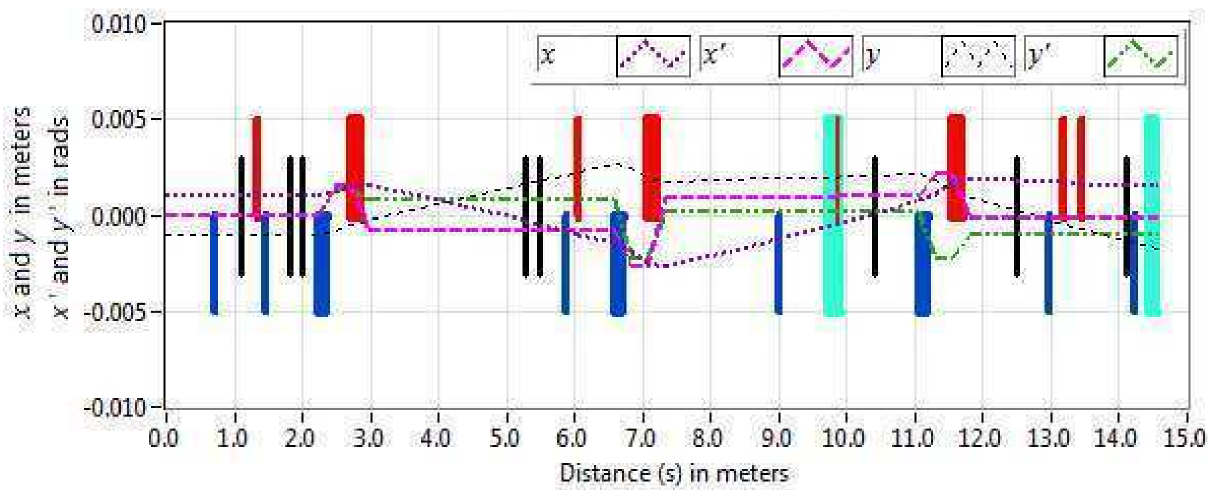


Figure 3.16: Propagation of $x, x^{'}, y, y^{'}$ with (s) for initial $x = 1mm$, $y = -1mm$ and $x' = y' = 0rad$. For the current control settings.

Table 3.2: List of drift space between different components of TL-1.

Sr.No	Identifier	Type	Parameters	Comments
1	D000	DRIFT	L=0.73000	Ext point to VSC(E) IMG
2	D00	DRIFT	L=0.37800	CVE to BPM-1
3	D01	DRIFT	L=0.22000	BPM-1 to HSC1
4	D02	DRIFT	L=0.14000	HSC1 to VSC1
5	D03	DRIFT	L=0.35300	VSC1 to FCT1
6	D06	DRIFT	L=0.19100	FCT1 to BPM-1
7	D07	DRIFT	L=0.22230	BPM-1 to QD1
8	D08	DRIFT	L=0.24430	QD1 to QF1
9	D09	DRIFT	L=1.42500	QF1 to BPM-2
10	D10	DRIFT	L=0.23100	BPI-2 to SEWM
11	D11	DRIFT	L=0.36700	SEWM to VSC2
12	D12	DRIFT	L=0.17800	VSC2 to HSC2
13	D13	DRIFT	L=0.54400	HSC2 to QD2
14	D14	DRIFT	L=0.24850	QD2 to QF2
15	D15	DRIFT	L=1.69430	QF2 to VSC-3
16	D16	DRIFT	L=0.71180	VSC-3 to DP1
17	D17	DRIFT	L=0.45280	DP1 to BPM-3
18	D18	DRIFT	L=0.65030	BPM-3 to QD3
19	D19	DRIFT	L=0.25360	QD3 to QF3
20	D20	DRIFT	L=0.70730	QF3 to FCT2
21	D21	DRIFT	L=0.47300	FCT2 to VSC-4
22	D22	DRIFT	L=0.21200	VSC-4 to HSC3
23	D23	DRIFT	L=0.26400	HSC3 to HSC4
24	D241	DRIFT	L=0.65800	HSC4 to HOLE MONITOR
25	D242	DRIFT	L=0.10000	HOLE MONITOR to VSC5
26	D25	DRIFT	L=0.21200	VSC5 to SEPTUM
27	D261	DRIFT	L=0.00001	SEPTUM to BPM (IMG)
28	D262	DRIFT	L=0.00001	BPM(IMG) to OUTPUT

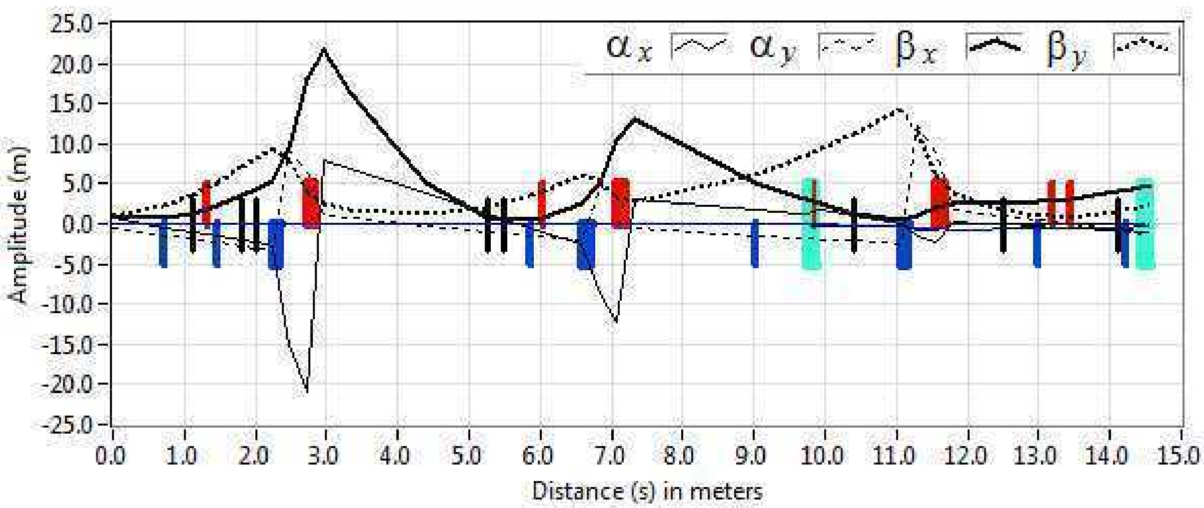


Figure 3.17: Propagation of Twis parameters (α, β) along transfer line with initial $\alpha_x = 0.6708, \beta_x = 0.9645, \alpha_y = -0.5046, \beta_y = 0.9945$, and for current control settings.

Figure 3.16 shows the propagation of $x, x^{'}, y$ and $y^{'}$ with (s) for (initial $x = 1mm, y = -1mm$ and $x^{'} = y^{'} = 0rad$, with normal control settings) a typical working case. Figure 3.17 shows the propagation of Twis parameters (α, β, γ) along transfer line with initial conditions $\alpha_x = 0.6708, \beta_x = 0.9645, \alpha_y = -0.5046, \beta_y = 0.9945$, with normal control settings.

3.10 Accelerator Simulation codes

For design and simulation of accelerator, accelerator physicists use different computer codes like TRACE-3D [114], MAD [70], TRANSPORT [49], PBO Lab [87], SIMION [109], SYNERGIA [5] and AT [110], particle tracking and general purpose accelerator physics codes. TRACE-3D is an interactive first-order beam-dynamics code that calculates the envelopes of a bunched beam, including linear space-charge forces, through a user-defined transport system. It provides a graphics display of the envelopes and the phase-space ellipses in three-dimensions. MAD stands for methodical accelerator design. It is a tool for simulation study of charged particle optics in alternating gradient accelerators and beam lines. Its main features include, linear lattice parameter calculation, linear lattice matching, transfer matrix matching, survey calculations, error definitions, closed orbit correction, particle tracking, chromatic effects and resonances, and intra-beam scattering. TRANSPORT is a beamline transfer-matrix calculation and fitting program that calculates floor coordinates, beam matrix, and first-, second-, and third-order transfer matrices. Elements include arc and rectangular bending magnets, quadrupoles, sextupoles, octupoles, solenoids, electrostatic septa, and accelerating cavities. It is capable of simulating misalignments, beam steering, and random errors. SIMION 3D and PBO Lab are the commercial accelerator simulation and design packages. The SIMION 3D is used for modelling of electron and ion optics for simulating ion flight through electrostatic and magnetic lenses, ion traps, quadrupoles, ion source and detector optics. It is a full Windows program featuring 3D electrostatic/magnetic arrays, cutaways of 3D views, movie effects, data recording, charge repulsion, solid geometry definition files, and a user program interface. It models ion optics problems with 2D symmetrical and/or 3D asymmetrical electrostatic and/or magnetic potential arrays. It incorporates the additional utility functions, such as CAD model import, a new user programming language, and API support. PBO Lab basic package consists of three main parts: the graphic user interface with beamline construction kit, an interactive tutorial system, and tools for analysing single particle trajectories, beam envelopes, and beamline layouts. PBO Lab is available with different Computation Modules. Current Modules include (1) TRANSPORT, a third-order program that uses matrix methods for modelling and optimizing particle beam optics systems, and (2) TURTLE, a multi-particle simulation program that can trace several thousand rays through beamlines designed with PBO Lab. SYNERGIA is a parallel beam dynamics simulation package with fully three dimensional space-charge capabilities and a higher order optics implementation. The code itself is a hybrid system based on previously developed accelerator physics codes. It includes enhancements to these codes as well as new integration and interface modules. It is unique in that it is designed to provide a high level framework specifically for studying 3D multi- particle dynamics in a massively parallel computing environment. AT stands for accelerator toolbox. It is an open-ended collection of tools MATLAB functions and scripts arranged in functional groups as opposed to one giant engine. Most of AT is written in MATLAB programming language. This code structure greatly simplifies the process of adding and sharing new tools. The computationally intensive routines are written in C/C++ and compiled into MEX-files (binary code executable from within MATLAB). Table 3.3 summarises the comparison for these accelerator simulation codes.

Out of these codes MAD and AT are the two most frequently and widely used code in accelerator community. MAD being the simple command prompt based executable provides the opportunity of its use by other programming languages that provides the construct for command prompt based execution of exe files within the programming environment. It uses a simple programming syntax with the component based declarative style, where one can construct the complete accelerator / transport line by declaring the components and its parameters in a text file. The AT is all together different from all the other codes in the sense that it is not a software package/ program in its own instead it is a collection of functions for simulating the accelerator in MATLAB programming environment. Thus providing a very flexible programming environment for simulation and its interface/use with the other MATLAB features like control system tool box and simulink. In this thesis we have used both MAD and AT for developing the accelerator models.

Table 3.3: Comparison of different accelerator simulation codes.

Name	Capability	Complexity	Availability	Maintainability and platform	Cost
TRACE-3D	RMS envelop space-charge modeling, EM cavity model with bends, photon beam transport.	Medium level, graphical display of envelops and the phase-space ellipses in three-dimensions feature.	Los Alamos National Laboratory, University of California, USA	Good, latest version released on 04/25/2006 Windows (WinXP and Win 2000) Linux	Free for non-commercial and personal use for registered users.
MAD	Linear lattice parameter calculation, linear lattice matching, transport matrix matching, survey calculations, error definitions, closed orbit correction, particle tracking, chromatic effects and resonances and intra beam scattering.	High level, graphical display of envelops and the phase-space ellipses. Complete particle trace information in the form of text files.	From CERN European Laboratory for particle Physics	Excellent, The latest MAD-X version released on 24/05/2013, Linux, Mac OS X and Windows	Free
TRANS-PORT	beamline transfer-matrix calculation and fitting program that calculates floor coordinates, third order beam matrix, simulates are arc and rectangular bending magnets, quadrupoles, sextupoles, octupoles, solenoids, electrostatic septa RF cavities.	Medium level, with graphical display of envelops and the phase-space ellipses. Graphical assistance in design with component selection like features.	Many versions are available. Graphical version from Paul Scherrer Institute, Switzerland is a good maintained one	Good, last release in 2010. Linux, Mac OS X and Windows	Free
PBOLab	It is a collection of particle accelerator related codes like TRANSPORT, TURTLE, TRACE 3-D, PARMILA with advanced features integrated for optimization and parameter fitting under single shell.	High level, with GUI framework that provides advanced tools for beamline design, modeling, and simulation tasks with easy-to-use drag-and-drop features.	From AccelSoft Inc, Del Mar, CA, USA	Excellent, last release 28/10/2011, Windows XP/Vista/7	Commercial software with option to select different modules according to the simulation requirements.

Name	Capability	Complexity	Availability	Maintainability and platform	Cost
SIMION	Electrostatic field solving in 2D and 3D, Magnetic field support, Low frequency time-dependent or RF support, Particle tracing, Geometry definition via CAD import from STL format, can utilize multiple CPU cores.	High level, with Contour and potential energy surface plots, interactive viewing with adjustment of parameters and viewing of the system even during ion flight. Support for cutting away volumes to see trajectories inside; zooming; viewing potential energy surfaces, contour lines, and trajectories; and reflying particles as dots for movie effects	Scientific Instrument Services, Inc. Ringoes, NJ, USA	Excellent, latest release July 2013, Windows XP/Vista/7	Commercial software.
SYNERGIA	Beam dynamics simulation capabilities with 3D space charge and higher order optics implementations. Designed specifically for studying 3D multi-particle dynamics in a parallel computing environment.	High level, with advanced integration and interface modules	developed at Fermilab through the Advanced Accelerator Modeling project through Collaborative effort	Available for Fermilab Computing Division's (Advanced Accelerator Simulation project), Linux	Open source, available on request
AT	It is a collection of tools to model particle accelerators and beam transport lines in the MATLAB environment	It is not a program / simulation code instead it is the library in MATLAB environment for simulating the accelerators, thus it provides the flexibility of simulation integration with the other features of MATLAB like control system toolbox, simulink, curve fitting and optimization.	Available from SLAC National Accelerator laboratory, California, USA	Good , latest release 2001, MATLAB	Free

3.11 Booster Model

Booster synchrotron at INDUS-1 complex is a synchrotron accelerator. It accelerate electrons from 20MeV to 450MeV for injection into INDUS-1. For injection into INDUS-2 booster-cum-storage ring[96], it accelerate electrons from 20MeV to 550MeV.

For simulating the amount of current injected in booster from the electron beam delivered by TL-1 for different TL-1 magnet settings, the model of booster is developed using MAD code. Since MAD directly does not provide the interface facility for changing the component settings by another program. The overall module of booster model is constructed by writing the wrapper code so that the model developed could be used in developing the accelerator system simulator. This Booster module accepts the electron beam defined in the form of array of structure. Each element of which represent a macro particles with x (horizontal position), y (vertical position), x'(angle with x axis), y'(angle with y axis), δ_x (dispersion in x direction), δ_y (dispersion in y direction) attributes. Depending upon the magnet settings the model constructs the machine lattice. Using this lattice it tracks the particle for the defined number of turns (1000 turns) and produces the macro particle beam with attributes x, y, x', y', δ_x, δ_y and the survived/lost condition for each macro particle. Depending upon the survived/lost condition of macro particles it also provides the normalised beam current injected into the booster.

3.11.1 Modelling procedure

The basic model of the booster is simulated in MAD software package. MAD provide the facility to simulate the accelerator model by integrating the basic building blocks of the accelerator like dipole magnets, quadrupole magnets, aperture etc for the stated model. It requires each component to be specified with the required/optional attributes using the specific syntax. Table 3.4 list the drift space used between different components in the Booster model simulation. Table 3.5 list all the booster components with the typical parameter values used in the simulation.

Table 3.4: List of drift space between different Booster components.

Sr.No	Identifier	Type	Parameters	Comments
1	D11	DRIFT	L=0.275000	Inj point to QF
2	D12	DRIFT	L=0.295000	QF to Inj kicker-2
3	D13	DRIFT	L=0.206000	Inj kicker-2 to BPI-1
4	D14	DRIFT	L=0.169000	BPI-1 to QD
5	D15	DRIFT	L=0.375000	QD to BM-1
6	D21	DRIFT	L=0.270900	BM-1 to Inj kicker-3
7	D22	DRIFT	L=0.491000	Inj kicker-3 to Ext kicker
8	D23	DRIFT	L=0.547700	Ext kicker to QF
9	D24	DRIFT	L=0.175000	QF to BPI-2
10	D25	DRIFT	L=0.190000	BPI-2 to VSC-1
11	D26	DRIFT	L=0.305000	VSC-1 to QD
12	D27	DRIFT	L=0.375000	QD to BM-2
13	D31	DRIFT	L=1.309600	BM-2 to QF
14	D32	DRIFT	L=0.175000	QF to BPI-3
15	D33	DRIFT	L=0.190000	BPI-3 to VSC-2
16	D34	DRIFT	L=0.305000	VSC-2 to QD
17	D35	DRIFT	L=0.375000	QD to BM-3
18	D41A	DRIFT	L=0.295000	BM-3 to Ext Septum
19	D41B	DRIFT	L=1.014600	Ext Septum to QF
20	D42	DRIFT	L=0.173400	QF to BPI-4
21	D43	DRIFT	L=0.176500	BPI-4 to VSC-3
22	D44	DRIFT	L=0.320000	VSC-3 to QD
23	D45	DRIFT	L=0.375000	QD to BM-4
24	D51	DRIFT	L=0.652000	BM-3 to Ext-septum
25	D52	DRIFT	L=0.657600	Ext-septum to QF
26	D53	DRIFT	L=0.175000	QF to BPI-5
27	D54	DRIFT	L=0.180000	BPI-5 to VSC-4
28	D55	DRIFT	L=0.315000	VSC-4 to QD
29	D56	DRIFT	L=0.375000	QD to BM-5
30	D62	DRIFT	L=0.295000	QF to Inj kicker-1
31	D63	DRIFT	L=0.202000	Inj kicker-1 to BPI-6
32	D64	DRIFT	L=0.173000	BPI-6 to QD
33	D65	DRIFT	L=0.375000	QD to BM-6
34	D66	DRIFT	L=0.554600	BM-6 to VSC-5
35	D67	DRIFT	L=0.480000	VSC-5 to Inj point
36	D61A	DRIFT	L=0.654800	BM-5 to RF Cavity

Using the discreet components first six super-periods are declared by declaring the sequence of different components in each super-period using the LINE command and then by declaring the sequence of these super-periods, the booster lattice is defined again using the LINE command. Table 3.6 lists the sequence of different components in the six super periods. For calculating the particle survival/lost condition during the injection process

Table 3.5: List of Booster components.

Sr.No	Identifier	Type	Parameters	Comments
1	DP1 to DP6	DIPOLE	L=1.887000 ANGLE=1.047197 E1=0.0,E2=0.0 HGAP=0.51e-3 FINT=0.00	All the dipoles are of sector type. HSC are placed midway between the dipoles.
2	QF1 to QF6	QUADRUPOLE	L=0.1250K1 KF= calculated	Focussing Quadrupole
3	QD1 to QD6	QUADRUPOLE	L=0.1250K1 -KF= calculated	Defocussing Quadrupole
4	HSC1 to HSC6	HKICKER	L=0.00 KICK= calculated	Horizontal Corrector
5	VSC1 to VSC5	VKICKER	L=0.0000 KICK= calculated	Vertical Corrector
6	INK1 to INK3	HKICKER	L=0.00 KICK= calculated	Injection Kickers
7	EC	ECOLLIMATOR	L=0.0000 XSIZE = 30.00E-3 YSIZE = 30.00E-3	Aperture

Table 3.6: Component sequence in super-periods.

Super period	Component sequence
1	D11,QF1,QF1,D12,INK2,D13,D14,QD1,QD1,D15,DP1,HSC1,DP1
2	D21,INK3,D22,D23,QF2,QF2,D24,D25,VSC1,D26,QD2,QD2,D27,DP2,HSC2,DP2
3	D31,QF3,QF3,D32,D33,VSC2,D34,QD3,QD3,D35,DP3,HSC3,DP3
4	D41A,D41B,QF4,EC,QF4,D42,D43,VSC3,D44,QD4,QD4,D45,DP4,HSC4,DP4
5	D51,D52,QF5,QF5,D53,D54,VSC4,D55,QD5,EC,QD5,D56,DP5,HSC5,DP5
6	D61A,D61A,QF6,QF6,D62,INK1,D63,BP6,D64,QD6,QD6,D65,DP6,HSC6,DP6,D66,VSC5,D67

the particle is placed at the injection point(start of super-period number 1) at the location defined by position vector (x, y), then using the particle state parameters (x', y', δ_x, δ_y) the particle is tracked for given number of turns (1000 numbers). After tracking, particle position vector and state vector are read to calculate the particle survived/lost condition. During simulation the following conditions are assumed.

1. There is no loss of particle during their traversal from TL-1 to Booster.

2. RF is in turned OFF condition.

3. Particles energy remains constant.

4. Injection-septum and kickers are not used.

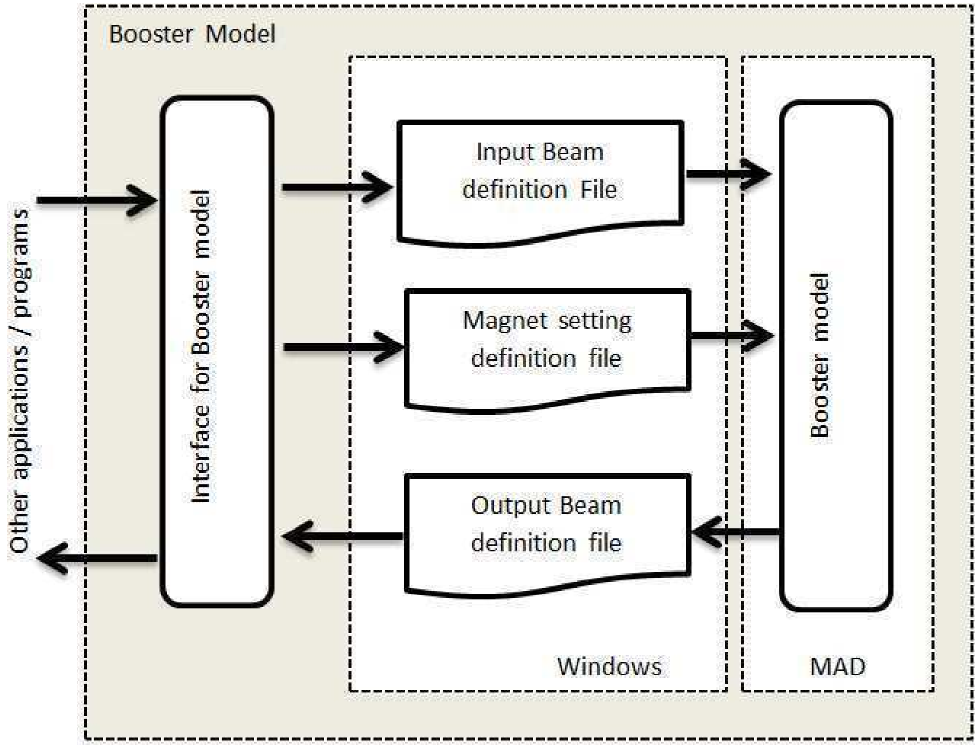


Figure 3.18: Booster model block diagram.

3.11.2 Module design

The MAD based model fulfils the physics related requirements of calculating survived/lost condition for the particle with the given attributes. But for integrating this MAD based booster model with complete accelerator simulation, an interface with this model is required so that other applications/modules can use this model for performing their calculations. A separate booster module is developed with the architecture shown in Figure 3.18 to add interface to the MAD based booster model. On receipt of a calculation request the wrapper program first directly modifies the 'Beam definition' file and 'Magnet setting' files according to the magnet setting data and beam data provided in the request message. It then executes the MAD compiler with the recently modified files. Finally it parses the output file generated by the MAD program to extract the required data and sends the data to the caller.

3.12 Validation of TL-1 and Booster model

3.12.1 System setup and experiment

For validating the TL-1 model it is required that one should be able to change the electron beam parameters at the input of TL-1. Then by measuring the beam parameters at the TL-1 output for different changes made to the TL-1 magnet settings, one can generate sufficient data for validation of the model. Similarly for validating the booster model it is required to change the electron beam parameters at the booster injection input point. Then by measuring the current in the booster one can generate the required data for booster model validation. But this cannot be done in the present machine due to the following reasons.

- The machine is built for serving as an injector system for INDUS-1 and INDUS-2 storage rings and thus the major change to the system are not allowed.
- The machine does not contain sufficient number of diagnostic devices to identify the beam parameters at the TL-1 output as this is not required in normal operation of the machine.
- There is no direct means to control the beam parameters at the TL-1 input.
- There is no direct means available to measure the parameters throughout the TL-1 or Injection point input.
- further to this the beam source is not always stable and sometimes changes during operation.

In the presence of above stated problems, one of the possible method for validating the models can be the combined identification of TL-1 and Booster models. In which the

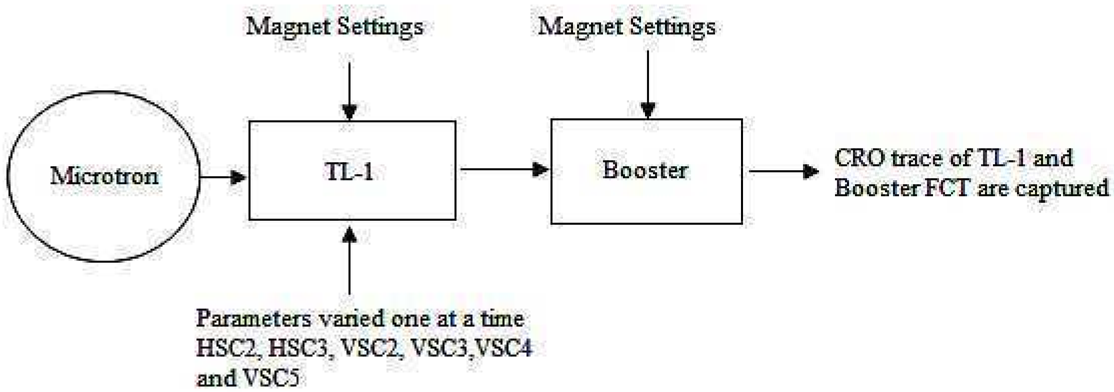


Figure 3.19: Setup for validation experiment.

output from TL-1 model can be given as the input to the Booster model and then output from booster model in the form of booster current can be used for validation against the measured Booster FCT current. Since the developed TL-1 and Booster models do not simulate the time dependent injection process and only uses the static values of kickers and septum magnet(value at 20μs after the trigger pulse), the absolute value of Booster current can not be used for validation. Thus we have used the normalised change in booster current as the validation criteria and validated the series combination of TL-1 and Booster model as shown in the Figure 3.19. Table 3.7(a) lists the set value for Booster magnets and Table 3.7(b) lists the set value for TL-1 magnets used during the experiment.

3.12.2 Procedure

To collect the validation data the following procedure is adopted.

- The Microtron is manually tuned to get the maximum FCT current with beam at reference position on Beam Slit Monitor (BSM).
- TL-1 is manually tuned to get the maximum FCT current at FCT2. The magnet settings of TL-1 at this operating point are listed in table 3.7(b).

Table 3.7: TL-1 and Booster magnet settings for the experiment.

(a) Booster

NAME	Set Value	Status
DCBASE	20.413	ON
DP-W1	25.024	ON
HSC-1	0.001	ON
HSC-2	-0.800	ON
HSC-3	0.900	ON
HSC-4	0.800	ON
HSC-5	0.480	ON
HSC-6	0.020	ON
QF-W2	1.220	ON
QD-W2	6.250	ON
VSC-1	0.000	OFF
VSC-2	0.098	ON
VSC-3	0.046	ON
VSC-4	0.140	ON
VSC-5	-0.004	ON
VSC-6	0.000	ON

(b) TL-1

NAME	Set Value	Status
QP-1	1.272	ON
QP-2	1.434	ON
QP-3	1.274	ON
QP-4	1.451	ON
DP-1	9.135	ON
HSC-1	0.000	ON
VSC-1	0.000	ON
VSC-2	0.000	ON
HSC-2	0.000	ON
HSC-3	0.100	ON
VSC-3	0.000	ON
VSC-4	0.000	ON
HSC-4	0.118	OFF

- The Booster is manually tuned to get the maximum FCT current in Booster. The magnet settings of Booster at this operating point are listed in table 3.7(a).
- The value of current settings in one of the steering magnet of TL-1, say VSC2 is increased and decreased in steps from its nominal operating value and the TL-1 FCT2 pulse and Booster FCT pulse are captured on CRO as CRO traces. The value of FCT, BR current at Inj value and BR current @3.3μs are calculated from the selected portions of these CRO traces as shown in the Figure 3.20.
- After completion of experiment the magnet set value is restored to its nominal operating value.
- This procedure is repeated to collect data for VSC2, VSC3, VSC4, VSC5, HSC2, HSC3 and HSC4.

The range for variation of magnet current are chosen such that it cover the variations in the measured BR FCT current from zero(when there is no current measured by BR FCT) to one (When BR FCT measures the maximum current), unless it is restricted by

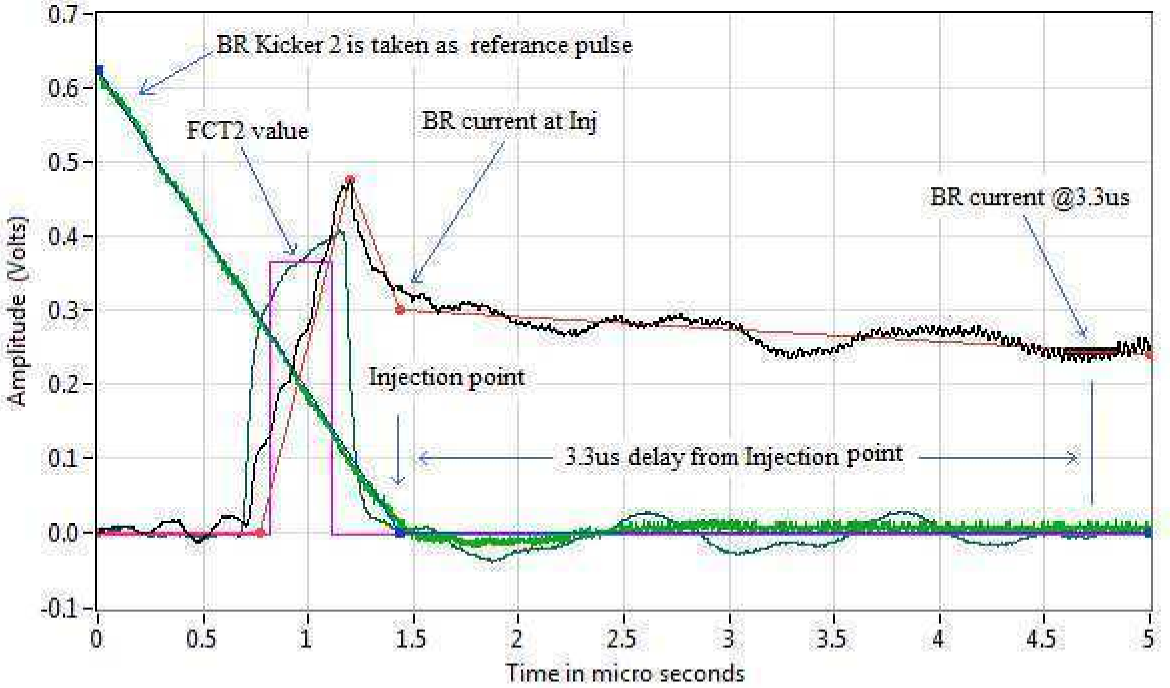


Figure 3.20: Measurement of FCT2, BR current at Inj and BR current @3.3μs parameters from CRO traces.

the magnet power supply current setting limits. All the BR current measurements, are then normalised with the maximum BR current observed for the case when no change is made to the corrector magnets. To remove the effect of pulse to pulse variations in the measurements, for each reading 10 measurements are made and average value of these measurements are taken as reading at that point. After collecting the data of injection current variations as a result of changes made in the set current of VSC2, VSC3, VSC4, VSC5, HSC2, HSC3 and HSC4. The same variation in currents are applied to the series configuration of models of TL-1 and Booster shown in Figure 3.19 and the variation in the injection current were calculated after 33 turns of revolution by beam in booster (that is after 3.3μs from injection point).

3.12.3 Results

The Figure 3.21 shows the plot of "BR current at inj", "BR inj @3.3μs"and injection current calculated from model. with respect to the variation in HSC2 set value. The injection value, measured as well as calculated are normalised before plotting. Similarly the Figure 3.23 to 3.27 shows the plot of "BR current at inj", "BR inj @3.3μs"and injection current computed using model for change in set values of HSC3,HSC4, VSC2, VSC3, VSC4 and VSC5. Table 3.8 list the calculated sum of square errors (SSE), R-square value , and root mean square error (RMSE) for the model output for "BR current at inj", "BR inj @3.3μs". The sum of square error is a measure of the discrepancy between the data and the estimated model. A small SSE value indicates a tight fit of the model to the data. SSE is calculated as:

$$SSE = \sum_{i=1}^{n} (y_i - f_i)^2 \tag{3.33}$$

where y_i is the i^{th} element of Y(measurement data set) and f_i is the i^{th} element of model computed fit. R-square value gives the goodness of fit for the model. It range from 0 to 1, with 1 representing a perfect fit between the data and the model computed values, and 0 representing no statistical correlation between the data and model computed values. It is computed as follows:

$$R - square = 1 - \frac{SSE}{SST} \tag{3.34}$$

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{3.35}$$

where y_i is the i^{th} element of Y(measurement data set) and $\bar{y}$ is the mean value of Y. The RMS error gives an estimate of the standard deviation of the random component in the data, and is computed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - f_i)^2}{n}} \tag{3.36}$$

Table 3.8: Goodness of fit for TL-1 and Booster model.

Name	BR current at Inj			BR current at @3.3μs		
	SSE	R-square	RMSE	SSE	R-square	RMSE
HSC2	0.10	0.96	0.07	0.03	0.98	0.04
HSC3	0.04	0.97	0.05	0.02	0.98	0.04
HSC4	0.31	0.76	0.15	0.31	0.76	0.15
VSC2	0.05	0.98	0.05	0.07	0.97	0.06
VSC3	0.04	0.98	0.05	0.01	0.99	0.02
VSC4	0.01	0.99	0.03	0.01	0.99	0.03
VSC5	0.18	0.89	0.10	0.22	0.85	0.11

where y_i is the i^{th} element of Y(measurement data set), f_i is the i^{th} element of model computed fit and n is the number of data points.

3.12.4 Discussion

The figures (figure 3.21 - 3.27) shows the Booster current at injection and the Booster current after 33 turns of beam revolution along with the model computed values for the cases when the corrector currents of TL-1 are varied. All these figures show the booster current in the changing acceptance scenario of the TL1 and booster synchrotron as a combined system.(Acceptance is related with the maximum deviated injected electron in position and in angle which can be successfully captured by the machine). The figure 3.21(a) shows the observed values for the Booster current at injection, Booster current after 33 turns of beam revolution and the model computed values for HSC2. From this figure it can be seen that both the curve(BR current at inj and BR current @3.3μs) shows large deviation from the model computed values at the sample points between HSC2 current deviation -0.06A to -0.02A respectively. This deviation is caused because of the sudden change in the Microtron operating condition that has resulted in the sudden drop in beam current output from Microtron. This is verified from the logged data of FCT shown in figure 3.21(b). Normalisation of the observed data in figure 3.21(a) with the FCT data shown in figure 3.21(b) gives the corrected curves shown in figure 3.21(c) The R-

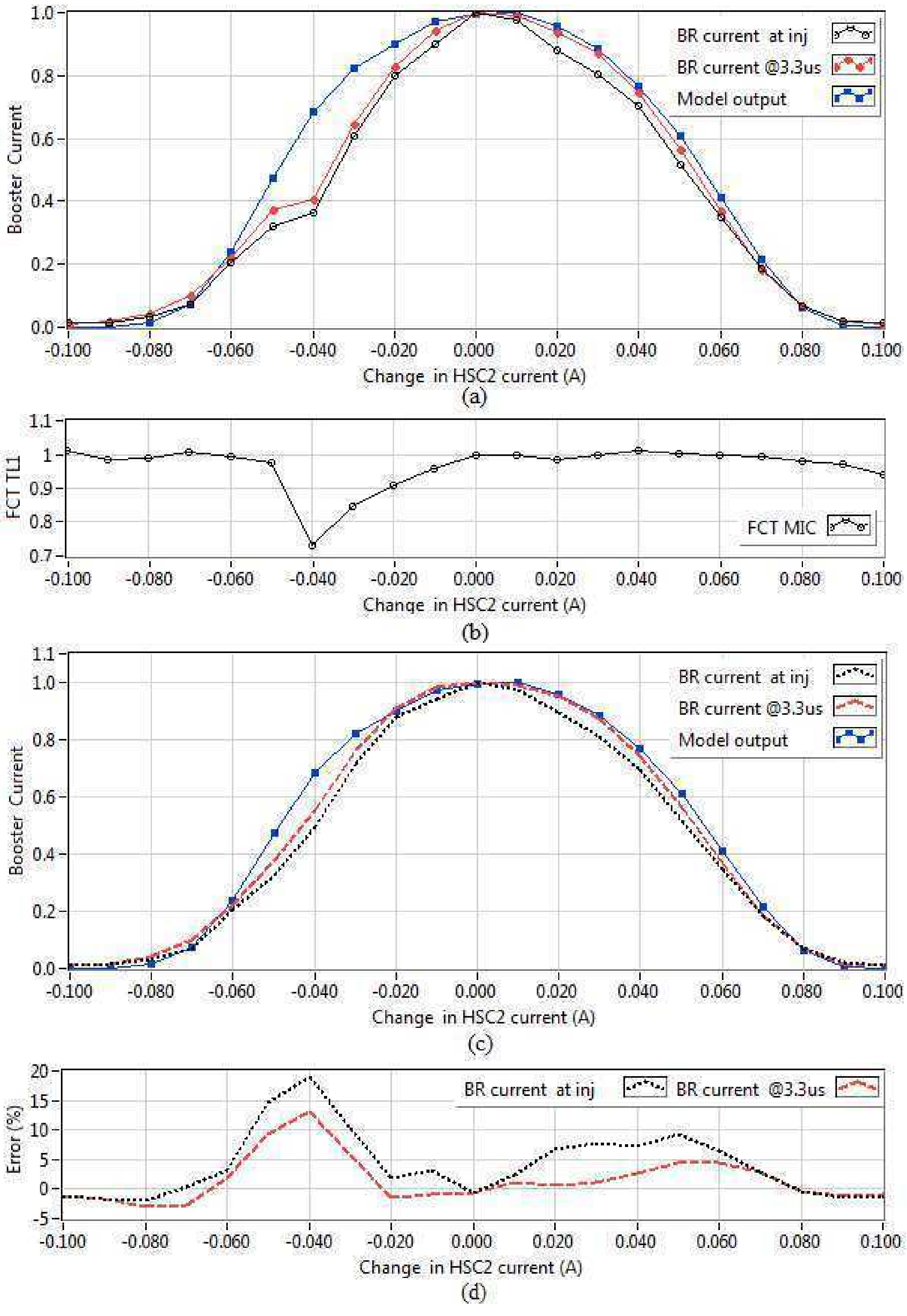


Figure 3.21: Plot of measured peak Inj, Inj @3.3μs and calculated Inj using model versus changes in HSC2 current settings.

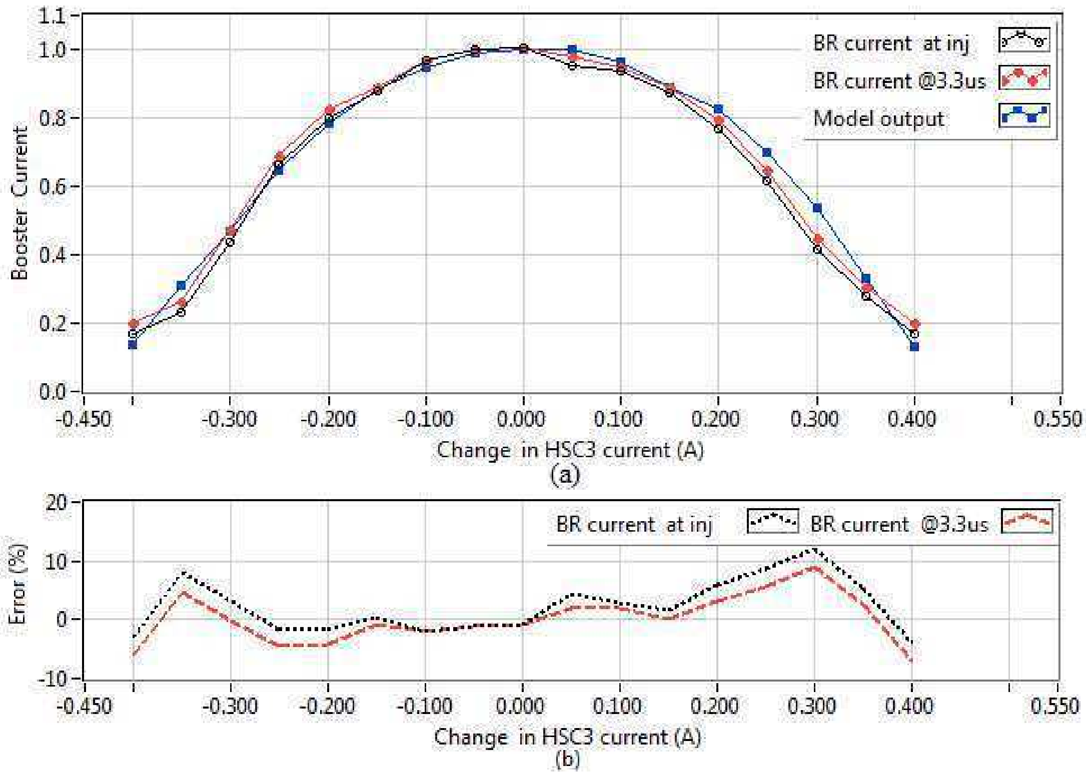


Figure 3.22: Plot of measured peak Inj, Inj @3.3μs and calculated Inj using model versus changes in HSC3 current settings.

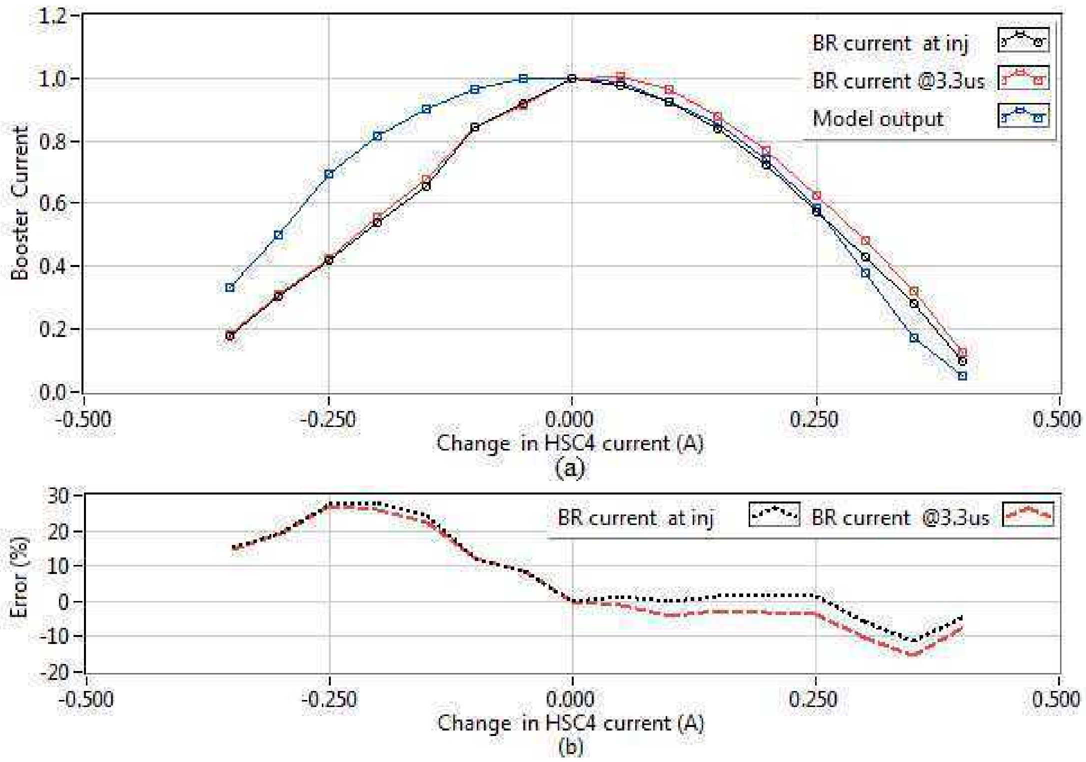


Figure 3.23: Plot of measured peak Inj, Inj @3.3μs and calculated Inj using model versus changes in HSC4 current settings.

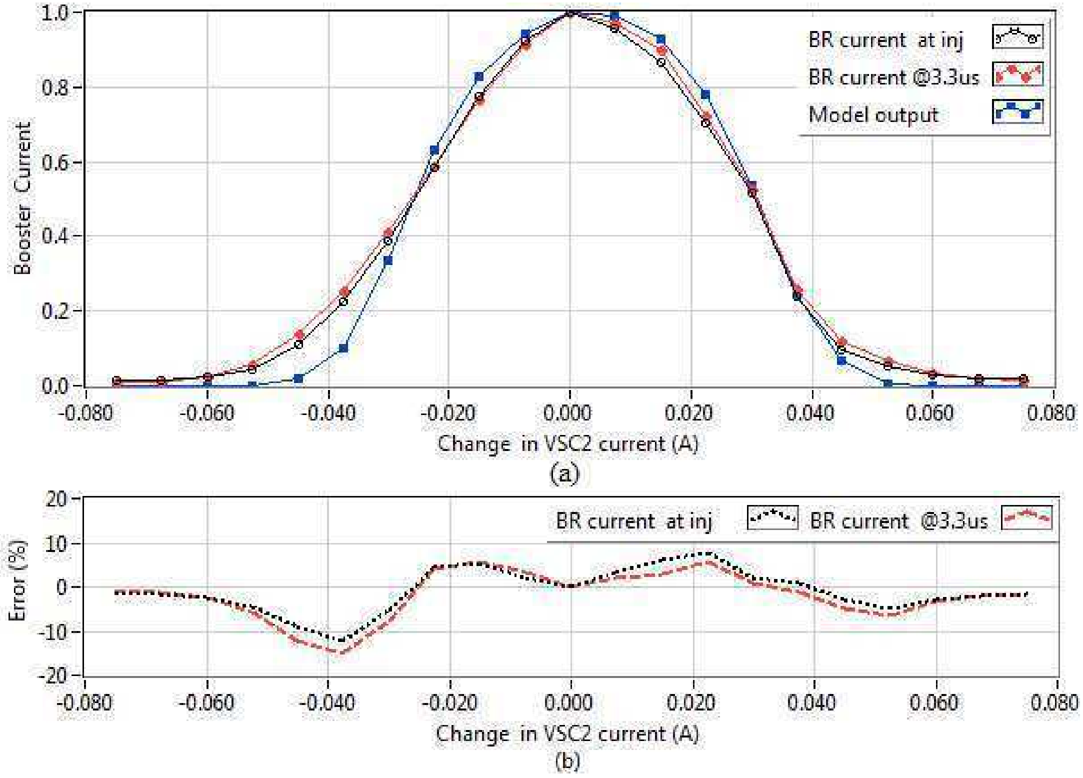


Figure 3.24: Plot of measured peak Inj, Inj @3.3μs and calculated Inj using model versus changes in VSC2 current settings.

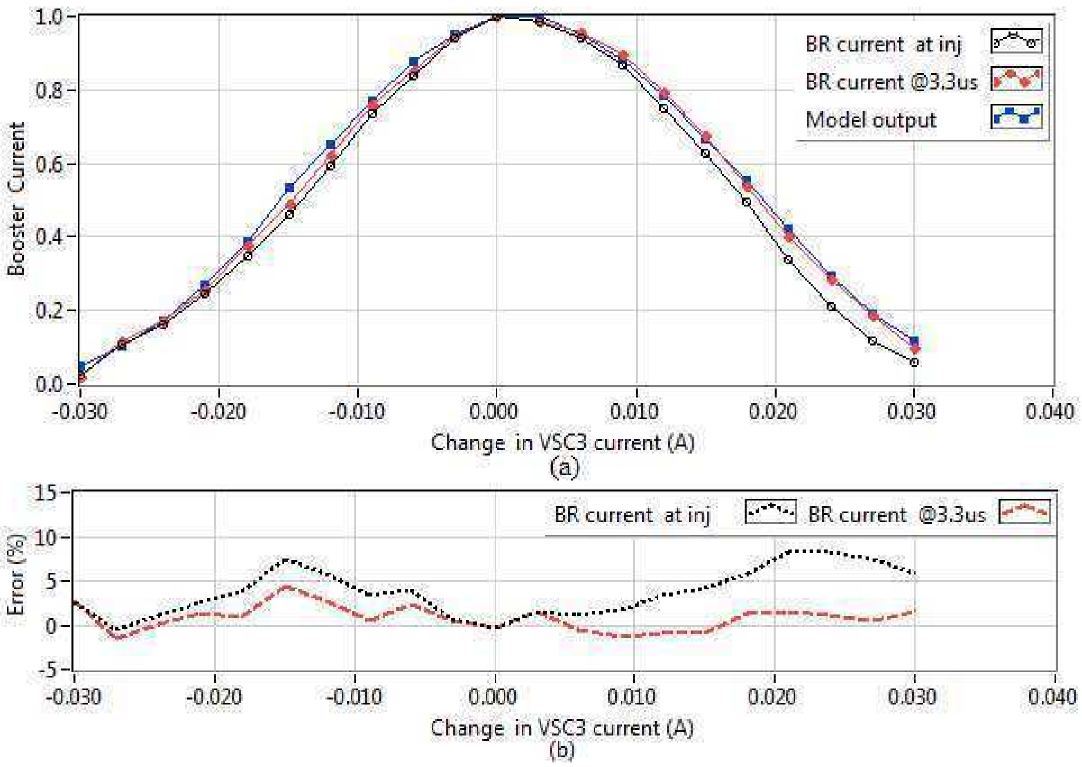


Figure 3.25: Plot of measured peak Inj, Inj @3.3μs and calculated Inj using model versus changes in VSC3 current settings.

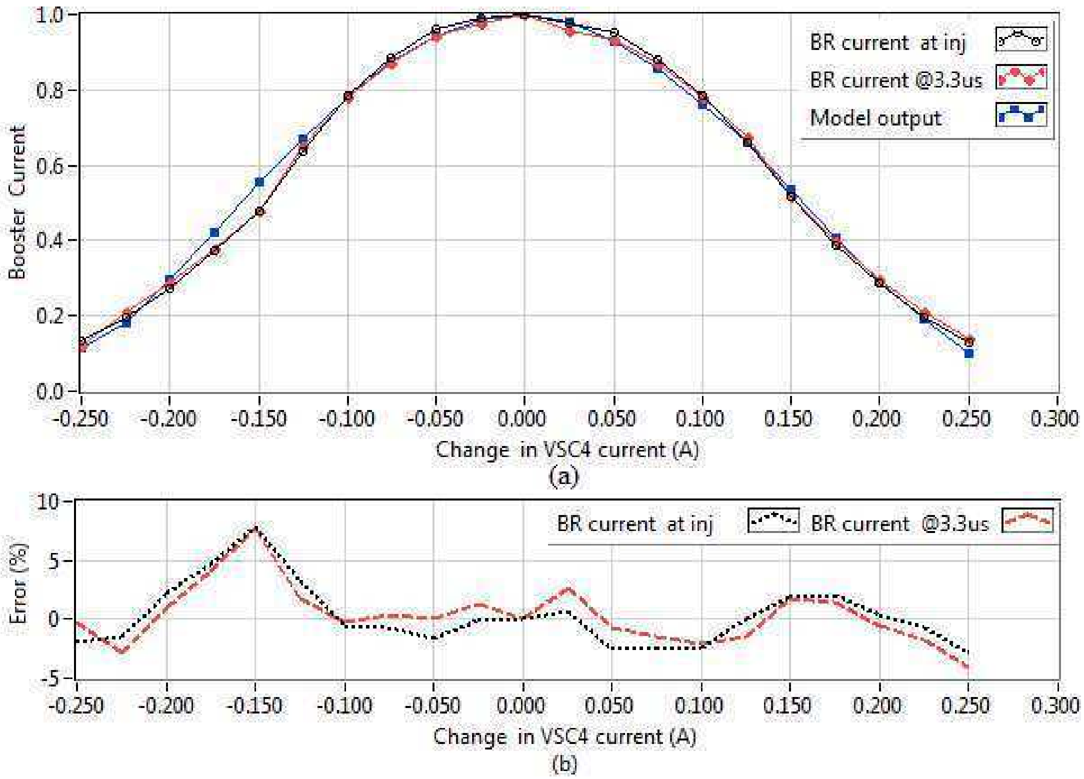


Figure 3.26: Plot of measured peak Inj, Inj @3.3μs and calculated Inj using model versus changes in VSC4 current settings.

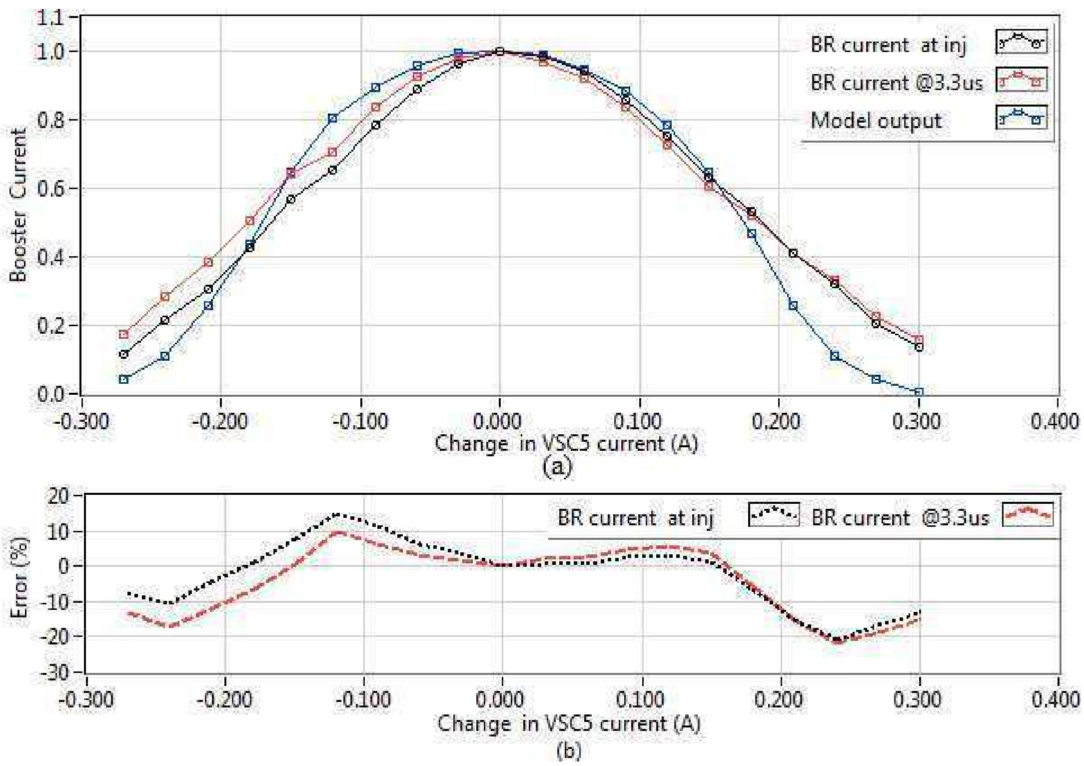


Figure 3.27: Plot of measured peak Inj, Inj @3.3μs and calculated Inj using model versus changes in VSC5 current settings.

square value for these corrected curves of "BR current at inj"and "BR current @3.3μs"are 0.96 and 0.98 respectively that gives a good model fitting value indicating that the model can explain about 96% to 98% of the total variations observed in the data. From Figure 3.21(d) we see that the absolute maximum error for both the curves is limited to less that 20%. Also the RMSE value of 7% and 4% for the "BR current at inj"and "BR current @3.3μs"shows that the developed model can predict the booster current variations quite accurately resulting by controlling HSC2.

Figure 3.22(a)shows the observed values for "BR current at inj"and "BR current @3.3μs"along with the model computed values for HSC3. The R-square value for these curves are computed as 0.97 and 0.98. The R-square value close to 1.0 along with the absolute maximum error less than 11% (figure 3.22(b)) indicates good agreement of model with the experimentally observed value. Further the RMSE value of 5% and 4% for the "BR current at inj"and "BR current @3.3μs"curves supports that the developed model can accurately predict the booster current variations if we use HSC3 for controlling.

Figure 3.23(a)shows the observed values for "BR current at inj", "BR current @3.3μs"and the model computed values for HSC4. Two observations are directly apparent in the HSC4 validation data curve. First: that experimentally observed values differ from the Model calculated values and this error(difference between model computed values and observed values) is larger in magnitude (maximum absolute error ≈26% (figure 3.23(b))) for the case when magnet current is decreased from the normally operated value as compared to the case when magnet current is increased. Second : that the slope of observed value curve also differs from the model computed values on the rising edge of curve (for the case when magnet current is decreased from the normally operated value). Since the developed model can not explains the reason for this observed behaviour, experiment was repeated and the data log for all the parameters of Microtron and TL-1 were checked for any abnormalities and was finally concluded that the observed behaviour is the true behaviour of the system and is not arising due to any erroneous condition. Finally on inspection of the HSC4 coil at site the reason for this abnormality is found as associated locally to the HSC4 magnet. On visually inspecting the HSC4 magnet it is found that due to geometrical constraints the HSC4 magnet is installed in TL-1 at close proximity of Booster Dipole magnet(Dipole coil is only at 80mm away from the HSC4 magnet edge, see picture 3.28). As the magnetic field produce by this Dipole magnets of booster synchrotron is very stronger as compared to the field produced by HSC4 magnet, the magnetic field of HSC4 magnet gets modified. Although the quantification of magnetic field perturbation for this case is not in the scope of this work but from literature [108] it can be said that these perturbation may be sufficient enough to produce the observed errors from the model. The R-square value of 0.76 and the RMSE value of 15% for this case shows that the developed model only predicts the booster current variation with moderate accuracy.

Figure 3.24(a), 3.25(a) and 3.26(a) shows the validation experiment data for VSC2, VSC3 and VSC4. The R-square value (Table 3.8) for all these curves is between 0.97 to 0.99 and the maximum absolute error (figure 3.24(b), 3.25(b) and 3.26(b)) is between 7% to 15% that shows the good agreement of model with the experimentally observed values for these magnets. Also the RMSE value between 2% to 6% for these curves shows that the developed model predicts the booster current variations accurately if we use VSC2, VSC3 and VSC4 as control elements.

The validation experiment data for VSC5 which is the last magnet in the TL-1 is shown in figure 3.27(a). The R-square value of 0.85 and the RMSE value of 11% shows that the model can only predict the booster current variations with moderate accuracy for this case. Moreover in figure 3.27(a) it can be seen that the model shows good match for

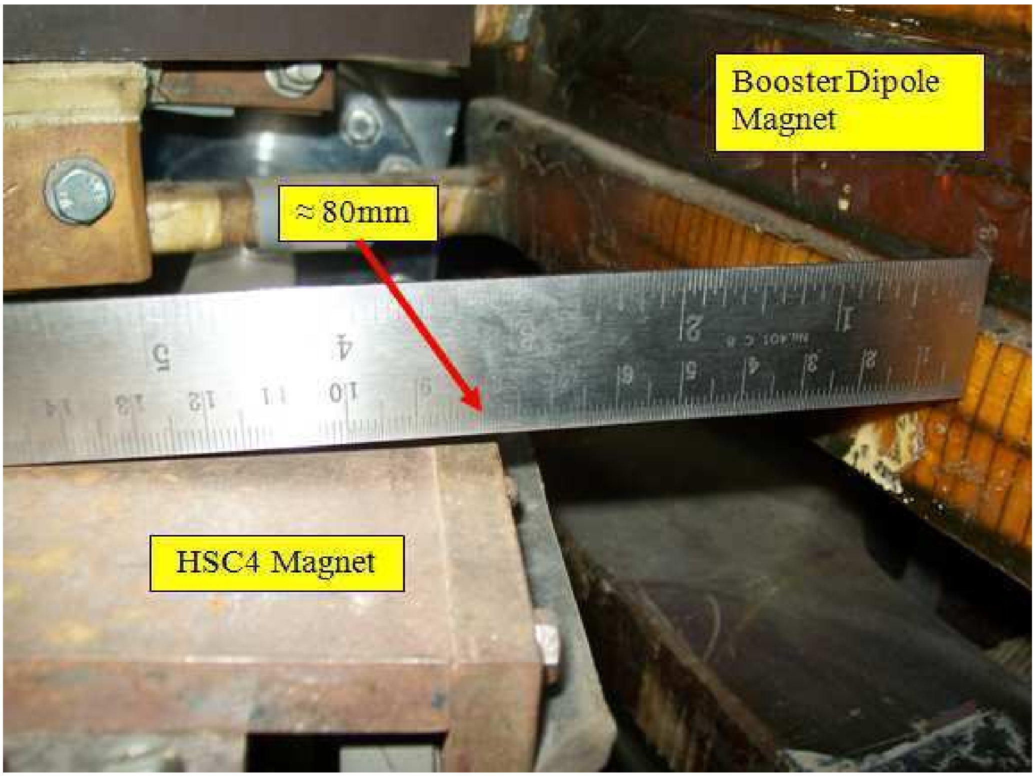


Figure 3.28: Picture showing the HSC4 magnet distance from the Booster Dipole magnet coil.

the case when magnet current deviations are small, but for that case when magnet current deviations are large that should resulting in large beam position deflections, the error in model predicted values increases (figure 3.27(b)). The reason for this abnormality is found associated locally to the VSC5 magnet on visual inspection at site. Due to mounting space restrictions, this coil is found mounted on the Septum Magnet chamber (picture 3.29). Septum is a fast pulsed magnet that generates high magnetic field. It also produces eddy currents in the septum chamber. In order to protect the rest of the ring from the effect of eddy currents produced by septum, the septum chamber is isolated using ceramic isolator at the flanges where septum chamber joins with the beam pipe. Whereas in the present case the VSC5 magnet is mounted directly on the septum chamber thus there are two effects that are perturbing the field if VSC5, one is the septum field itself and second is the eddy current generated by the septum in the beam chamber that can produce the sextuple

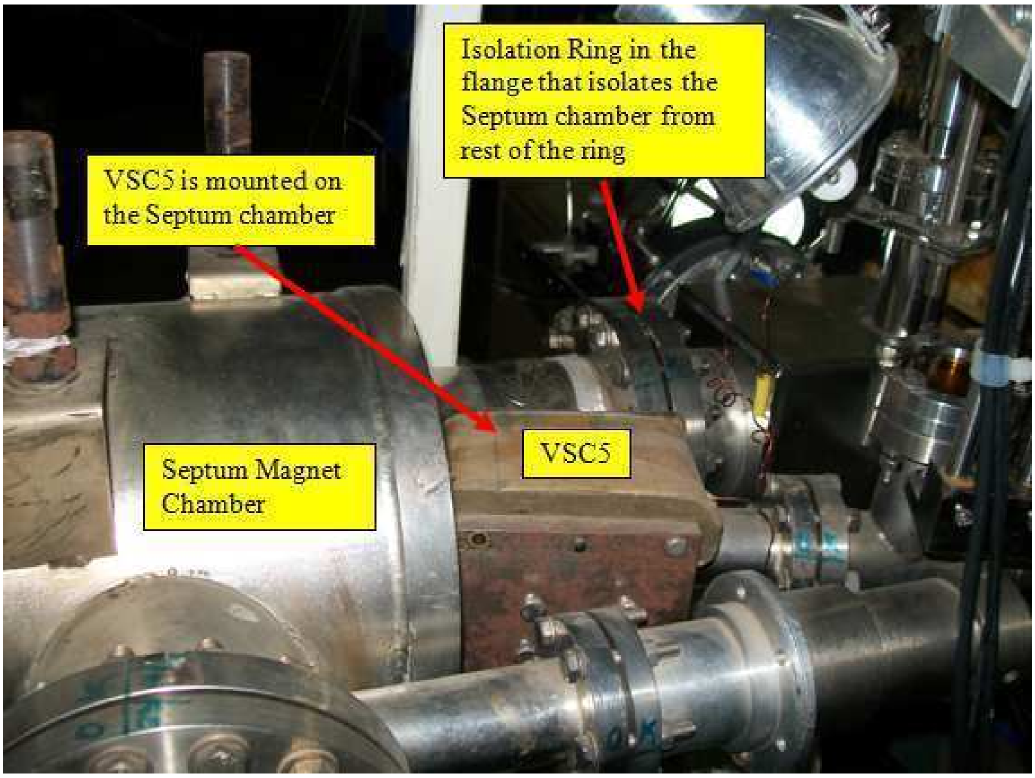


Figure 3.29: Picture showing the VSC5 magnet mounting on the septum chamber.

field ([25] page no 112-113). Both of these effects are the main reasons for the observed large error between model predicted and observed values for this magnet. The results of the model validation experiments shows that if we exclude the case of HSC4 and VSC5 for which there are reasons to cause the observed deviations in the model computed values, then the model gives R-square values better than 0.97 and the RMS error is limited to a maximum value of 7% (Table 3.8). This gives the confidence that the model can be used for representing the machine components TL-1 and Booster for simulations.

3.13 Summary and conclusion

This chapter introduces the particle accelerators and different concepts used in acceleration process. Specifically the synchrotron accelerators and their use at synchrotron radiation sources facilities along with their different subsystems are discussed. Starting from the coordinate system, the particle motion in magnetic lattice and different accelerator physics parameters necessary for accelerator modelling are explained. In particular the accelerator component transport lines and booster ring are discussed. The physics of transport lines along with the electron beam model and electron beam transfer matrices for different transport line components (magnets) are discussed. Different simulation codes available for accelerator subsystem modelling are discussed with their merits and demerits. As a case study and for evaluating the proposed agent based control system scheme for accelerator control system, the model of different accelerator subsystems at Indus-1 complex naming TL-1 and Booster were developed with an emphasis on developing the model that describes the electron beam propagation in TL-1 and electron beam injection in booster. This model behaves like the actual machine producing the outputs (beam image at BPM's and Booster current) similar to the one available from actual TL-1 and Booster ring in response to the actual control parameters available in the TL-1 (HSC and VSC) and Booster ring (different magnet settings of booster). Thus the output of this work has provided us with the TL-1 and booster model that has interface similar to the actual machine interface that enable us to use it in studies on accelerator control system simulations related to Indus-1. The validation experiments performed over the machine serves two purposes; first it validates that we have modeled the system with the sufficient details and that our assumption of taking the Microtron beam with Gaussian profile in transverse plane is justified, secondly the results brings out the model portions (that uses control parameters HSC2, HSC3, VSC2, VSC3 and VSC4) that are in good agreement with the experimentally observed system. Thus becomes the preferred choice as the control parameters for the next work of agent based automatic control system implementation. The results also indicates the weak model portions (that uses control parameters HSC4 and VSC5) that if selected as control inputs may produce inaccuracy in the case of automatic control system implementation and indicates that if these control variables are found necessary to be included for control then the model must be improved by considering the stated reasons responsible for modelling errors. This developed model is now going to be used for the evaluation of the agent based control scheme in later chapters.

Chapter 4

Accelerator Control Systems

Synchrotron radiation sources are the accelerator facilities built for the purpose of producing Synchrotron Radiation to be used in different scientific and industrial applications. Control system of these machines have evolved from small hard-wired systems to complex computer controlled systems with many types of graphical user interfaces and electronic data processing. With distributed layered architecture of control system these facilities have successfully fulfilled the specific operational requirements demanded by different classes of users and different technical requirements imposed by various subsystems of these large facilities. With the development of open source toolkit based Supervisory Control and Data Acquisition (SCADA) systems like Experiment Physics Industrial Control System (EPICS) and TAco Next Generation Objects (TANGO) the collaborative efforts have benefited the existing and upcoming third generation source control systems in terms of reduced development costs and timeline with increased reliability due to the high degree of software reuse. The next generation of synchrotron radiation sources will require much bigger and better control systems. The advances in technology can support the network bandwidth and CPU power required for reasonable update rates and requisite timings. Beyond the scaling problem, next generation systems face additional challenges due to growing cyber security threats and the likelihood that some degree of remote development and operation will be required.

4.1 Introduction

The particle accelerators started appearing around 1930. The first accelerator used for accelerating protons to energy of 400KeV (Cockcroft & Walton split lithium atom for the first time with this accelerator) was built with few sub systems like H.V. generator, proton source, vacuum pump, lithium target and scintillation screen [15, 21]. It was fully manually controlled during experiments and experimenter had to sit in observation cubicle (experimental area) immediately below the acceleration tube as shown in Picture 4.1 (taken from reference [104]). The whole accelerator was housed in a single room.

With advancements in various technologies much bigger accelerators are now being made with a large number of subsystems spread over large geographical areas. Much more complex instruments and machines are to be run in a synchronized and sequential manner for their operation. Also the number of parameters to be controlled and monitored during the experiments have increased to such an extent that the manual control and observation is beyond the scope of human capabilities. This has led to the development of accelerator control systems in accelerator facilities. In today's accelerator facilities almost all the operations are exercised remotely and centrally with operators sitting in control room and working from operator console PCs with the help of Graphical User Interfaces (GUI) running on them. Synchrotron Radiation Sources (SRS) are built around the accelerators facilities and comprises of a pre-injector (small accelerator usually Linac or Microtron), beam transport lines, booster (bigger accelerator usually synchrotron) and storage ring. Thus they comprise of many accelerators and hence their control system has evolved out of accelerator control system. The first generation SRS machines were the electron

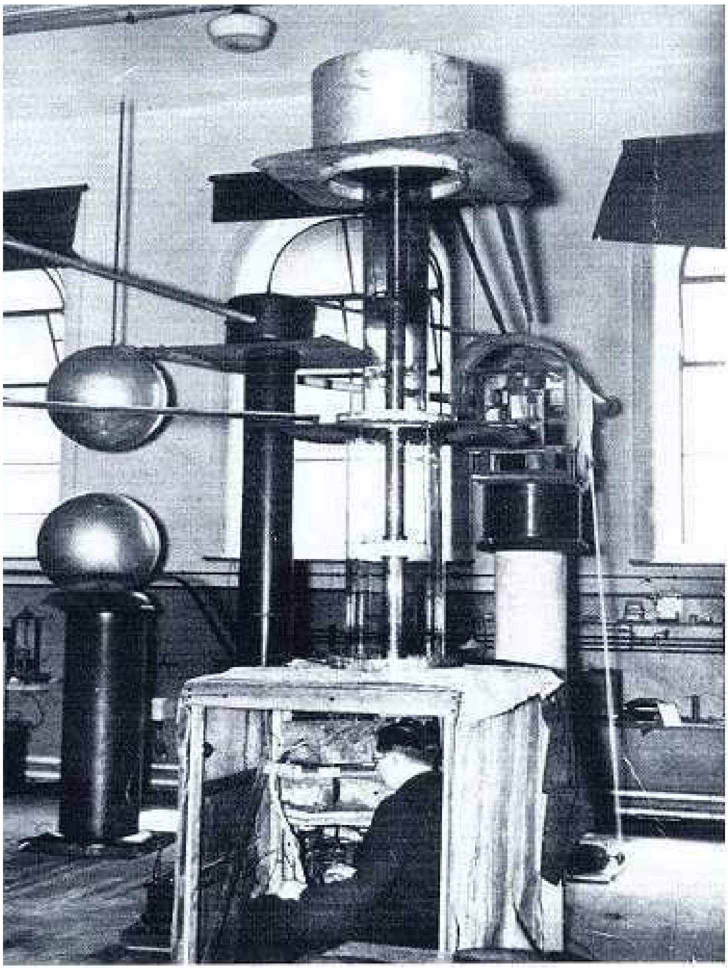


Figure 4.1: The original Cockcroft-Walton installation at the Cavendish Laboratory in Cambridge [104] Walton is sitting in the observation cubicle (experimental area) immediately below the acceleration tube, which was covered with black velvet so that the faint scintillations might be observed by the detector.

storage rings built specifically to store continuously circulating electron beam at a fixed energy for periods up to many hours. The control system of some of these early machines were analogue type but with the availability of computers in market around the same time control systems of accelerators started using minicomputers with Central Processing Units (CPU) at the upper layer (operator interface layer) and CAMAC compliant I/O cards in CAMAC crates at lower layer. The second generation of synchrotron radiation sources started around 1980 where the accelerator and storage rings with modified lattice structure were built to attain the increased brightness by minimizing the electron beam emittance. With the use of microprocessor for most of these facilities, control system architecture still remain two layered with much power full computers at the upper layer and low capacity computers at lower layers. The use of Real time operating systems started appearing during this time at the lower layer controllers where as the upper layer computers mostly used the proprietary OS to the concerned computers. The applications were built using the concept of distributed databases, device description tables, device tables with about 2500 parameters and data archiving rates of once per 2 minutes were developed. The development languages used were mainly C, FORTRAN and Assembly, and the databases used were of text type and relational databases. Some of the Facilities developed around 1985 for the first time used the VME crates at the middle layer and were among the first to adopt the three layer architecture. the software programs at layer one started using the new concepts like event based programming, inter program communication facilities and were based on enhanced graphical plotting abilities (bar graphs, joined line graphs, scatter plots). The operator interfaces were equipped with Alphanumeric terminals, Color TV-raster scan with interactive cursors, track boll, computer controlled knobs with incremental encoders and the facility of multi-parameter control linked with single computer knob used for example for producing bump on closed orbit. The third generation synchrotron radiation sources were the machines specifically built with the motto of providing higher brightness and to accommodate large number of insertion devices. Theses machines are the modern present day synchrotron light sources. They have the most advanced control systems and some of them also provide the facility of top-up injection. In these facilities the computer controlled system normally comprises of two parts. The first part is the hardware that directly interfaces with the sensors and actuators. Since the accelerators are spread over a large area and comprise of many subsystems, normally this hardware part is spread over one or two layers with many small

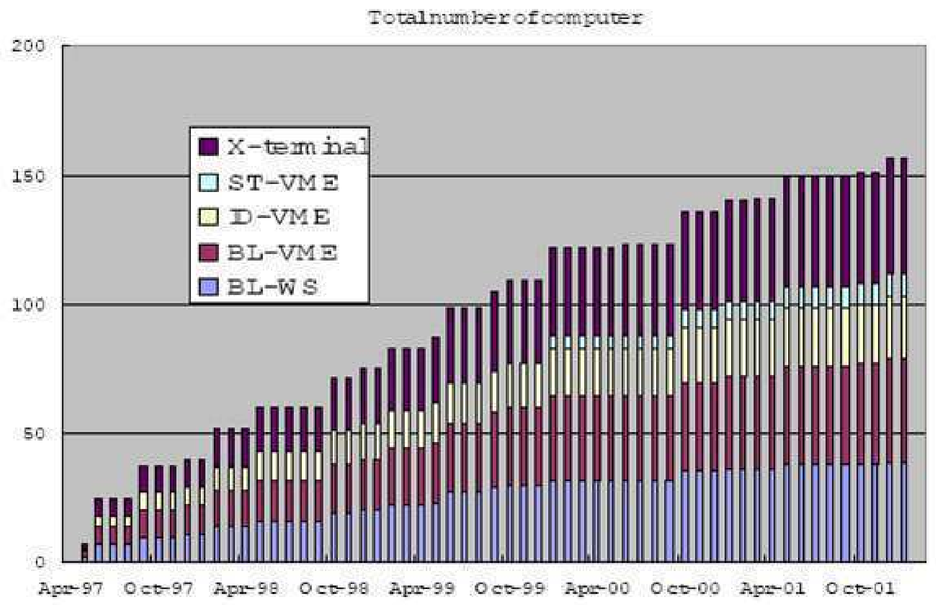


Figure 4.2: Change in the total number of computers in the control system of Spring-8 facility over the time period in the Beamline user area (taken from reference [115]).

computers/controllers connected over field-buses constituting the lower layer and higher computing capacity based computers constituting the upper layer. One or more upper

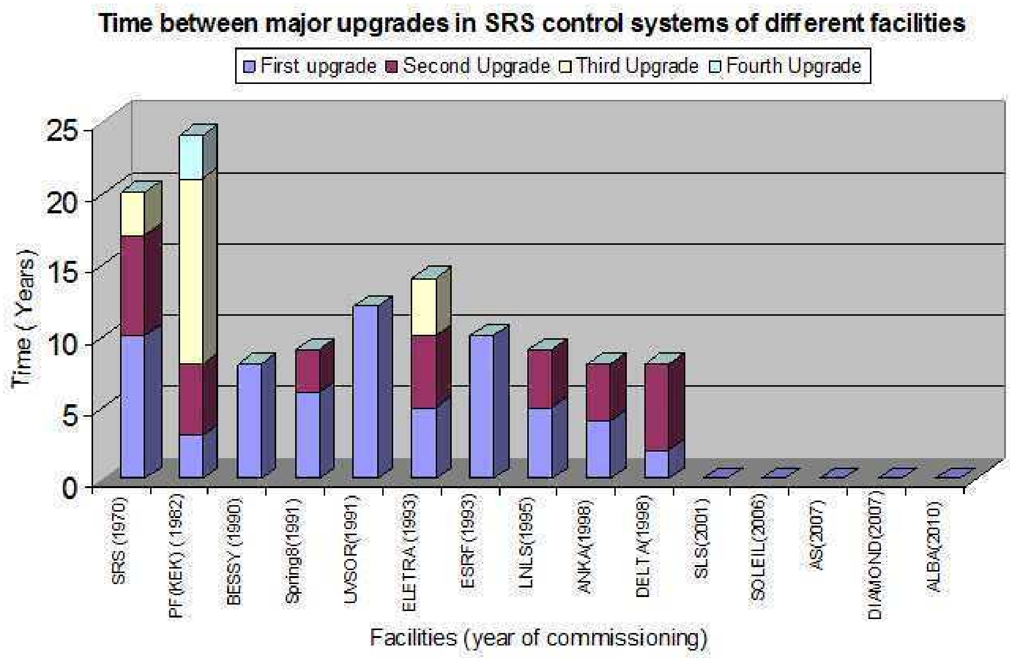


Figure 4.3: Time between major upgrades in SRS control systems of different facilities.

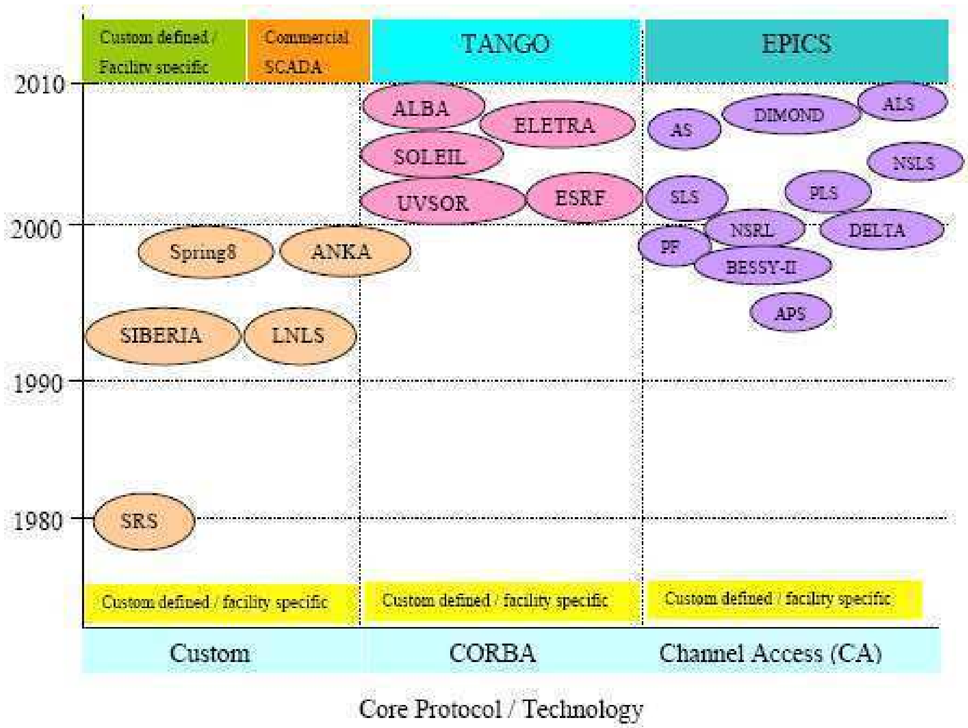


Figure 4.4: The control system framework adopted by different facilities.

layer computers logically combine to make the control system for one sub-system. The control systems of various sub-systems together make the overall hardware part of control system for the complete facility. The second part is the software part that comprises of the lower layer subroutines running at the lower layer controllers, the control scripts and Graphical User Interface algorithm running at the operator PC consoles. The control system of theses machines has undergone a continuous phase of extension after commissioning to accommodate the signals arising from the large number of experimental station beam lines and insertion devices of increasing complexity as most of theses facilities were initially commissioned with few beam lines and later on the users added the new beam lines on demand, for example figure 4.2 shows the increase in the total number of computers in the control system of Spring-8 facility over the time period in the Beamline user area. With the increase in the computing powers of computer and the changing electronic market scenarios the control system of many facilities has seen upgrades at different times, figure 4.3 shows the time between major upgrades in SRS control systems of different facilities. The SRS control system has passed through many technological changes from 1970 to 2010 and the present trends shows the growth around two main technologies first EPICS and the second TANGO figure 4.4 shows the trend in the control system technologies adopted by different sources. Figure 4.5 shows the evolution of different concepts and technologies in SRS control system with time. With the increase in the computing power and lowering of prices of electronic components and computers the present day control system of the new facilities are again moving from three layer architecture to two layer architecture with the fast computers directly provided at lower equipment interface layer.

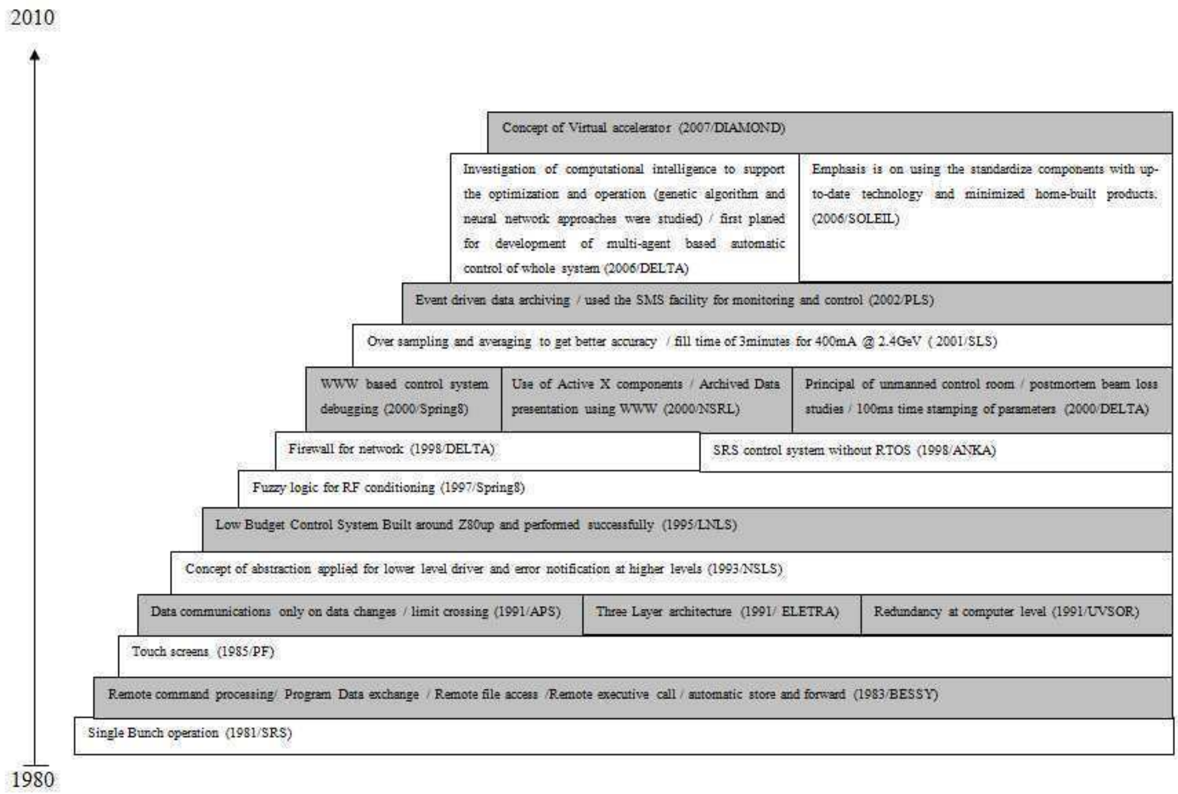


Figure 4.5: The evolution of different concepts and technologies in SRS control system with time.

4.2 Scope of SRS control system

The control system of today's SRS facilities is designed to monitor and control model-based and computed data from the accelerator, allied facilities, experimental, safety and operating subsystems to accomplish supervisory control, automation and operational analysis. The scope of the control system extends from the interface of the equipment being controlled through to the designers and operators of the accelerator facility, as well as synchrotron beamline users and staff. The control system addresses the aspects like timing, deterministic data communication, network communication, control room operations, automation and optimization. It comprises of the computers and software required to implement and integrate all subsystems including beam diagnostics, power supplies, low level RF, vacuum, personnel protection, equipment protection, insertion devices like undulator and wigglers, experimental beamlines and conventional facilities. In order to provide this comprehensive monitoring, control and automation, the control systems need to be scalable to support thousands of physical input/output connections and computed variables that can be correlated to analyse events and provide data for all control aspects.

4.3 SRS control system requirements

4.3.1 General conclusions

The actual technical details for control system requirements differ from facility to facility but on examining the various papers on the control system for accelerators some general conclusions can be drawn.

1. The control system in accelerator facility is a support activity. In many cases of installation as well as up-gradation there is always a lack of budget, manpower and the most important, the lack of machine down time and the control engineers are

forced to complete the upgrade process without sacrificing the machine operation time.

2. As Accelerator machines are in research environment there is a continuous requirement for new features/ up-gradation in control system mainly arising from operation side and at times due to up-gradation or addition of new hardware components to the facility.

3. For some of the specific requirements such as fast timing system etc., off-the-shelf solutions are not always available and some development is always needed.

4. Control system engineers are often not involved in early phases of machine design. This seriously affects the over all performance of the control system developed later as there is not much left in his hands towards standardization of the interfaces with the machine components already produced or under production. This compels the control engineers to adopt the complex architecture loosely coupled and with a large number of different types of electronic boards which are difficult to maintain and upgrade through out the life span of the facility. One has to understand that it is only the control system which is connected with all the subsystems and has to take care of all the general and specific requirements of all the sub-systems collectively whereas all other systems only have to perform their specific jobs.

5. From figure 4.3 It is clearly seen that only a few of the control systems have served without up-grade throughout the life span of the facility. The major reasons behind the upgrades are as below:

    (a) The continuous process of software upgrades in the form of software patches reduces the software reliability due to addition of bugs and increase the debugging time.

(b) In present time where the technologies are changing very fast, the lack of staff often poses the maintenance problem.

(c) The fast growing technological changes especially in the field of computers and microprocessor technology in the form of more and more computing power and with additional new features lured the engineers as the implementation of many previously thought functions now become possible.

(d) With the fast growing technology many devices and development tools become obsolete or are on the verge of obsolescent thus the fear of maintaining a obsolete technology / devices without industry support sometimes forces the engineers to chose for upgrade even when the deployed technology /devices suffice their purpose.

(e) Often the maintenance of the old technology becomes more costly affair as compared to the new off-the-shelf available products.

(f) Sometimes the need of participating in the development and becoming a part in the up-to-date technology in accelerator control system by means of collaborations with other facilities, as this is one of the ways to cope with the problem of limited man power, also strongly advocates the system upgrade with reasonably low cost.

6. Although the hardware architecture of almost all facilities can be grouped in two categories (two layered or three layered architecture) but the software architecture of most of them differ from each other.

7. Previously large variation is observed in development tools used by different facilities even when they were developed around the same time. Though with the availability of open source EPICS the trend seems to concentrate towards the common tools. (

Figure 4.4)

8. Many facilities have used Real Time Operating System (RTOS) at lower layer i.e. front ends in past and many more are using at present also. But some facilities questioned this and have demonstrated the successful operation of control system without RTOS at any layer. (For example ANKA, DAFNE, ESRF, and LNLS).

9. Different synchrotron radiation sources facilities have chosen their operating electron energy considering various aspects like, demand from the user, maximum potential of research in material sciences, design emittance and brightness, allotted budget etc. Figure 4.6 shows the operating electron energies for different SRS facilities. In the figure it can be seen that the SRS facilities commissioned before 2000 shows a large variation in choosing their operating electron energy spreading from few hundreds of MeV to 8GeV. Whereas for the SRS facilities commissioned after 2000 clearly shows the narrow spread in selecting their operating electron energy with almost all the facilities have chosen the electron energy levels between 2.5GeV to 3 GeV. Thus this indicates that the electron energy range between 2.5GeV to 3Gev is the range that fulfils the current user demand and is mostly be preferred by the new upcoming machines in near future.

10. The evolution of control system shows the continuous developments and inclusion of new ideas based on common facilities developed for society. For example the use of WWW and SMS facility for accelerator control debugging and fault removal from remote places in case the expert has gone to some distinct places. (Figure 4.5)

From above it is evident that the control system must be modular, incrementally upgradeable, scaleable and extendable. Expansion of the control system to accommodate the build-up of the accelerator and Beamlines from early testing, through installation

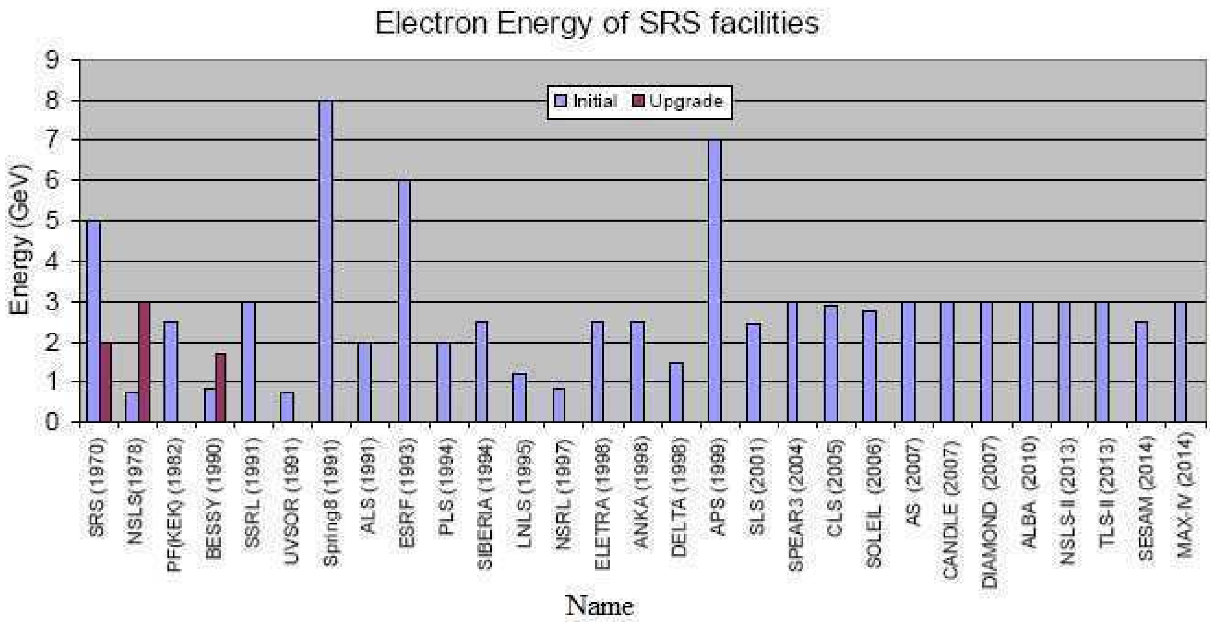


Figure 4.6: Electron energy of synchrotron radiation sources.

and commissioning, and during the life of the facility should not impact the performance. The control system must be available to support all aspects of the project schedule; from component tests during prototyping to beam characterization and optimization at commissioning.

4.3.2 Technical Requirements

Technical requirements differ from machine to machine; some of the technical requirements expected from third generation SRS control system facilities are presented below. [88]

1. Support for about 100,000 direct parameters and about 30,000 derived parameters.

2. Support for about 1 or 2 Hz model-based control (used to correct steering, the orbit, tune, linear chromaticity, optics etc).

3. Synchronous power supply and RF ramping, where the power supplies may be spread over a large geographical area. Some rings are having circumference in Km (for

example APS with circumference of 1104m)

4. Event based timing system with jitter less than 100ps and resolution down to 20 ns. Synchronization signals are to be supplied to different subsystems such as RF and P/S located at different geographical areas.

5. 500 MHz RF control Amplitude, phase, synchronization, etc,

6. 5 KHz fast orbit feedback (for coherent bunch instabilities) and about 50 to 100 Hz global and local orbit feedback.

7. About 20 to 200 millisecond equipment protection system.

8. About 5 Hz updates to operators of up to 1,000 chosen parameters.

9. Coherent turn-by-turn orbit data for up to $2^{10} = 1{,}024$ consecutive turns (for FFT).

10. Archive up to 6,000 parameters at a rate of 0.5 Hz continually.

11. Latch the last 10 seconds of data from all parameters in the storage ring when a fault is detected in the Machine Protection System (MPS), for postmortem analysis.

12. Archive up to 1,024 consecutive turn by turn data for 1,000 parameters at a rate of 10 Hz.

13. Provide pulse-to-pulse beam steering in the injectors (Linac or small accelerators) at 1 Hz.

14. For improved overall reliability the use of proven technology such as uninterruptible power supply (UPS), programmable logic controllers (PLC), battery backup, and redundant power supplies for Equipment crates are required for control of some of the critical sub systems such as the cryogenic, machine safety and personal safety

systems. Further to this to reduce the machine down time, control system modifications that add new functionality are performed during scheduled down times and after operational testing on equipment test stands before installation.

15. To address different class of users the system must manage the *Access Control* requirements to guarantee security of its computers and network systems. At the same time the overall system should be well integrated irrespective of the source of data origin, for example, it should be possible to display data from the control system from the associated relational database and from an accelerator model on one full-screen synoptic.

16. The overall software framework should be designed with well defined interfaces to support the modular upgrade/replacement of code to enable future upgrades in an economical way.

17. For supporting the hassle-free reconfigurability of the systems all the system design and configuration data of accelerator components and signal lists such as: magnetic lengths, min/max currents, calibration coefficients for currents vs. gradients, diagnostics channels, and configuration parameters should be managed version wise.

18. For future maintenance and upgrades, the control system hardware should have a layered structure. The functionality implemented at layers should be tried to be self consistent and built with standard interfaces so that the modification in one layer does not affect the other layers.

19. The networking hardware should be selected such that it is easily scalable, supports redundant configuration and include provision for network evolution. Where ever possible the network traffic should be isolated between different network layers using switches. Network security is to be implemented through physical security that

limits access to the control network from outside using gateways and firewalls.

20. The presentation layer comprising of operator's consoles, database manager, simulation computer, alarm generation/recording, data logging, displays and a gateway to the local-area network should be built with the features that distinctly addressed the individual requirements of different user classes such as accelerator operators, accelerator physicists, Technical group, Beamline Staff & Experimenters, Control System Engineers and Facility Managers.

    (a) Accelerator operators should be provided with a complete and consistent interface. The data presentation should be logical and rational to support easy equipment behaviour identification. Further to this the operation of the accelerators requires real-time control and monitoring of the equipment, archiving, alarm handling, sequencing, backup and restore for routine operation. For operators the alarm and error messages should be supported by information regarding recommended courses of action. The control system should allow the automation of plant operating tasks. It should provide applications that encourage and facilitate the keeping and passing of operation logs, particularly from shift to shift.

    (b) For accelerator physicists the control system should provide methods to integrate different accelerator models with the system. Functionality is required to allow easy acquisition of data produced as part of an experimental run, and to provide the ability to switch between different accelerator models. Data retrieved from the control system must be acquired with sufficient time accuracy to enable accurate correlation.

    (c) For technical groups all the diagnostic information necessary to assist in commissioning and debugging of equipment should be provided through easy interface. An easy interface to databases of equipment properties, manufacturers, documentation, cabling data and fault histories is required, as well as access to information clearly identifying the geographical location of equipment and a system of fault prediction facilities to allow for scheduled maintenance of components likely to fail.

(d) For beamline staff and experimenters the machine data needed during experiment should be provided.This is particularly necessary in the case of synchronizing scanning of a sample with changing of a parameter on an insertion device in the storage ring e.g. the gap of an undulator. Experimenters require clear information on light source status, performance, and timing signals, and may require remote access (i.e., from off site) to experiments and beam-lines.

(e) Control system engineers require current and archived data on the status and behaviour of the entire control system. Information required includes CPU loading, network loading, application monitoring (for frozen/crashed applications), connectivity status and reports of any control system faults.

21. The control system must include a relational database as a central repository for all configuration information. This should include all static information about accelerator components such as coefficients to calculate field magnetic strength from current. Consideration should be given to extending the database to include all technical information to enable subsequent support and maintenance. At the application level, there should be a unified and seamless interface to both the static and dynamic data.

4.4 Accelerator Subsystems

4.4.1 RF

The booster and storage ring RF system mainly comprises of RF source, RF pre amplifiers, RF power amplifiers and the RF cavities along with the RF transport mechanism. These along with the normal operational control also requires the control loops for cooling water fine temperature control, frequency control, amplitude control and phase control. Some times the RF of booster also requires ramping and hence needs special programmable waveform synthesizers and RF generators to be interfaced with the control system often on GPIB or on LAN. Normally all the components are installed in the near vicinity of the RF cavities and hence usually single equipment controller suffice the need.

4.4.2 Vacuum

Vacuum system mainly consists of sputter ion pumps, vacuum gauges and valves, fast closing shutters and residual gas analyzer. Digital I/O and serial lines are the electrical interfaces used for most of vacuum devices. For high voltage power supplies powering more than one pump, special interfaces are used to read individual pump currents, based on voltage to frequency converters so as to insulate Equipment controllers from high voltage devices. This is the most wide spread (geographically) system and mainly comprises of more than one equipment controllers.

4.4.3 Injector (Linac / Microtron)

The injectors are small accelerators which accelerates the electrons to energies up to few tens of MeV for example at RRCAT we are having Microtron as injector to booster which provides electrons at 20MeV energy. Mostly machines use either Linac or Microtron for this purpose. The control requirements of these are similar to the other accelerator control requirements but since theses are the small machines the number of parameters is low and all devices are placed near the accelerator and hence usually one or two Equipment controller is sufficient for this.

4.4.4 Diagnostics

Accelerator diagnostics is the crucial system for proper functioning of the facility and mostly requires the state of the art technology to address its stringent requirements. These include the fast digitizers and frame grabbers to grab the images of fluorescent screens and synchrotron radiation profile monitors installed through out the ring and transport lines. Current measurement devices like DCCT (DC Current Transformers) and FCT (Fast Current Transformers). For monitoring and controlling top-up injection, high-precision DC current measurements are carried out by digital multimeters interfaced over GPIB. Beam Position Monitors (BPM) provides the information about the beam position on-line and requires the fast and precise electronics. Since this data is used for implementing the fast and slow orbit control system the equipment controllers for this subsystems usually are provided with boards with feedback loop implemented using the FPGA and some times with the dedicated communication links to other equipment controllers of the subsystems for fast data transfer between them. This layer also many times uses the principle of over sampling to improve accuracy of analog parameters by averaging.

4.4.5 Magnet Power Supplies

This subsystem comprises of many different power supplied both AC type and DC type. Because of the high inductance of the dipoles the tracking of the reference signal by the

PS can only be guaranteed to within a given maximum accuracy. On the other hand, quadrupole current have to precisely track the bending magnet current because tracking errors translate into undesirable betatron tune changes. Further more, eddy currents proportional to the bending field time derivative generate sextupolar fields that may have to be compensated by powering the sextupoles with waveforms tracking the said derivative.

To meet all the above requirements, the control system should allow the generation of independent, programmable, synchronized waveforms of arbitrary shape(sinusoidal, linear ramp, compensated ramp, etc) for each power supply. Even in references of DC power supplies (i.e. the power supplies meant for supplying current to DC magnets) provision is to be provided for arbitrary wave forms as they have to undergo the cycling process in order to attain the desired level of field uniformities and accuracy. To achieve this digital sample buffers and D/A converters, whose buffer content can be changed in between cycles to allow cycle-to-cycle tuning are used. The required precision determined by the maximum allowed tracking error is generally met by 16 bits at a conversion frequency of 10 KHz for both reference and read-back. New facilities are also investigating the implementation of feed-forward or feedback systems based on magnetic field or tune measurements.

4.4.6 Interlock System

This system must fault-protect equipment and people and prevent damage by mishandling to either. Some facilities divide this into two systems as Machine safety system and personal safety system, where as some facilities have a common machine interlock system. This system requires high reliability and in most of the machines is built using high reliability technology. This includes the use of an uninterruptible power supply, programmable logic controllers, battery backup, and redundant power supplies for VME crates used for these subsystems. Subsystems to be interlocked are:

- Power supplies and magnets
- Injection and extraction HV power supplies
- Vacuum
- Injectors (Linac / Microtron)
- Insertion Devices (i.e. inhibit injection if certain devices are closed).
- Beam lines and experimentation stations.

This system generally collects signals from other subsystems and often sends commands to other subsystems. It is usually built using industrial PLC and also many times uses the industrial SCADA for their control. Often the signals used are digital signals and the response time of 20 to 200 ms serves the purpose. Since the signals are to be collected from almost all parts of the machine, the signals are distributed and normally use more than one equipment controllers.

4.4.7 Timing System

The Timing System in accelerators is responsible for synchronising beam and RF and all control and data acquisition tasks. It generates the master trigger patterns that governs all the operation mode events in the machine. It generates all the synchronization pulses needed to control the beam injection into the ring for initial fill and top-up. The timing system many times also communicate data that are required for operation and correlation, as well as data communicated to the subsystems that change with the mode of the machine. Like time stamp/pulse ID, machine mode, and global machine status. It mainly comprised of electronic delay generator cards where the delays can be generated in synchronism to the master pulse. these synchronisation pulses are required for control of the beam transfer from the electron source to the storage ring and are also provided to diagnostic equipment and beamline equipment for synchronisation.

Fast Timing

Some of the accelerator synchronisation tasks such as those related with the injection and extraction process that requires the triggering of particle source and firing the transfer line components, such as injection and extraction pulsed magnets, beam diagnostic components (such as beam position monitors and current transformers) at the correct times are termed as fast timing tasks. These tasks require synchronisation with fine time resolution, to RF frequency, clock precision, and low jitter. Table 4.1 shows the main specifications of the hardware available from KEK and Stanford Research for fast timing system.

Table 4.1: Fast timing hardware specifications.

	KEK TD4V	Stanford Research DG535
Form	VME 6U	Bench/Rack mounting
Delay	16 Bit/RF clock	0 to 1000s -5ps steps
EPICS Support	Yes	Yes , via GPIB
Channels	1	4
Jitter	4.5ps at 508MHz	<60 ps

Event System Signals

There are some other accelerators components synchronisation tasks that are at course level where the requirement on timing resolution is more relaxed. This includes triggering the magnets for an acceleration ramp, triggering operational sequences such as the filling of the storage ring, BPM acquisition, feedback timing, insertion device control, and supplying the distributed control system with time synchronization pulse to control and correlation of data. These tasks are termed Events. Event signals are produced with a precision set by the storage ring revolution period and with predictable jitter. Table 4.2 gives the specification of one of the most advanced event based timing system developed at APS and enhanced by SLS and, more recently, DIAMOND.

4.4.8 Beamline Front-end and Experimental System

This sub system is the attachment between machine and the experimentation stations. On one hand the beam line users need to control insertion devices which are in the machine controls network, need information about beam position, as well as numerous other signals from the accelerator control system where as on the other hand beamline information, along with intensity and beam position from the beamlines, is needed in the accelerator control system to provide continuous control of the beam. Thus a bidirectional data flow is needed. Often it is seen that the facility is built with few beamlines (about 4 to 7) at first and then at later stages the number increases to fully occupy the beamline slots for which machine is designed. Also this addition of new beam lines is done when the machine is in normal operation with the beamline front-ends (specific zone designed for coupling of beam lines with the accelerator facility). Many times, the beamline developing agencies

Table 4.2: specification of timing event system at DIAMOND.

Parameters	Value
Events	8-bit code -255 events
Resolution	8ns
Event Tx Trigger	Hardware inputs, software, Event Ram Clock
Event Rx output	Hardware outputs, software (EPICS record process)
Transmission medium	Gigabit Ethernet

are also different as organisations some time sell the beam lines to other units. And most important, the beam lines are designed to perform different types of experiments and thus all beam lines have different operational requirements.

Thus the machine control system and its interfaces are needed to be built in such a way that the continuous development and commissioning of beamlines do not affect the normal operation of the machine. Normally the beam line control systems are separated from the machine control system with the help of gateway PC and dedicated networks. Also since this layer is mostly built by different agencies, the software for these show the blend of different technologies.

4.5 Future trends in SRS control system

Control systems for future large machine will continue to use the increase in CPU speed, memory size and network bandwidths offered by future electronic market. Future sys-

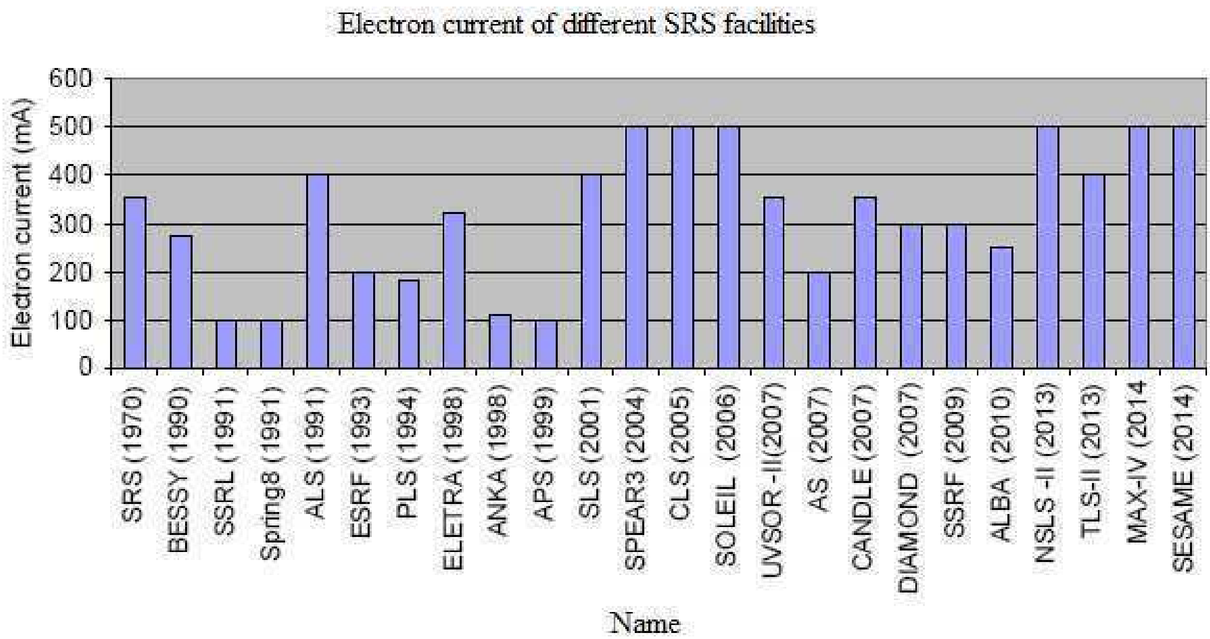


Figure 4.7: Electron current of synchrotron radiation sources.

tems will employ a substantially larger number of devices, and with the movement towards front end processors becoming embedded in each device, the number of processors communicating on the network stands to grow by two orders of magnitude [48]. Though the upcoming machines show less increase towards energy front (see figure 4.6 for the new facilities i.e. facilities coming after 2000 are having energies between 2.5GeV to 3GeV) but some machines in future may also operate at high energies for example the upcoming fourth generation ultimate high-energy X-ray source PETRA-III to be operated at particle energy of 6GeV)[24]. As the circumference or size of machine depends on the electron energy the most of the future machines are expected to be of intermediate size and few like PETRA-III will be physically quite large, thus causing the control system to span tens of kilometers. From figure 4.7 one can see that the new machines may limit to the electron energy of 2.5 to 3GeV but they will certainly have to be operated at higher beam currents 300mA to 500mA, thus presenting control engineers with the challenge of beam instabilities arising out of nonlinear effects observed at high currents. Also, the use of undulators for getting the same photon energy as obtained by large high energy storage rings from a lower energy electron beam but with higher harmonic of the undulator emission and/or by using a shorter undulator period poses the tight requirement on beam emittance [24] and beam orbit stability [33], thus requiring a faster orbit feedback control mechanisms. Additionally, timing requirements will be tighter and the amount of data to be archived will increase. As the future systems will use larger number of devices higher reliability requirements will be imposed on the control system in order to ensure the uninterrupted operation of the machine. Thus the future control systems may consider the use of redundant components with automatic changeover as a way to increase reliability. But due to budget limitation every component can not be made redundant; therefore, engineers will need to evaluate the characteristics of each element in order to use redundancy to maximum advantage. Another method to improve system availability is to reduce the repair time and since, the repair time consists of the time taken to diagnose the problem and the time to replace the faulty component, the future trend is towards minimising the diagnostic time. Thus by incorporating intelligence at machine component level to provide integrated diagnostics features that can detect problems at early stages and support the control system can be of great advantage. So that the control system can constantly verify, in a uniform fashion, that all devices are functioning correctly, or, in the event of an error, enabling quick identification for repair.

To address the issue of maintainability, future control system software will trend towards increased use of modular architecture as this facilitates changing components in one software layer without impacting the rest of the control system. With the increase in the size of generated data in future machine, the interface designers will have to consider carefully what data is needed by each type of user. As it is not enough to present such large volumes of data and expect humans to make efficient interpretations and responses, the control system would provide a high degree of automation, with interfaces becoming more for information than for control. To provide the needed level of automated setup, control will rely on having accurate machine model available to all applications and having software adapt to empirical data. All this requires the addition of intelligence at upper level of control system to automatically adjust according to the machine and user requirements.

4.6 Summary and conclusion

In early days the SRS facilities grew around large accelerator laboratories mainly built for particle physics experiments and allowed the synchrotron radiation users to use the SR produced in booster or storage rings in parasitic mode. Later on, inspired by the increasing demand of SR users, the dedicated SRS facilities started appearing. The control system, for these SRS facilities have mainly evolved out of accelerator control systems built for large accelerators facilities with modifications to adopt the extra signals coming from the SR beam lines and the insertion devices integrated in the bending magnets and the straight sections of storage rings. From the beginning, the SRS control systems have faced the challenge of surviving and fulfilling the SRS control demands in the continuously upgrading, expanding and evolving SRS environment. Older control systems have undergone upgrades, incorporating advances in technology to meet ever increasing requirements for speed, accuracy and automation. Architecturally, control systems have evolved from two-layer architectures popular in early days to the three layer structures followed by again to two layer structures favoured today by successfully adopted technology advances to add functionality and enhance performance.

The line based CRTs used as control system user interfaces of old facilities have been uniformly replaced with animated graphical displays featuring multiple windows and often multiple physical screens per console. Today's control rooms are populated with dozens of monitors, some even prominently featuring wall sized displays. Numerous graphics software packages are used to create a wide variety of graphical user interfaces including synoptic displays and graphs with advanced alarms systems providing the facility to filter out of context alarms and collapse related alarms into trees. Early SRS control systems were almost exclusively custom creations. With the exception of commercial computers, virtually every piece, both hardware and software, of early systems was created by laboratory scientists and engineers. Today the controls community is increasingly inclined towards the use of commercial and shared components in an effort to reduce development costs and improve reliability. EPICS and TANGO are the two emerging free tool kit based control system development frame works adopted by many facilities to reduce development costs and time line, and increase reliability with high degree of software reuse. Reusing software components eventually results in code that has been tested far more extensively than is possible with individual custom developments.

Although the requirement of operation of future SRS machines with higher beam currents and very low beam emittance will throw new challenges to control engineers for beam stabilities, the current pace of development of commercial technology will likely meet the basic requirements for the building blocks of future control systems. It will be more challenging to provide a reliable, maintainable, secure and operable control system due to the international nature of future machines and the likelihood of distributed development, standards need to be developed first and enforced throughout the project. The requirements for the control system need to consider all stages of the project, including maintenance and the inevitable upgrades, rather than just making the control system work for the initial machine configuration and commissioning. Incorporating commercial solutions and the best features of existing systems along with the use of vigilant engineering practices, future SRS control systems will successfully operate the facilities.

Chapter 5

Accelerator subsystems as intelligent agents

5.1 TL-1 intelligent agent

Transport lines are accelerator components primarily utilised for transport of charge particle beams across two physical location. This particle transport is unidirectional and during the course of transport, transport line's control the charged particle beam to maximise the injection efficiency. This injection efficiency is affected by various factors like imperfect beam production, beam misalignment by source, residual magnetic fields, flawed control elements, displaced/misaligned magnets, imperfect detectors, error and drifts in magnet power supplies and injection & extraction timing errors.

The tuning of transport lines is a routine and time consuming task performed during the start-up and conditioning phase after a maintenance period or an unexpected fault. Also the sudden as well as gradual variation of beam parameters occurring at source imposes the intermittent transport line tuning requirement on operators during normal accelerator operation.

For maintaining the injection efficiency of transport lines under the dynamic system conditions the concept of feed-forward correction and feed-back correction are the possible solutions. For automatising the tuning process in transport lines the model assisted rational, knowledge based sequential approach is a feasible solution. In the presence of uncertainties in system and insufficient measurable system parameter the operator assisted machine learning approach could be the best mechanism for choosing the tentative values of different machine operation parameters.

Accelerator environment being a complex system imposes limitation in the measurement of all the affecting parameter thus the conventional feed-forward approach could be used here only upto a limited extent. Further due to non availability of direct measurement for the system parameters that are needed to be controlled the conventional feedback control methods could not be used here directly. Artificial intelligence based transport line and accelerator tuning methods has been proposed in literature that are discussed in detail in chapter 2. The development of agent based control for transport lines will be able to effectively utilise the feed-forward based correction concept using supervised learning, predictive corrections and through cooperation mechanism between different agents in the overall multi-agent based control of accelerator facility. With the goal based control approach the agents can utilise the model assessed methods for exercising the feed-back correction approach through derived system parameters. It can successfully integrate the AI based accelerator tuning methods to the overall control infrastructure thereby fulfilling the gap left between the AI based fault diagnostic and reactive control through systematic prioritised multi-objective optimisation. Further to this it also provides the possibility of inclusion of other concepts like model based tracking, pre-structured online system identification and system dynamics learning thus providing an open platform for analysing the performance of new concepts with accelerator control.

5.1.1 TL-1 intelligent agent formulation

In accelerator control system for deploying any new scheme of accelerator control for existing accelerator facilities, needs that there should be minimum switch-over time and if possible the work should be carried out in parallel i.e. without stopping the machine operation. These requirements strongly advocates the choice of modular architecture for intelligent agent based machine control, as in almost all of the accelerator facilities the control system itself is modular with multi-layered architecture. The uppermost layer usually runs on PC and comprises of industrial or open source SCADA package like PVSS, EPICS and TANGO. All of these provide well defined interfaces for data interchange between different applications. Hence this uppermost layer can directly be used in the agent architecture in *perception* and *execution* blocks thus reducing the switch-over time to a minimum with almost no changes in the existing operator GUIs. The new intelligent object based control facility can be added to the system with new GUI panels on separate PC so that operator can quickly switch between two modes normal machine mode and auto machine mode in case erroneous behaviour is observed.

As a case study in this thesis work the agent for controlling of TL-1 transport line of INDUS-1 complex whose model is discussed in detail in chapter 3 is used. For developing the TL-1 intelligent agent we have adopted the bottom up approach to find the system capabilities of existing system which can be used without much change to the system. For implementing the agent architecture we have developed our own framework as this provides us a greater flexibility in terms of system integration and development as opposed to the use of any other already available framework. Further to this we are going to follow an approach of first building the agent architecture with simple and limited capabilities and then enhancing the capabilities one by one after refining the overall system through simulations. Though the complete deployment of the developed agent based system seems

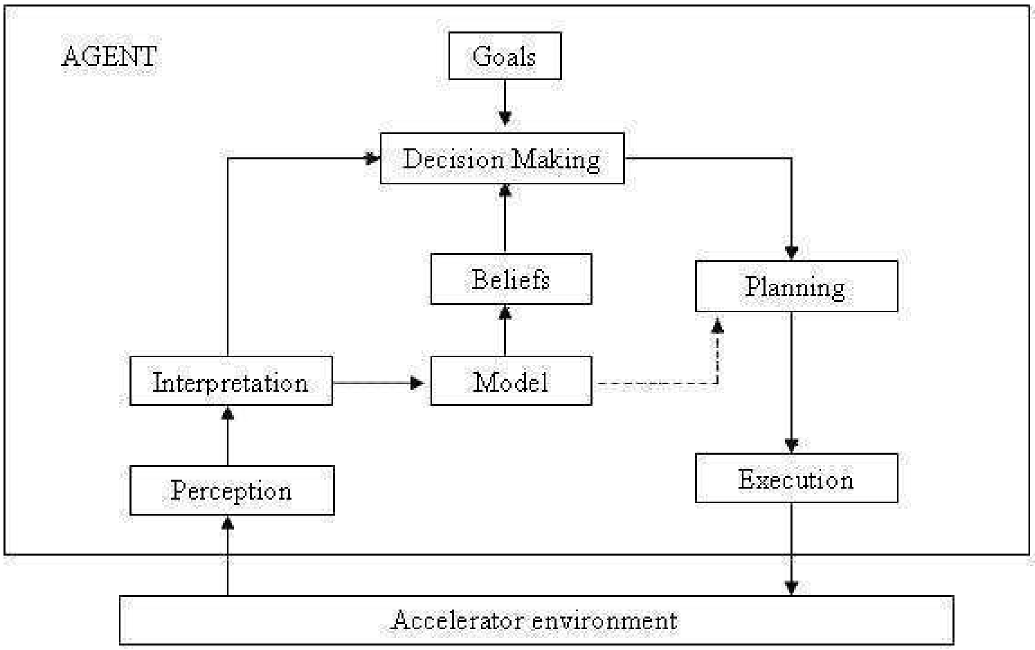


Figure 5.1: Model-based goal-based modular agent architecture adopted for TL-1 agent.

long time objective and may not be possible in limited time. The efforts will be tried at the end, to test the developed method on the actual machine. As INDUS-1 control system is having a modular architecture, the TL-1 agent is also developed with modular architecture to use maximum of the available resources.

TL-1 intelligent agent architecture

TL-1 intelligent agent is formulated with a modular architecture as shown in Figure 5.1. Here the *perception* and *execution* blocks directly interact with the accelerator environment. In the TL-1 case it interact with the TL-1 Power supplies and Beam diagnostic devices (FCT through CRO, fluorescent BPM screens). In TL-1 some of the power supplies (CV1, CH1, QF1 and QF2) are controlled through Microtron control system and rest all power supplies are controlled through INDUS-1 P/S control system. This requires the development of interface for remote reading and writing of power supplies values for both of the control system(TL-1 control system and Microtron control system).

The interface for these have been developed and integrated with the above two modules. The function of the *perception* block is to read different P/S settings and read-back values, CRO trace for FCT and the BPM images. Depending upon the read data it then generates the appropriate event. Events are passed directly to the respective blocks in the form of messages along with the required data. The *interpretation* block serves the purpose of processing the raw data acquired by the *perception* block to convert it to the required form (in TL-1 this block extracts the beam position (x , y) and beam sizes (σ_x, σ_y) from the BPM images and the injection current value from CRO trace of FCT) . *beliefs* block is basically the agentś data storage. This stores the system state and other meta data required in the processing/decision making steps. *goal* block contains the definition for all the goals and provision for enabling/disabling of goals. Definition of goal comprises of plan list. Plans in the list are the alternate plans by which the goal could be achieved in different system conditions. The position of the plan in the plan list decides its priority. The plan at higher level in the list has higher priority; Table 5.1 lists all goals and the associated plans for TL-1 agent. The *decision making* block depending upon the current events and the agent beliefs decide the plans to be executed to achieve all the active goals. It does this by evaluating the plan applicability function and selecting the highest priority applicable plan from the list for each active goal. The *planning* block serve the purpose of executing the selected plan in synchronised/coordinated way and updating the active goal list. Each plan body comprises of sequence of action steps to be followed to attain the desired goal. Table 5.3 lists the action sequences for each plan listed in Table 5.1. The *execution* block sends the commands obtained from different blocks in the form of messages to machine components after checking them for the validity (setting within limits, system in appropriate mode). The *model* block in itself is an intelligent agent comprises of the TL-1 model and serves the purpose of providing the information

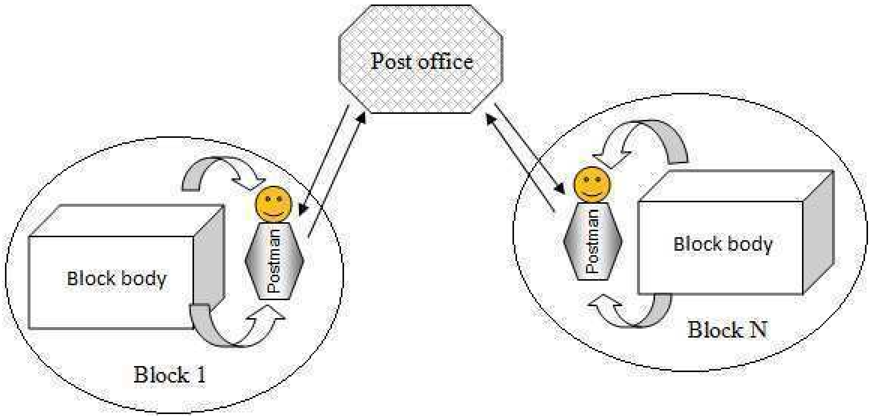


Figure 5.2: Architecture adopted for TL-1 agent blocks.

about the probable outcome of the stated actions on the machine. As an agent it provides functionalities listed in Table 5.2 to different TL-1 agent blocks necessary for achieving the required goals.

Architecture of agent blocks

Each agent block needs to do its intended function on receipt of action request from either block of same agent or from another agent and seeing the accelerator control system architecture it becomes clear that these blocks can be physically situated on different machines thus the postman-postoffice based communication mechanism as shown in figure 5.2 is adopted for building different blocks of agent. In this type of architecture there is a postoffice (whose address is known to all the agents) and an individual postman is assigned to each block. The postman periodically sends the messages to be sent to other blocks to postoffice and collects the messages addressed to the corresponding block from postoffice. This type of communication scheme is open for extension of the system as well as provides an easy debugging interface by presenting the inter-block messages. Each message is comprised of the following fields.

1. Time : Time field represents the time when the message was originated by the

source.

2. ID : Identity field represents the unique message identity number.

3. Source: The source field represents the name of message originating source.

4. Destination: The destination field represents the name of the recipient (block/agent) for which the message is sent.

5. Type/Sync : This field provides the type of message information and if required the synchronisation related information related to the message.

6. Data: this field contains the data specific to the message type.

The postman and postoffice are connected to each other through TCP/IP protocol using socket programming. The developed code for this framework is shown in Figure 5.3.

5.1.2 Simulation program design

Computer system

The computer is a Lenovo PC (Intel(R) Core(TM)2 Quad CPU @2.66GHz , Memory (RAM) 4.00 GB) running windows Vista^{TM} Business edition (32-bit) operating system. Application software included Microsoft packages for graphics, spreadsheet and text. For program development, LabVIEW [65] development environment and MAD [38, 70] package available from website (`http://wwwslap.cern.ch/mad/`) is installed. The LabVIEW is chosen for fast development cycle, MAD package being the standard package for simulating the accelerator components is used for simulating the accelerator standard components.

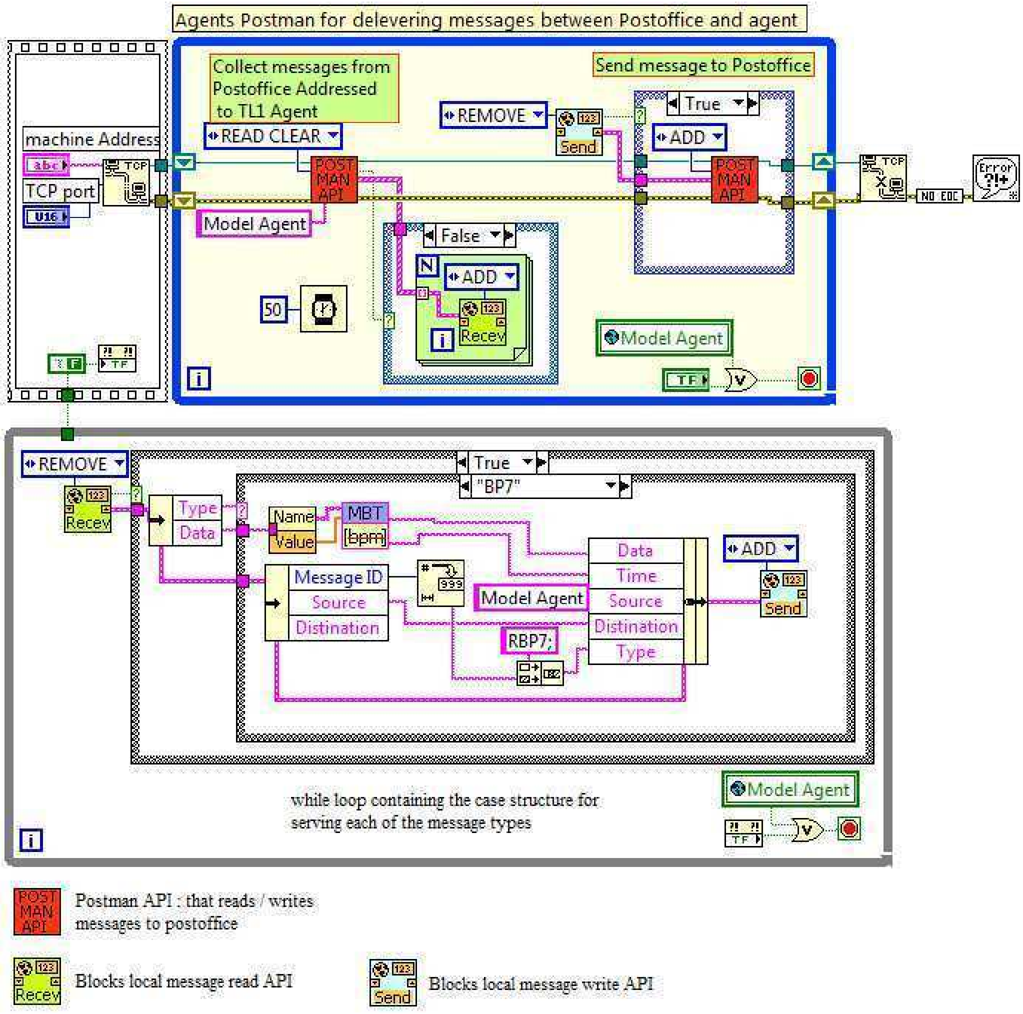


Figure 5.3: Developed code for agent blocks framework.

Program organisation

The simulation program comprises of three main parts. These are the multi-agent system (MAS), the model of the accelerator environment with which the agents interact and the user interface. The principal data flow involved is shown in figure 5.4. The accelerator model is made in the form of a software module, comprised of four different sub-modules: Microtron, TL-1, Booster and Storage Ring. These modules are designed to provide the interface similar to the interfaces available in the INDUS-1 accelerator machine for these subsystems. In an Integrated form the accelerator model provides the facility to simulate the machine behaviour under different machine parameter settings. The multi-agent

Table 5.1: Goals and associated plans of "TL-1 agent".

<table>
<tr><th>Goal</th><th colspan="3">Plan</th></tr>
<tr><td rowspan="6">Tune beam position and angle</td><td>No.</td><td>Name</td><td>Plan applicability</td></tr>
<tr><td>1</td><td>Measure Beam Position at all BPMs</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && optimise allowed</td></tr>
<tr><td>2</td><td>Correct Position and angle using two correctors without BPM data</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && (the suggested corrector values of two correctors are within constrain limits) && (without BPM correction is allowed) && (correction applicability $>$ set-limit) && (system not in starting state)</td></tr>
<tr><td>3</td><td>Correct Position and angle using four correctors without BPM data</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && (the suggested corrector values of four correctors are within constrain limits) && (without BPM correction is allowed) && (correction applicability $>$ set-limit) && (system not in starting state)</td></tr>
<tr><td>4</td><td>Correct Position and angle using two correctors with BPM data</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && (the suggested corrector values of two correctors are within constrain limits) && (system not in starting state)</td></tr>
<tr><td>5</td><td>Correct Position and angle using four correctors with BPM data</td><td>Event⟨new-inj available⟩ && ($inj < inj_{LO}^{lim}$) && (the suggested corrector values of four correctors are within constrain limits) && (system not in starting state)</td></tr>
<tr><td rowspan="3">Cooperatively optimise injection current</td><td>1</td><td>Suggest nearest desired beam position and angle to Microtron agent for constrained TL-1 settings</td><td>(The corrector values of any of the four correctors exceeds constrain limits)&& (system not in starting state) || Event⟨suggest new-OP⟩</td></tr>
<tr><td>2</td><td>Evaluate injection current improvement for suggested beam position, angle and beam current</td><td>Event⟨Evaluate OP⟩</td></tr>
<tr><td>3</td><td>Cooperatively correct position and angle using four correctors</td><td>($inj < inj_{LO}^{lim}$) && (The corrector values of any of the four correctors exceeds constrain limits) && (system not in starting state) || Event⟨ invoke CO⟩</td></tr>
<tr><td rowspan="3">Auto Start/Stop TL-1</td><td>1</td><td>Start TL-1 normal</td><td>Event⟨Start TL-1 normal⟩</td></tr>
<tr><td>2</td><td>Start TL-1 optimised</td><td>Event⟨Start TL-1 normal fail⟩</td></tr>
<tr><td>3</td><td>Stop TL-1</td><td>Event⟨Stop TL-1⟩</td></tr>
</table>

system comprises a set of entities representing agents. To simulate real-world behaviour the models must run concurrently and this is achieved by using multi-threaded feature of graphical programs. In the single processor environment of the PC, threads are managed

Table 5.2: Agent "LiveTL1Model"capabilities.

	Capability	Details			
		No.		Data needed	Algorithm used
1	Provide x,x',y,y'	1	Provide beam position for given beam position and currently applied power supply settings	x,x',y,y' at the TL1 input	TL1 model
		2	Provide beam position for given beam position and given power supply settings	x,x',y,y' at the TL1 input and power supply setting values	TL1 model
2	Provide power supply settings value for matching the beam position and angle	1	Provide value of CH1, CH2, CV1,CV2 for matching the beam position at both ends	Beam position at ends	Genetic algorithm and TL1 model
		2	Provide value of CH1, CH2, CH3, CH4, CV1,CV2, CV3, CV4 for matching the beam position at both ends	Beam position at ends	Genetic algorithm and TL1 model
3	Provide QP settings for matching of beam size	1	Provide value of QF1, QD1 for matching beam size	Beam size at input	Genetic algorithm and TL1 model
		2	Provide value of QF2, QD2 for matching beam size	Required beam size at output	Genetic algorithm and TL1 model
4	Provide the nearest optimum beam position at TL1 input	1	Provide the nearest optimum beam position at TL1 input with constrained power supply setting values	Present beam position at input and constrained power supply values	Genetic algorithm and TL1 model
5	Predict the gain in injection for the given beam current, beam position and TL1 settings	1	Predict the gain in injection with the given beam current, beam position and TL1 settings	Beam current , beam position and TL1 settings	TL1 model, Booster model

Table 5.3: Plans and associated action sequence.

Plan ID No.	Plan name	Action sequence
1	Measure beam position at all BPMs	1. Put system in measurement mode 2. Select BPM1 and capture beam image 3. Select BPM2 and capture beam image 4. Select BPM3 and capture beam image 5. Restore system in normal state 6. Interpret beam position at all BPMs from the beam images 7. Update the agent beliefs about the beam position at BPMs 8. Generate event[new BPM data available]
2	Correct position and angle using two correctors without BPM data	1. From TL1 model obtain the set of all beam position and angle values corresponding to the present injection current 2. Select the most probable beam position and angle value depending upon history data and priority 3. From TL1 model obtain the new corrector values corresponding to most probable beam position 4. Apply new values to system through interface module 5. Update beliefs depending on the action results on injection value
3	Correct position and angle using four correctors without BPM data	1. From TL1 model obtain the set of all beam position and angle values corresponding to the present injection current 2. Select the most probable beam position and angle value depending upon history data and priority 3. From TL1 model obtain the new corrector values corresponding to most probable beam position 4. Apply new values to system through interface module 5. Update beliefs depending on the action results on injection value
4	Correct position and angle using two correctors with BPM data	1. from TL1 model obtain the beam position and angle at the input from the present BPMs data 2. From TL1 model obtain the new corrector values for present beam position and angle values 3. Apply new values to system through interface module
5	Correct position and angle using four correctors with BPM data	1. from TL1 model obtain the beam position and angle at the input from the present BPMs data 2. From TL1 model obtain the new corrector values for present beam position and angle values 3. Apply new values to system through interface module
6	Suggest nearest desired beam position and angle to Microtron agent for constrained TL-1 settings	1. From TL1 model obtain the beam position and angle at the input from the present BPMs data 2. From TL1 model obtain nearest desired beam position and angle values at Microtron output/TL1 input from constrained TL1 settings 3. Suggest new values to Microtron agent
7	Evaluate injection current improvement for suggested beam position, angle and beam current	1. From TL1 model obtain the injection current improvement for suggested beam position angle and beam current 2. Update the belief

<table>
<tr><td>Plan ID No.</td><td>Plan name</td><td colspan="2">Action sequence</td></tr>
<tr><td>8</td><td>Cooperatively correct position and angle using four correctors</td><td colspan="2">1. From TL1 model obtain the beam position and angle at the input from the present BPMs data
2. From TL1 model obtain the new corrector values (constrained) for present beam position and angle
3. Apply new values to system through interface module
4. Update beliefs</td></tr>
<tr><td rowspan="7">9</td><td rowspan="7">Start TL1 normal</td><td colspan="2">1. Switch ON all the power supplies of TL1
2. Apply the default values to all the power supplies
3. Make the deviation report for parameters
4. Below options</td></tr>
<tr><td>No deviation observed</td><td>Deviation observed</td></tr>
<tr><td>Proceed</td><td>1. if small deviation, message to operator and proceed
2. if large deviation, message to operator and stop</td></tr>
<tr><td colspan="2">5. Prepare system for cycling
6. Prepare cycling report</td></tr>
<tr><td>Cycling pass</td><td>Cycling fail</td></tr>
<tr><td>Proceed</td><td>1. redo cycling
2. if three retries fail,stop, otherwise proceed</td></tr>
<tr><td colspan="2">7. Declare system in started state</td></tr>
<tr><td rowspan="7">10</td><td rowspan="7">Start TL1 optimised</td><td colspan="2">1. Switch ON all the power supplies of TL1
2. Apply the default values to all the power supplies
3. Make the deviation report for parameters
4. Below options</td></tr>
<tr><td>No deviation observed</td><td>Deviation observed</td></tr>
<tr><td>Proceed</td><td>1. if small deviation, message to operator and proceed
2. if large deviation, message to operator and stop</td></tr>
<tr><td colspan="2">5. Prepare system for cycling
6. Prepare cycling report</td></tr>
<tr><td>Cycling pass</td><td>Cycling fail</td></tr>
<tr><td>Proceed</td><td>1. redo cycling
2. if three retries fail,stop, otherwise proceed</td></tr>
<tr><td colspan="2">7. Measure beam position at all BPMs
8. From TL1 model obtain the beam position and angle at the input from the present BPMs data
9. From TL1 model obtain the new corrector values for present beam position and angle values
10. Apply new values to system through interface module
11. Declare system in started state</td></tr>
<tr><td>11</td><td>Stop TL1</td><td colspan="2">1. With fixed rate set all power supply values to minimum
2. Switch off all the power supplies</td></tr>
</table>

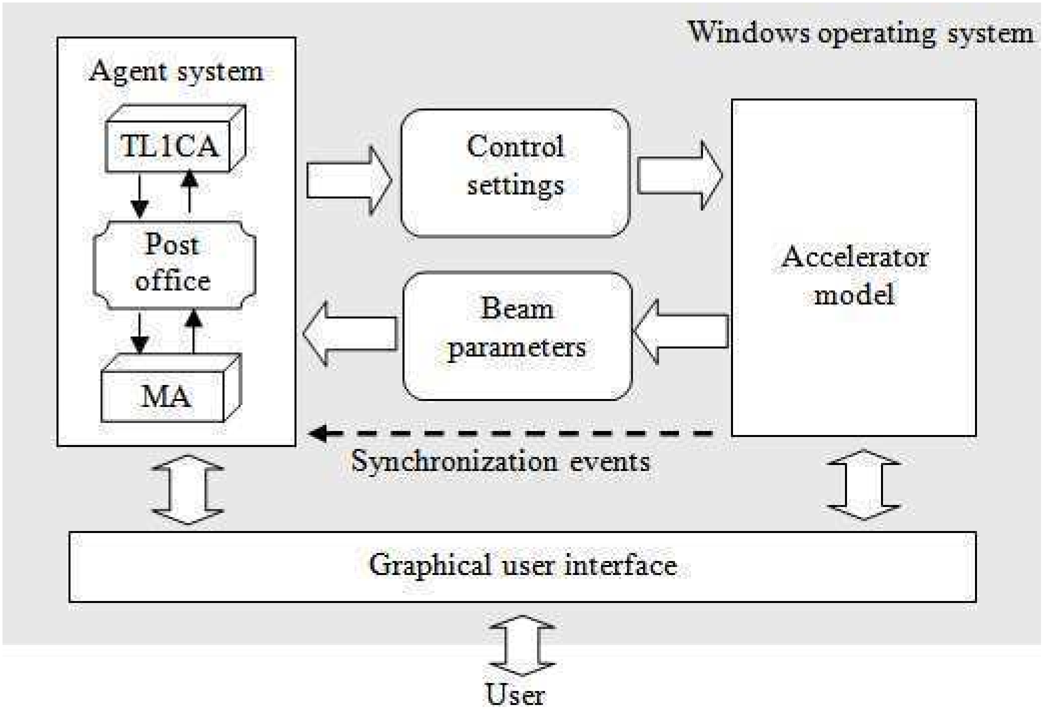


Figure 5.4: Simulation program organisation.

by the windows multi-tasking operating system. For synchronising the execution process between MAS system program and the accelerator simulation module the events in the form of messages are used through the postoffice scheme. The arrangement is shown diagrammatically in figure 5.5

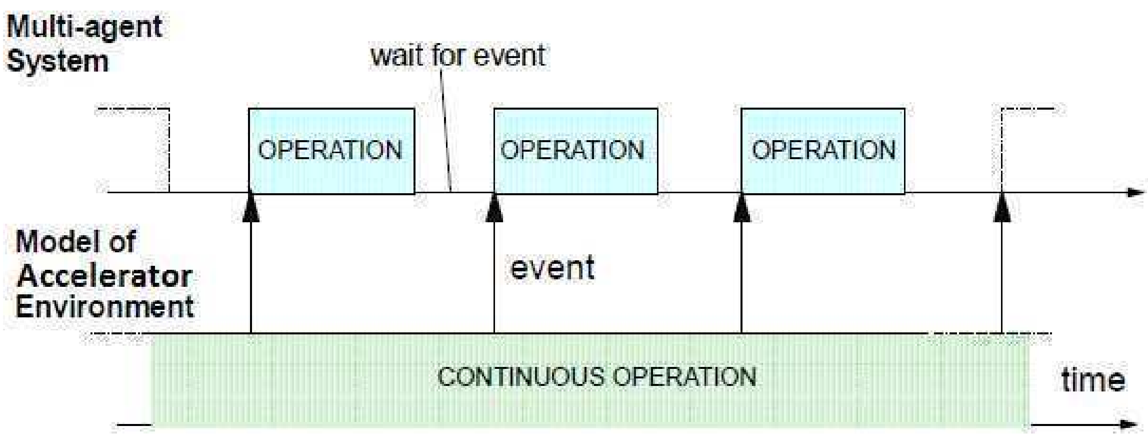


Figure 5.5: Thread synchronization.

Execution synchronization is incorporated into the simulation program design however it is tried that this interaction between the accelerator model and the agents remain

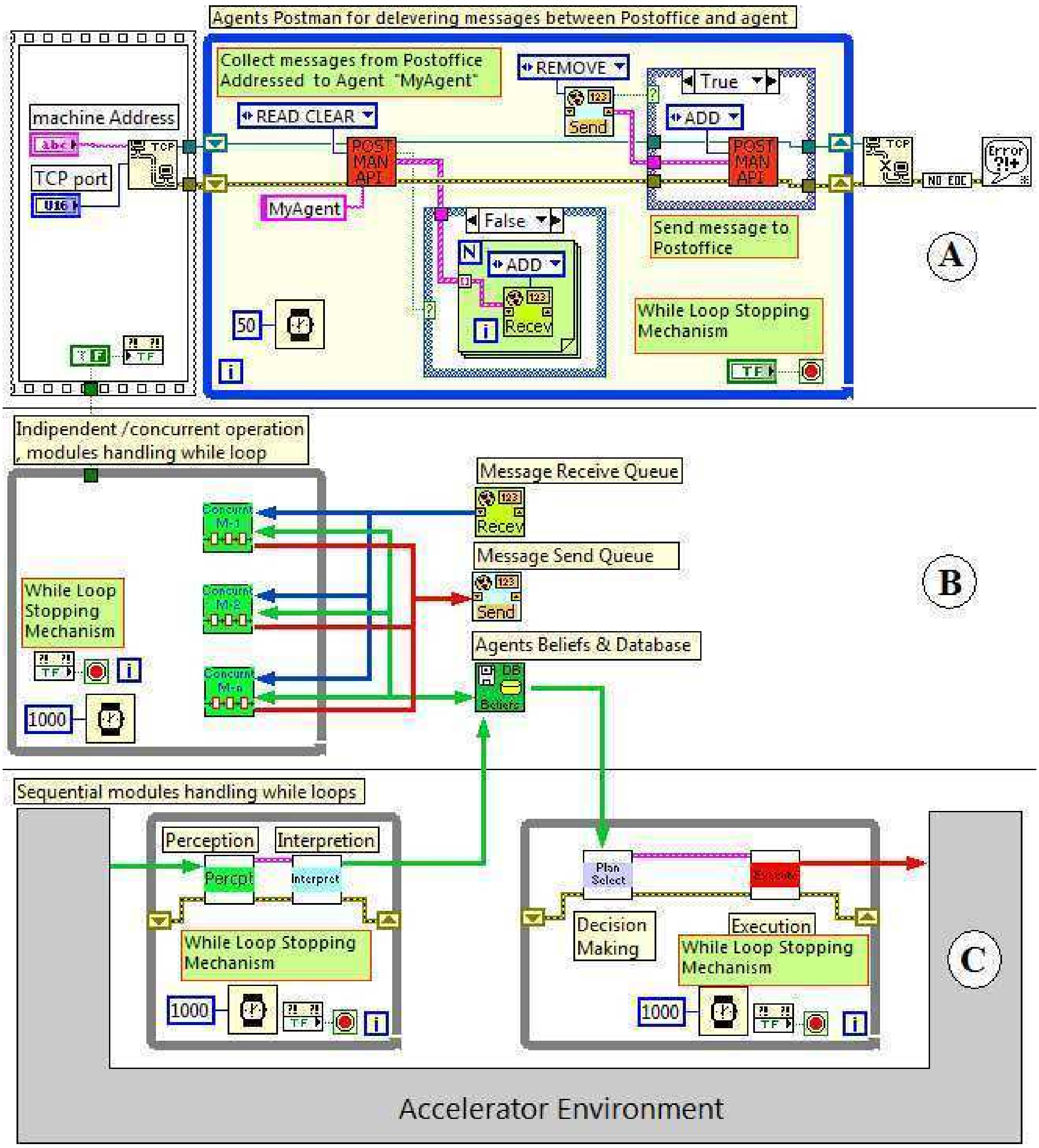


Figure 5.6: Graphical programming language framework for agent representation, layer (A) implements the agents communication with the postoffice, layer (B) implements the concurrent process modules, layer (C) implements the agents interaction with the accelerator environment.

Note: The Red and Blue arrows are only for illustrative purpose to clearly show the interconnection between different modules.

distinct from any behavioural interaction which is being modelled in the simulation.

For data exchange between programs the method of messaging, application program interface(API) and through global variables is adopted. The method of messaging is used to provide the modules ability to operate even in distributed environment where the module could be deployed on the respective machine, physically connected to the subsystem.

Agent representation

There is no facility in the graphical programming language which enables an agent to be represented directly, so a framework is developed in graphical programming environment for representing the agent with modular architecture. The Figure 5.6 shows the developed graphical code for this agent framework. This framework takes advantage of inherent parallel execution principle of graphical programming environment. By virtue of dataflow principle every unconnected piece of code executes in parallel and in unsynchronised manner(i.e. there is no guarantee that which piece of code will run first). The overall frame work is implemented through parallel while loop structures as shown in Figure 5.6 where all the loops are arranged in three layers, layer (A), layer (B) and layer (C). Layer (A) is the agents communication handling layer and implements the message delivery between postoffice and agent using postman API. In this loop the code periodically collects the messages from postoffice addressed to the agent and inserts these new messages to the *Message Receive Queue*. It then sends the messages from *Message Send Queue* to the post office. The layer (B) implements the concurrent operating modules. These modules normally implements the capabilities that serves the message based event handling. These modules uses the *Message Receive Queue*, *Message Send Queue* and *beliefs* & *Agents Real Time Database* for implementing their functionalities. The layer (C) implements the agents interaction with the accelerator environment and is comprised of two while loops . the first while loop implements the *Perception* module and *Interpretation* module that periodically reads the sensor reading from the accelerator environment, converts it to the required accelerator physics parameter format and updates the agents *Beliefs* accordingly. The second while loop implements the *plan selection* and *execution* process. All the loops in the three layers execute in parallel thus the agent execution cycle can be summarised as following.

1. Sensor data from accelerator environment are read and processed to update the beliefs.

2. Messages are read from *Message Receive Queue* and processed for message handling or event generation.

3. By the concurrent process modules, beliefs are read and Goals are updated: (i.e. new goals are generated and goal priorities are determined).

4. Plan is selected from plan library and is executed.

In the absence of external messages by other agents the agent program developed using this agent framework works autonomously to achieve its goal under changing environment conditions by selecting the appropriate plans from the plan library. If required it can send messages addressed to itself to influence their future course of actions thus providing the self-determining behaviour to the agents. With the ability to communicate with other agents through messages the framework extends the cooperative multi-agent system development ability. which is utilised to implement the complex behaviours for the system control.

5.1.3 Simulation phase overview

Objective

The objective of this simulation phase is to simulate the accelerator system that comprises of Microtron, TL-1, Booster and Storage ring and capture the operating characteristics of this system to reveal the potential for enhancing the machine performance through intelligent agent based control. The capturing of operational characteristics is achieved by varying parameters (beam position in first case) in 'open loop'operation of the accelerator model where as the performance enhancement in operation against the introduced variations is being investigated by introducing a simplified multi agent system comprises of two agents ***"LiveTL1Model"*** and ***"TL-1 Agent"***, capable of providing 'closed loop'control by means of adjusting the TL-1 power supplies to maximize the injection current in booster.

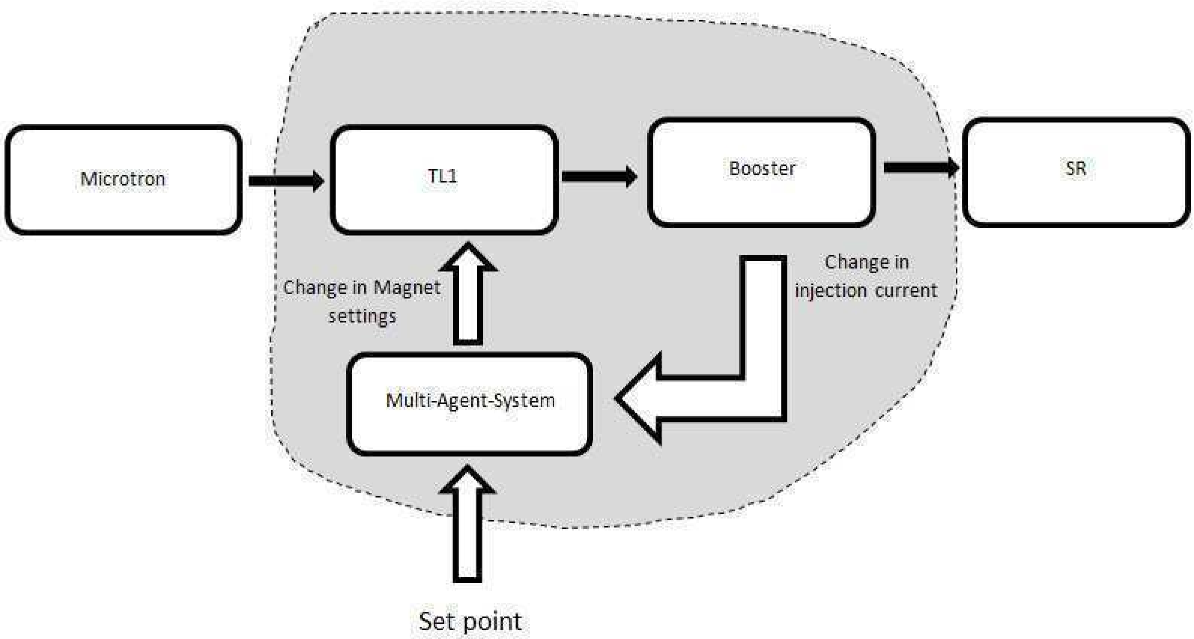


Figure 5.7: Total system model.

Total System Model

The diagram of the total system model (developed using Eq.(3.11) to (3.32)) is shown in figure 5.7, in which the main constituent parts are separated for convenience of representation. External to the model is the beam coming from Microtron and going to TL-1 input and the beam going out from booster to storage ring (SR). The Microtron is assumed to be capable of providing the beam with the design emittance ($e_x = 8.0 \times 10^{-7}, e_y = 3.0 \times 10^{-6}$) and size ($\alpha_x = -0.6708, \beta_x = 0.9645, \alpha_y = -0.5046, \beta_y = 0.9945$) at all the beam positions in the extraction chamber. Provision is made to vary the position and angle of beam coming out of Microtron to simulate the behaviour similar to actual beam position variations observed in Microtron output beam. The storage ring is assumed to be in fully tuned condition for storing the accelerated electron beam current with no control given to user for alteration in any of its parameters and only models the current decay behaviour based on the experimental data collected for decay of beam from $125mA$ to $0.2mA$.

Microtron model: For this simulation phase of optimization with two agent the Microtron model is represented with a program that generates the electron beam with the design parameters($e_x = 8.0 \times 10^{-7}, e_y = 3.0 \times 10^{-6}$) and size ($\alpha_x = -0.6708, \beta_x = 0.9645, \alpha_y = -0.5046, \beta_y = 0.9945$) comprises of given number of macro particles and capable of providing this beam with any desired beam position and angle attributes (x, y, x', y'). figure 5.8 shows one such simulated beam comprises of 2000 macro particles with $x = -6.0mm$, $y = 0.0mm$, $x'= 0.0mrad$ and $y'= 0.0mrad$. Table 5.4 list the specification of this Microtron model.

TL-1 model: The TL-1 model is developed as discussed in chapter 3. It accepts beam comprises of macro particles with different beam attributes and depending upon the setting values of all of the power supplies it tracks all the macro particles in the beam

Table 5.4: Microtron model specifications.

Name of parameter	Parameter value		Units
Number of macro particles	340-2000		Number
macro particle distribution	2D Gaussian		
Twiss parameters	Allowed values	Design value	
	$\alpha = \pm 1$	$\alpha_x = -0.6708$	
		$\alpha_y = -0.5046$	
	β =0 to 1	$\beta_x = 0.9645$	
		$\beta_y = 0.9945$	
	γ = calculated	$e_x = 8.0\times10^{-7}$	
		$e_y = 3.0\times10^{-6}$	
	$\eta = 0$	$\eta = 0$	
Beam position variations in horizontal plane(x)	-20mm to +20mm		mm
Beam position variations in vertical plane(y)	-20mm to +20mm		mm
Beam angle variations in horizontal plane(x')	-20mrad to +20mrad		mrad
Beam angle variations in vertical plane(y')	-20mrad to +20mrad		mrad
noise in position	0-0.5mm (uniform distribution)		mm
noise in angle	0-0.5mrad (uniform distribution)		mrad

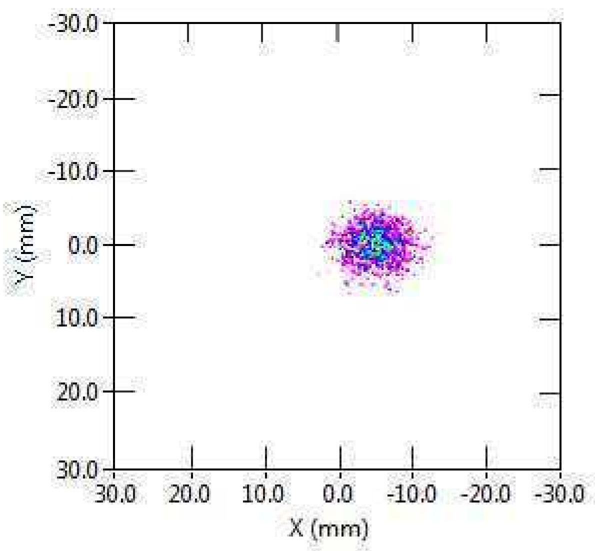


Figure 5.8: Simulated beam image at beam position monitor 1.

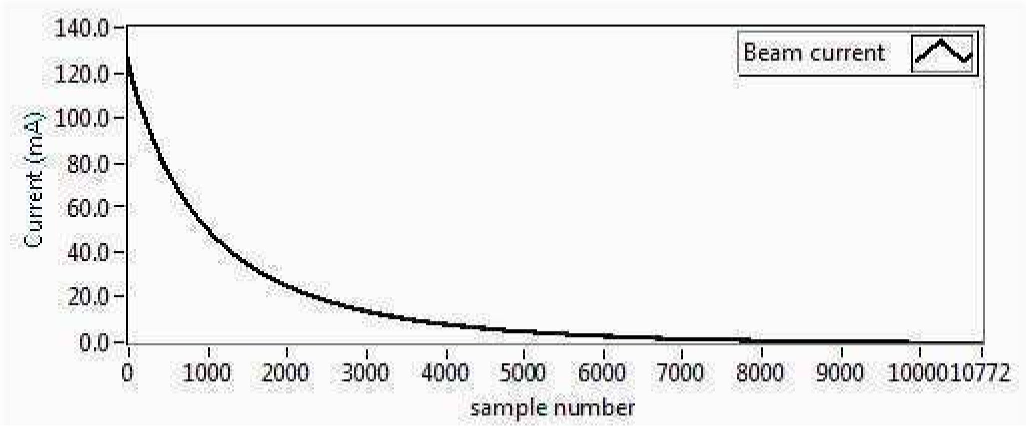


Figure 5.9: SR current decay curve.

throughout the TL-1 and provides beam image at all beam position indicators and the beam at TL-1 output.

***Booster model*:** The Booster model is developed as explained in chapter 3. It accepts the macro particle beam and depending upon the magnet set values it track all the macro particles in the beam for the stated number of turns (1000) and provides the beam at the output. After calculating the lost and survived particles it provides the normalised injection current i.e. the current which actually get injected into the booster out of the incoming electron beam. To keep the simulation simple the injection process is not calculated and is assumed that there is no loss at the injector portion and the entire beam successfully transfers from TL-1 output to Booster centre orbit.

***Storage ring model*:** The storage ring is modelled as the electron storing device based on the experimentally observed current decay curve shown in figure 5.9 where the current stored in the storage ring decays with time. The beam life time in storage ring is affected by many parameters such as vacuum quality, ion trapping, integrated SR beam time and RF voltage. Thus the beam lifetime is not constant and it changes throughout the beam decay process. Normally it is less at high beam currents and it increases as beam current

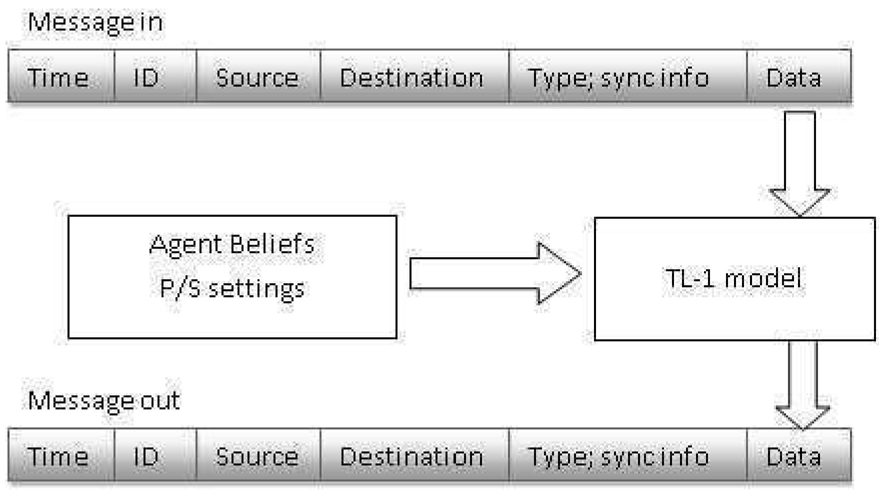


Figure 5.10: Block diagram for the *"LiveTL1Model"* capability at serial no 1 and 2.

decays to lower values. Thus a lookup table approach is adopted for modelling the current decay in storage ring. As the experimentally observed samples are spaced five seconds apart and simulation requires the current decay value at each second interval, the linear interpolation between observed data points is used. For calculating the current decay towards the lower current values where current decays very slowly this linear interpolation is applied between two monotonically decreasing points.

Multi-agent system*:** In the first phase of design only two agents named ***"LiveTL1Model" and ***"TL1 Agent"*** is designed with few functionalities. The ***"LiveTL1Model"*** agent is developed with capabilities listed at serial number 1 and 2 in table 5.2. Figure 5.10 shows the simplified block diagram for both of these capabilities. The ***"TL1 Agent"*** is developed with capabilities to fulfil the ***"Tune Beam Position and angle"*** goal listed in table 5.1

Simulation parameter limit calculation

To calculate the beam position variations observed in the normal Microtron operation, the data is observed for one complete period of cathode lifetime i.e. between the two

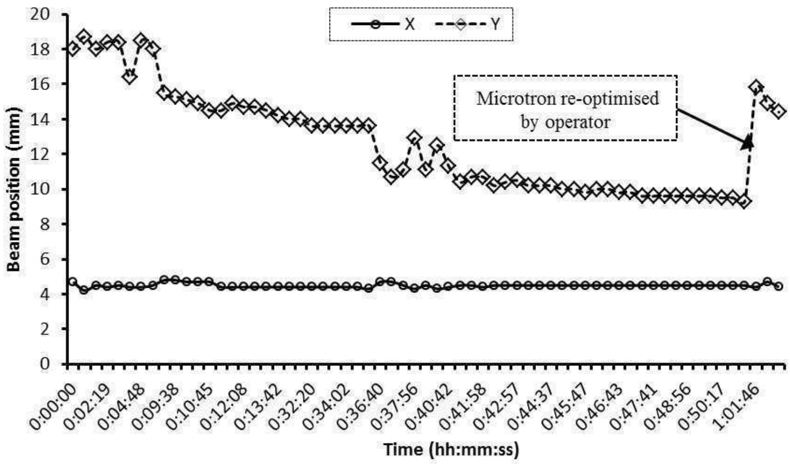


Figure 5.11: Drift observed in beam position at output of Microtron for one hour after starting.

cathode replacement times (23-8-2010 to 1-11-2010) and it is observed that electron beam coming out of Microtron shows beam position drift which can be broadly classified into two categories- category-1 and category-II. Here category-I is the drift which is observed at the time of starting of the Microtron for about an hour of operation before the Microtron stabilises for its normal operation state, this is shown in figure 5.11 this position drift mainly arises because of the shift in the RF cavity resonant frequency with the cavity temperature and slowly dies out as the cavity thermal stabilises with time. The category-II is the slow drift in the stabilised beam position seen on day to day basis shown in figure 5.12 this is mainly due to cathode ageing and day to day environmental variations. In normal course of operation the operator waits for the stabilisation time of about 30-40 minutes and then optimises the Microtron to obtain the desired beam position on beam slit monitor, if required the operator then adjusts the TL-1 power supplies to obtain the desired value of injection current in the booster synchrotron. In case of degradation of the injection current in the booster synchrotron the operator manually re-adjusts the power supply settings to obtain the desired value of injection.

From figure 5.11 one gets the variation in vertical plane spread from 10mm to 19mm

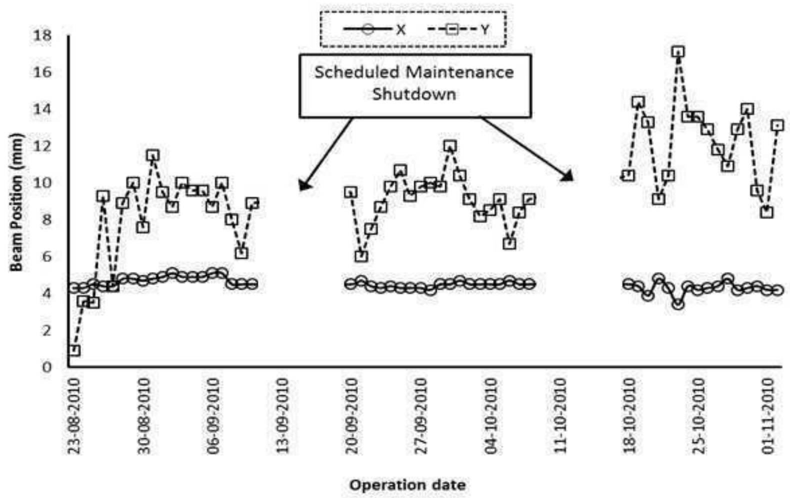


Figure 5.12: Drift observed in beam position at output of Microtron for complete cathode life on day to day basis.

where as in horizontal plane it is between 4mm to 6mm. similarly from figure 5.12 one gets the variation in vertical plane spread from 1mm to 18mm where as in horizontal plane it is between 3mm to 5mm. To simulate the operational characteristics of the electron beam, the beam position variation of $\pm 10mm$ in vertical plane and $\pm 1.5mm$ in horizontal plane will suffice but the variation $\pm 10mm$ in both the planes is taken for simulation purpose as the possibility of beam position variation in horizontal plane due to misalignment at the time of cathode replacement or other maintenance activity will favour the overall system by increasing the operational range in case of miss-alignment.

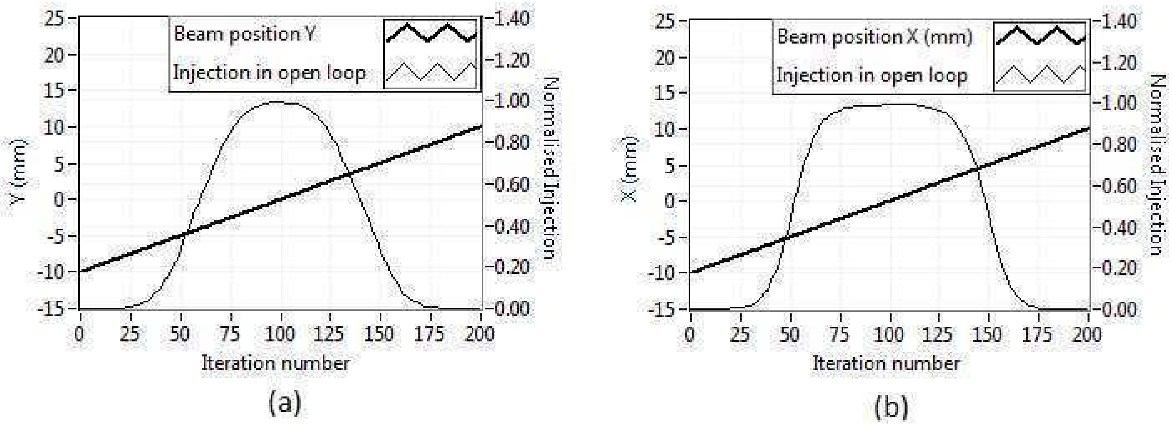


Figure 5.13: Injection in open loop with respect to beam position variation ,(a) for (y : $-10mm$ to $+10mm$) and (b) for (x : $-10mm$ to $+10mm$).

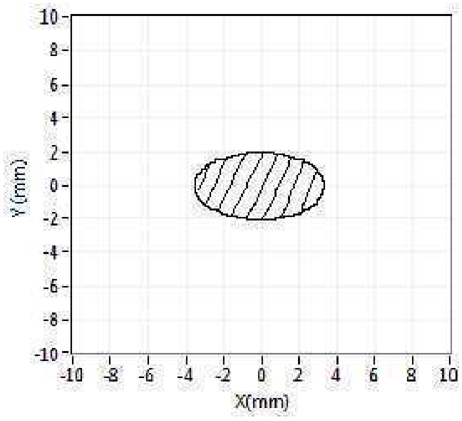


Figure 5.14: Normal operating area for beam at Microtron output for 90% injection criteria in open loop condition.

5.1.4 Machine behaviour in open loop

To simulate the machine behaviour in open loop for the beam position variations at the Microtron output in both the planes the simulation program is run to determine the system operation characteristics which are shown in figure 5.13(a) and 5.13(b) for beam position variation from $-10mm$ to $+10mm$ in both the planes i.e. in horizontal plane as well as in the vertical plane. The beam position is varied with a rate of $100\mu m$ per second with TL-1 P/S set to there design values. Further figure 5.14 shows the normal operating area for beam at Microtron output for 90% injection criteria in open loop condition i.e. if the beam lies within the bounded area, then the injection current in the booster will not fall below the 90% of its nominal value.

5.1.5 Machine behaviour with TL-1 intelligent agent based control

The machine control is simulated with TL-1 intelligent agent in feedback loop as shown in figure 5.7. TL-1 agent is built with ***"tune beam position and angle"*** goal with the plans listed at serial no 1 and 4 in Table 5.1. In simulation it is assumed that the machine is equipped with non-destructive type BPMs that can provide the beam position data online for every beam pulse.

Figure 5.15(a) shows the BR injection current when beam position at Microtron is varied from -10mm to +10mm with the rate of 100um/second, for the case of with and without applying the correction by the TL-1 intelligent agent.

In this figure one can observe that at the first simulation step the BR current is zero as the TL-1 is initially tuned at its normal operating value whereas the beam at Microtron output is displaced by -10mm and thus all the electrons are lost during injection process. In the second simulation step TL-1 agent correctly estimates the beam position at Microtron output using the beam images (at BPM-1,BPM-2 and BPM-3) and TL-1 model. It then successfully tunes the TL-1 to new operating point such that all the electrons are correctly injected into the booster and thus the BR injection current increases to its normal state. In the next few simulation steps the beam continues to move its position whereas TL-1 still remained tuned to its previous operating point as a result the injection current in booster starts decreasing. This decrease in injection current continues till it reduces to below 90% level. At this level the "TL-1 agent "invokes the plan to retune the TL-1. With the TL-1 tuned to new optimum operating point injection current is restored again. This process continues throughout the beam movement to +10mm value.

Similarly figure 5.15(b) shows the injection current with respect to the beam position variation in horizontal plane for the cases of with and without, applying the correction by the intelligent agent. In this figure it is seen that BR injection current is zero for the first simulation step because TL-1 is tuned to its normal operating value, whereas beam is displaced by -10mm. Similar to previous figure case in the second simulation step TL-1 agent tunes the TL-1 to new operating point. But for this case the BR injection current don't increase to its normal value, instead it remained even below the 90% value.

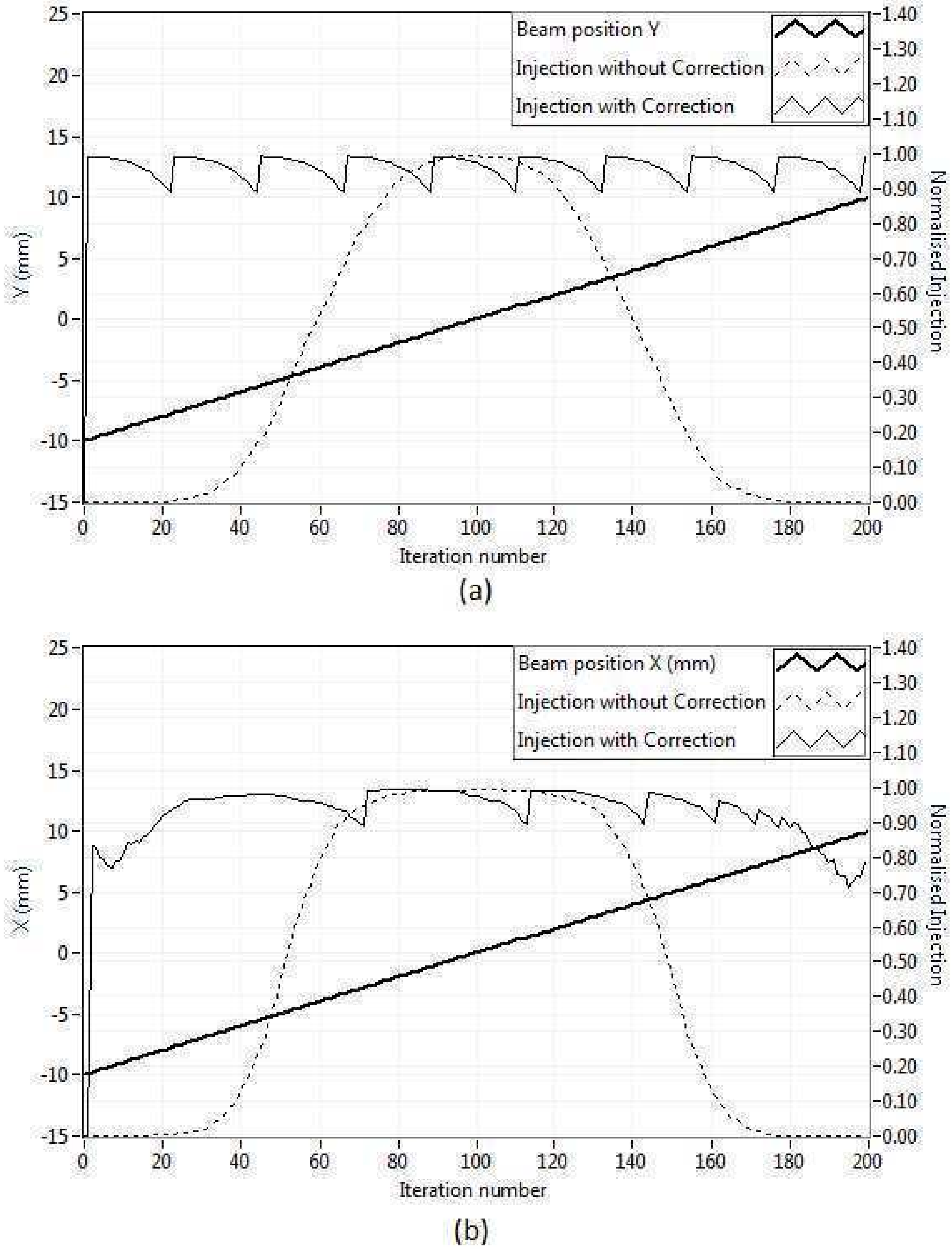


Figure 5.15: Injection with respect to beam position variation in (a) vertical and (b) horizontal plane with and without correction.

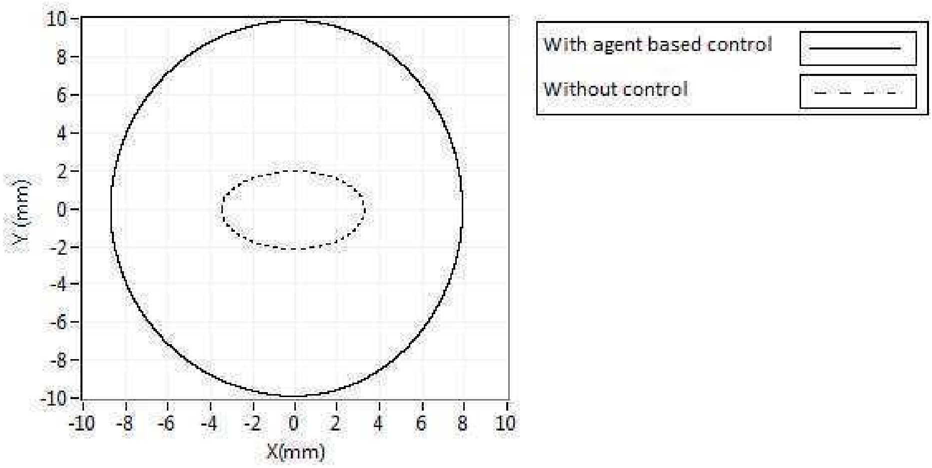


Figure 5.16: Normal operating area for beam at Microtron output for 90% injection criteria in close loop condition.

This has resulted because of the inability of the TL-1 corrector magnets to produce the magnet strength required to fully compensate the error in beam position. The corrector power supplies are having minimum and maximum current limits. In the next step as the injection current still remained below 90% injection value, agent again retunes the TL-1 at new optimised operating value for the corrector currents limited within their minimum to maximum current range. This process continues till around simulation step no 20 when the agent gets successful in correctly tuning the TL-1 with the criteria of injection current greater than 90% value, thus it stops tuning of TL-1. For next few simulation steps injection current shows variations depending on beam movement, till around simulation step 70, where the injection current falls below 90% that causes the agent to retune the TL-1 to new operating value.

From previous two figure results, the gain in the normal operating area for beam at Microtron output for 90% injection criteria obtained as a result of applying the close loop control is shown in Figure 5.16.

Considering that the assumption of simulating booster with the perfectly tuned condition may not be the case at all times and one may want to use the booster with different power supply settings. It is a requirement that the agent based control strategy should work in such a case also. To simulate this behaviour of booster working with different operating conditions it is assumed that the worst operating conditions will actually decrease the dynamic aperture available to the beam and hence this behaviour is simulated by inserting the physical aperture in the booster model. The simulation results of injection for different aperture vales with and without applying the agent based control for the beam position variations at Microtron output for booth planes is shown in figure 5.17 and figure 5.18

In figure 5.17(a) and 5.18(a) it can be seen that with the increase in the aperture size (i.e. with relaxed optics condition) the beam movement available to the system for given injection current criteria increases. This results in reducing down the total number of TL-1 tuning instances by the agent. This can be seen in figure 5.17(b) and 5.18(b) where the injection current curves for the case of larger aperture value show small number of rising edges. Further we see that with the reduction in the physical aperture size, the operation area for percentage (say 90% injection) injection criteria decreases but still the agent based control scheme shows improvement in the available operating area for beam. Hence this scheme can be used in the cases where booster is also allowed to be tuned to new operating values. Further the effect of applying this correction on the stored current in the storage ring (SR) is simulated and the results are shown in figure 5.19. In this figure one can see that for the case of agent based closed loop control as the BR injection current becomes available from the starting (i.e. when beam is displaced to -10mm) and is maintained at higher values by the agent through retuning of TL-1. The rate of accumulation of current in SR remained high for the complete beam movement simulation phase. This results in higher value of achievable current in SR for the agent based closed loop control scheme as opposed to the open loop case. Thus with agent based closed loop control scheme the

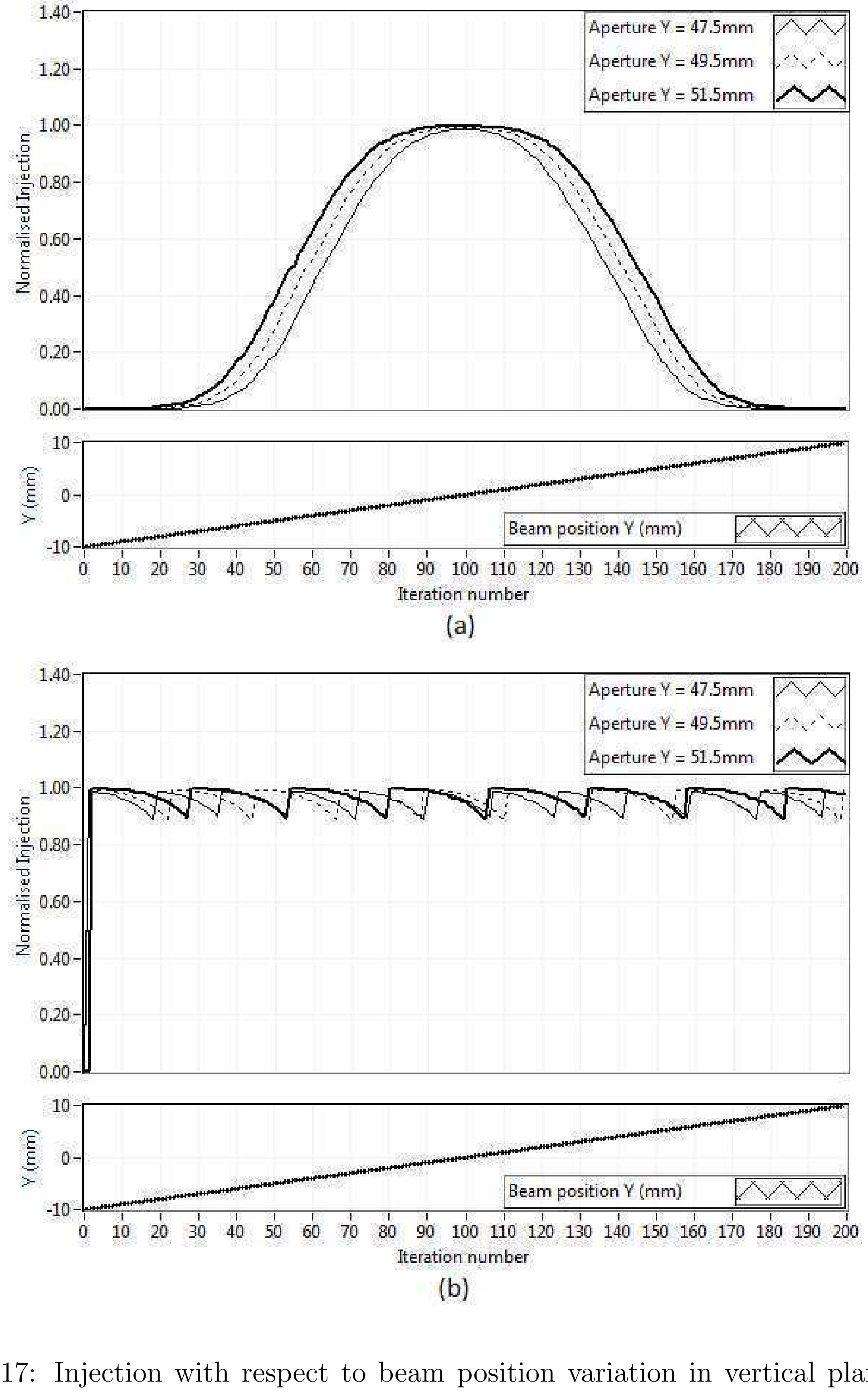


Figure 5.17: Injection with respect to beam position variation in vertical plane,(a) for system in open loop, and (b) for system in close loop, for different aperture values.

rate of injection in SR can be increased and maintained for the cases of beam position variations. The assumption of machine equipped with non-destructive type of BPM with facility of providing beam position online at every pulse may not be the cases in some of

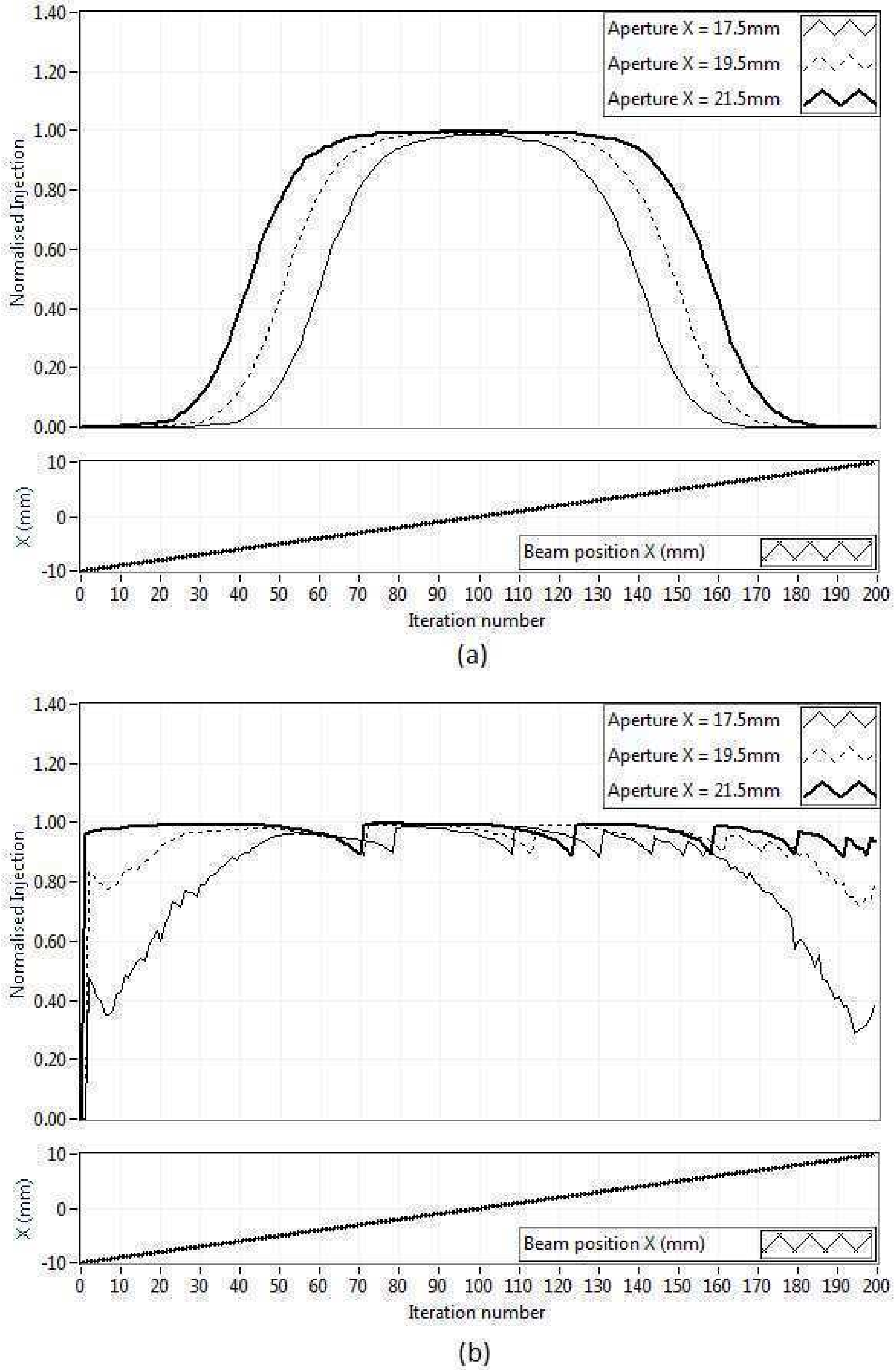


Figure 5.18: Injection with respect to beam position variation in horizontal plane ,(a) for system in open loop, and (b) for system in close loop, for different aperture values.

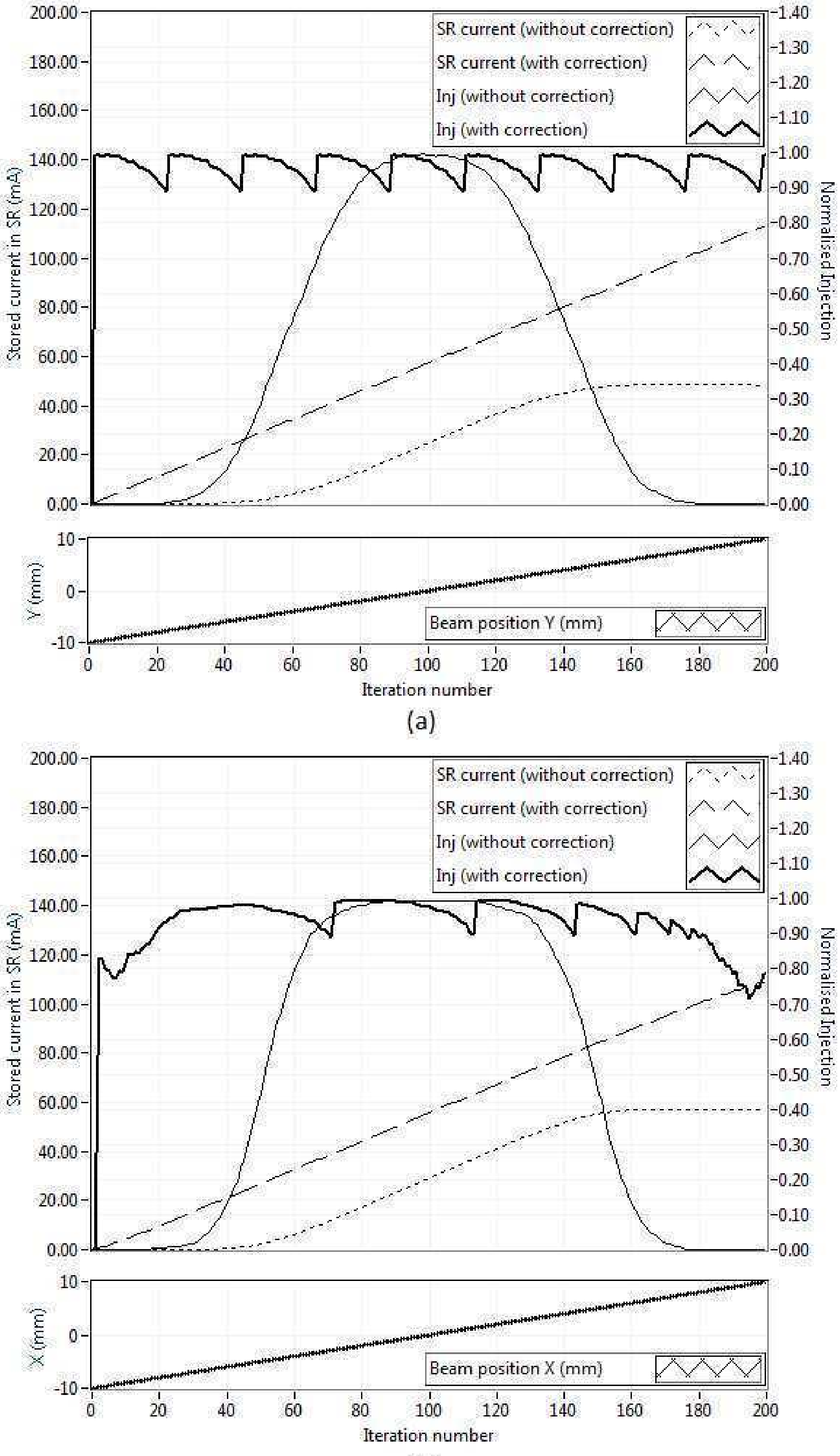


Figure 5.19: Injection and SR current with respect to beam position variation in, (a) Vertical plane, and (b) Horizontal plane, for with, and without correction.

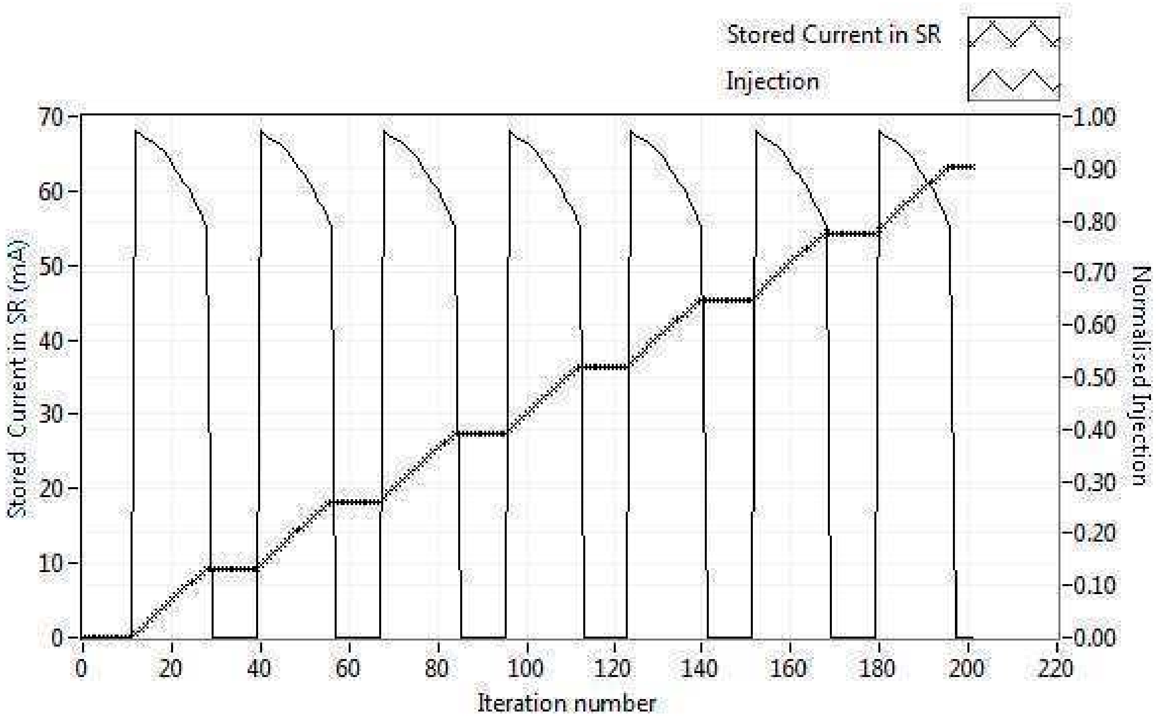


Figure 5.20: Injection and SR current for beam position variation ($y : -10mm$ to $+10mm$) considering the measurement delay of 10 seconds.

the machines like our TL-1 case. In TL-1 the fluorescent type BPM are employed which are destructive type and takes about 10 to 20 seconds of time for one measurement. Thus to obtain the effectiveness of the above stated scheme for the machines with destructive type BPM the system is simulated with a 10 seconds BPM measurement time for beam position variation in vertical plane from $-10mm$ to $+10mm$ with a $100\mu m$ per second beam shift and the low injection limit value of 80% . The simulation result in the form of booster injection current and SR stored current is shown in Figure 5.20. The results shows that the choice of going for the beam position measurement decision places a penalty of non-availability of beam pulse for the time equal to the measurement delay. A higher value of injection-low-limit will cause the agent to start the corrective action in early stage of beam current decay, thus this will increase the number of corrective actions taken by agent. Therefore the choice of selecting the appropriate value for the injection-low-limit and the measurement delay introduced by BPM will affect the stored current in SR this is shown in Figure 5.21. In this figure one can see that as the injection-low-limit value is increased initially, the increase in the corrective actions by the agent causes the stored SR current to increases. This increase in SR current continues till a maximum value after which the phenomenon of beam decay during the measurement period (as during measurement beam is not available) starts dominating and as a result from this point onwards the SR current starts decreasing with increase in the injection-low-limit value. Further this decrease in SR current becomes more pronounced as the measurement time increases from 3 seconds to 15 seconds. As a reasonable choice of parameter one has to choose the lowest injection-low-limit for getting the reasonable amount of SR stored current as the lower value of injection-low-limit will reduce the measurement frequency and hence will be a relaxing condition for the BPM equipments thus increasing there life. From Figure 5.21 it is clear that with the plans incorporated in the intelligent agent behaviour so far can not compensate for the reduction in the stored current in SR resulting from the measurement delays and hence now the other stated plans at serial number 2 and 3 are being incorporated. These plans will be based on the beam position tracking and correction based on the injection current variation feedback and history data thus will try to minimise the frequency of measurement hence will eliminate the effect introduced by the measurement delays.

Discussion

In this section specifically the agent based control framework developed for implementation of TL-1 intelligent agent is described along with the developed accelerator system simulation framework comprising of electron beam input to the TL-1, TL-1 model, Booster model, and the storage ring model. Then the goals and associated plans are formulated for the model based goal based architecture of Tl-1 intelligent agent. The system behaviour in open loop (i.e. without agent based active control) for the simulated disturbance in the

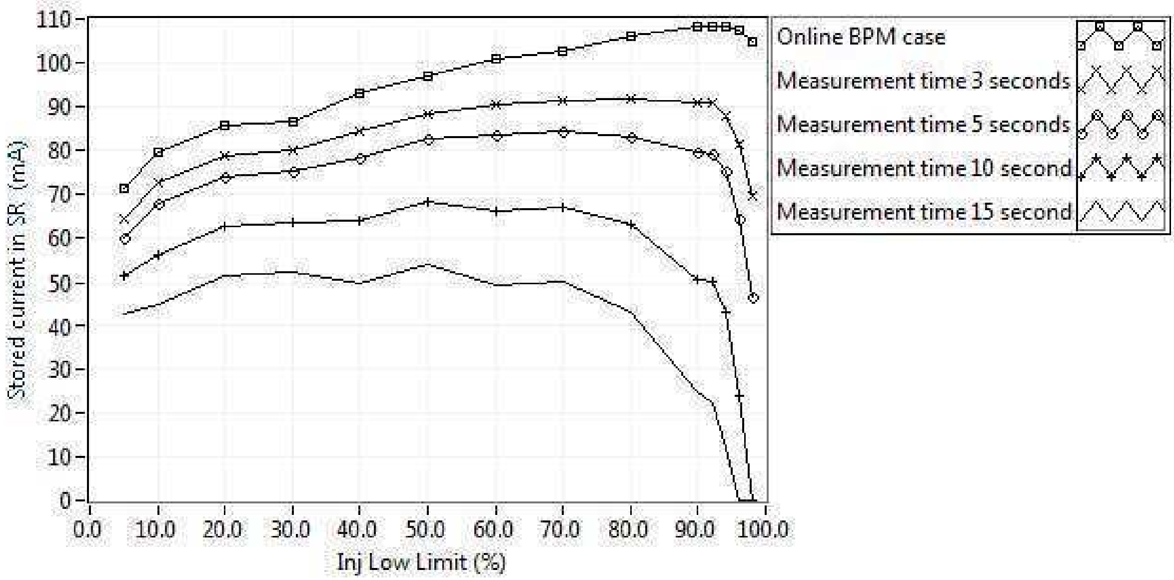


Figure 5.21: SR current for beam position variation ($y : -10mm$ to $+10mm$) with respect to Inj-Low-Limit for different measurement delay values.

electron beam at TL-1 input is captured. To obtain the normal useful operating parameter boundaries for this system. Then the simulation trials are run to capture the system characteristics for the case of closed loop control with developed Tl-1 agent controlling the power supplies of Transport line-1 automatically with the active goal of keeping the injection current in booster above 90% for the simulated disturbance in the electron beam at TL-1 input. The simulation results show that the developed TL-1 agent can successfully tune the Transport line -1 of Indus complex for the case of normally observed beam disturbance in the electron beam at TL-1 input. This means that the developed agent if is allowed to automatically control the TL-1 power supplies will relieve the operator from the job of retuning of TL-1 and to same extent Microtron in there day to day operation. This will also improve the beam quality as the intelligent agent tune the TL-1 quickly and near optimal operating point as soon as the beam current degrades below 90% limit as opposed to transport line tuning by the human operators. This continuously maintained high injection current value will result in shorter filling time to accumulate the injection current in Indus-1 and Indus-2 storage rings. Also the simulation results shows that the beam obstruction caused during the measurements by the existing destructive type of beam position monitors of TL-1 at Indus complex affects the electron accumulation in storage ring adversely to the significant level and thus a non-destructive online type of beam position indicators is advised for actual implementation of the agent based control system on this machine. Also the search for new methods based on some sort of system state retrieval/tracking in the dynamic environment that can provide the system state information using the initial system state and the online available signal like Booster current seems to be the obvious methods that can be further explored towards minimizing the measurement time.

5.1.6 Integration of Model based tracking in TL-1 intelligent agent

In the previous sections it is seen that the agent based transport line control can well extend the systems operating range for the case of beam position variations observed in the electron beam at Microtron output. Another possible benefit of agent based control of transport lines is investigated for improving the system reliability by making plans for handling the sensor and actuator failure scenarios. For this the concept of model based tracking is utilised with the agent based control systems of transport lines. The concept of agents decision making based on the tracking principle has been utilised in past by different researchers. Scheme in [124] shows the advantage of agent's decision making based on the tracking of moving target while avoiding collision with moving obstacles and other agents. Yan et al. in [125] extended this work by applying constraints on input and formulated a local and dynamic optimal algorithm. Here each agent exchanges information only with its neighbours with optimization implemented at each update cycle. The model based tracking (MBT) is a well known principle, primarily used in the field of image processing,[61, 64, 121] for reconstruction of three dimensional object information (position and orientation) from two dimensional image of the scene. In this technique the model consists of precise three dimensional geometrical representations of known object (mainly vehicle), which can be placed in arbitrary positions and orientations, together with a carefully constructed camera model and scene model. Using this model, and given provisional position and orientation, the three dimensional object is then projected onto the two dimensional image plane and a goodness-of-fit score is obtained by comparing the modelled features with the acquired image. A search in position-space and orientation-space is then used to maximize this evaluation score. At each position and orientation in this search the model is re-instantiated onto the scene and a new goodness-of-fit score is evaluated. Once a maximum score is found the three dimensional position and orientation of the object is known and is used to predict a provisional position and orientation for the same object in the next frame of the scene. Thus by having position of an object in the initial frame, it is possible to track that object in subsequent frames along with its location and direction of travel in three dimensions. This same concept is used here for tuning of accelerator transport lines under limited sensor/actuator failure cases for agent based accelerator control systems. Plans for transport line tuning under various BPM failure conditions for goal-based intelligent agent with modular architecture have been formulated using the concept of model based tracking. Model-based tracking concept is used to track the beam parameter vector at start of transport line using the injection current as feedback instead of data from beam position monitors. This allows the agent to correct injection, even in the absence of all BPM data. The effectiveness of the proposed scheme is investigated through simulation. Particularly, the problem of beam position variation at source is simulated for agent architecture with new MBT plans applied to the Indus accelerator model comprising of Microtron output, Transport line-1 (TL-1) and

Booster.

System simulation

The simulation program is comprised of three main parts. These are the agent system, the model of the accelerator environment with which the agents interact, and the graphical user interface (GUI). The overall organisation of the simulation is same to what is discussed in Sec. 5.1.2. The accelerator model is made in the form of a software entity comprising of three sub-modules. The agent system comprises of a set of agents and inter-agent communication facility. For synchronizing between the accelerator model and the agent system, messages are used. The data exchange between programs is done through messages and global variables.

***Agent system*:** The agent system is mainly composed of two agents "TL-1 control agent"(TL1CA) and "model agent"(MA). TL1CA interprets the system state from percepts. Depending upon the current events and the agent beliefs, the agent decides the plans to be executed to achieve all the active goals. The agent evaluates the plan applicability function and selects the highest priority applicable plan from the list for each active goal. The plan list of the TL1CA is extended by adding the plans stated in this section ahead to the TL tuning goal of this agent. MA contains the model of TL-1, Booster, and Storage ring in its body and assists TL1CA in its plans by providing the information about the probable outcome of the actions on the machine. To support the newly added plans of TL1CA, three more capabilities (predict position with BPM, predict position using MBT with BPM, predict position using MBT without BPM) are added to the function list of this agent.

***Accelerator model*:** The accelerator model comprises of three sub-models: Microtron model, TL-1 model and Booster model. The Microtron model generates a beam consisting

Table 5.5: Value of beam parameters used in simulation.

Parameter	x	y
$emittance(e)$	8.0×10^{-7}	3.0×10^{-6}
α	-0.6708	-0.5046
β	0.9645	0.9945
δ	0.00	0.00
$Angle(x'/y')$	0.00	0.00
$position$	0.00	varied between $\pm$10mm
e_{noise}	$200\mu m$(RMS)	$200\mu m$(RMS)

of 2000 macro particles. The various parameters of the beam are e_x,e_y, α_x,α_y,β_x,β_y,δ_x,δ_y, x', y', x and y. Table 5.5 gives the value of different beam parameters used in this simulation. The TL-1 model is developed using MAD and is discussed in Chapter 2 in detail. It accepts beam comprising of macro particles with different beam parameters. Depending upon the set values of all the power supplies it tracks all the macro particles in the beam along the TL-1. It provides beam image at all beam position indicators and the beam parameters at TL-1 end. The booster model used here is already discussed in chapter 2. It accepts the macro particle beam with beam parameters at TL-1 end. Depending upon the magnet settings, it constructs the machine lattice. Using this lattice it then tracks the particle for the defined number of turns (1000 turns) and produces the survived/lost attribute for each macro particle. Depending upon the survived/lost condition of macro particles it provides the normalised beam current injected into the booster. To keep the simulation simple the injection process is not calculated and is assumed that there is no loss at the injector portion and the entire beam successfully transfers from TL-1 output to Booster centre orbit.

***Graphical user interface*:** The simulation program is developed with GUI shown in Fig.5.22. to allow the user to conveniently monitor and control the simulation. Separate

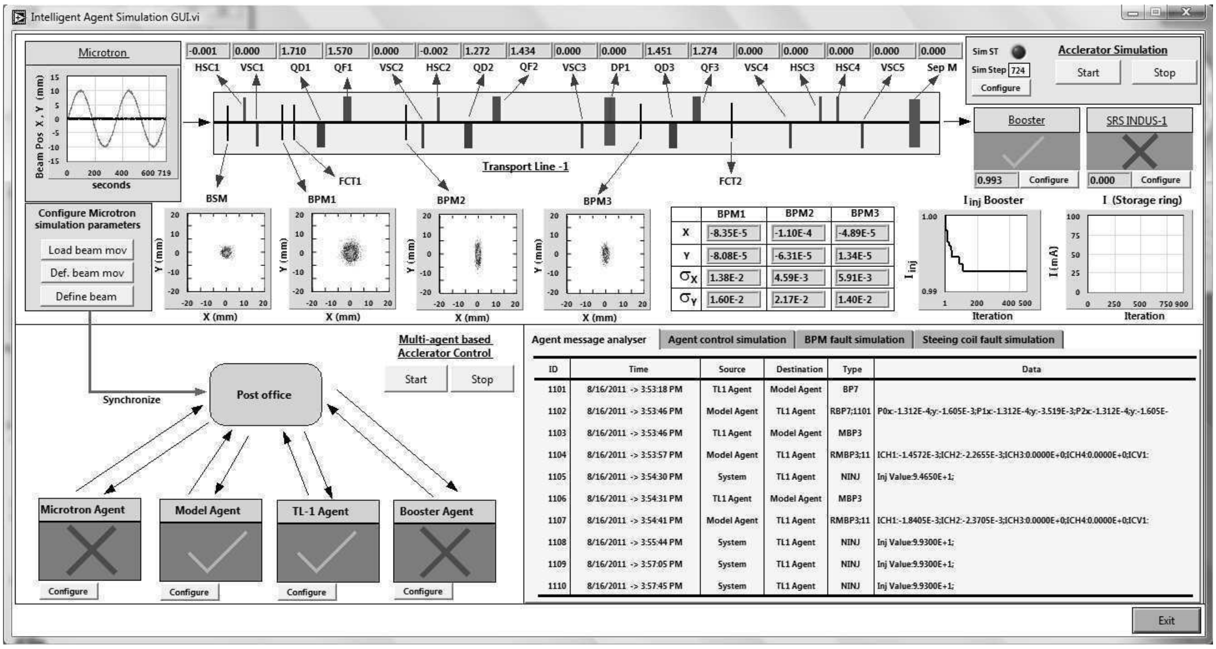


Figure 5.22: GUI for the agent based accelerator control simulation.

windows are developed for defining beam, defining beam movement, configuring accelerator simulation parameters, configuring agent parameters, and configuring agent based simulation parameters. Results of simulation can be viewed in graphic control panel and can also be stored in text file. For debugging, agent communication analyser is given in tab control. Using GUI different fault conditions of devices can be introduced manually as well as can be defined through configuration file.

TL tuning using model based tracking

GA based TL tuning method using the beam position feedback from BPMs is discussed by Schirmer et al.[103]. For considering the cases of BPM/actuator failure scenarios, this method is modified using the model based tracking concept. From methodology point of view the BPM/actuator failure scenarios are divided into four cases, case 1: when some of the BPMs fail, case 2: when only one BPM is available, case 3: when all BPMs fail, and case 4: when actuators fail. Further the case 4 may simultaneously occur along with any one of the cases, case 1 to case 3.

$$fitness_{DE-I} = \sum_{i=1}^{n}(a_i\breve{B}_{i,mes} + b_i\breve{B}_{i,stor} - \breve{B}_{i,sim})^2 \tag{5.1}$$

$$a_i = 1, b_i = 0, \left[\breve{B}_{i,stor}\right]_{n+1} = \left[\breve{B}_{i,mes}\right]_n \qquad \text{if } BPM_i = \text{Normal}$$

$$a_i = 0, b_i = 1, \left[\breve{B}_{i,stor}\right]_{n+1} = \left[\breve{B}_{i,MSO}\right]_n \qquad \text{if } BPM_i = \text{Faulty}$$

$$fitness_{DE-II} = fitness_{DE-II(x)} + fitness_{DE-II(y)} \tag{5.2}$$

$$fitness_{DE-II(x)} = M_0 + m_1M_1 + m_2M_2 + m_3M_3 + m_4M_4 + m_5M_5 + m_6M_6$$

$$M_0 = \sum_{i=1}^{4}(\widetilde{X}_{end,opt}(i) - \widetilde{X}_{end,sim}(i))^2$$

$$M_1 = (x_{HSC2(start),opt} - (x_{HSC2(start),sim}))^2$$

$$M_2 = (x'_{HSC2(end),opt} - (x'_{HSC2(end),sim}))^2$$

$$M_3 = (x_{HSC3(start),opt} - (x_{HSC3(start),sim}))^2$$

$$M_4 = (x'_{HSC3(end),opt} - (x'_{HSC3(end),sim}))^2$$

$$M_5 = (x_{HSC4(start),opt} - (x_{HSC4(start),sim}))^2$$

$$M_6 = (x'_{HSC4(end),opt} - (x'_{HSC4(end),sim}))^2$$

similarly the $fitness_{DE-II(y)}$ is defined for vertical plane

TL-1 tuning using BPM

For tuning of TL-1 using feedback data from BPM the agent based control (ABC) utilizes the three step Differential evolution [112](DE) based scheme with fitness functions given by Eq.(5.1) for DE-I and Eq.(5.2) for DE-II. The Eq.(5.1) for fitness function DE-I is derived based on the fact that the best assessment of beam parameters at the TL-1 start point will be the one, that produces the simulated values of beam position at BPM1, BPM2, BPM3, which gives the minimum value for sum of square errors. Similarly for fitness function DE-II the Eq.(5.2) are derived from the criteria that the best magnet current settings for the TL-1 tuning will be the one that satisfies two conditions, first: that produces the minimum value of sum of square errors between the declared optimum beam parameters and the simulated beam parameters at TL-1 end point. And second: that the provision has be provided for selectively minimise the difference between declared optimum and simulated value of beam position & angle at all the corrector positions throughout the TL-1. Where the two best correctors to participate in beam position correction action can be chosen according to the then observed situation by the agent. For handling the case of sensor failure TL1CA dynamically selects the BPM (the symbol $\breve{B}$ is used to represent BPM) data source by suitably selecting the value of a_i and b_i depending upon the sensor's health/failure condition. For the faulty BPM the agent selects the data from the data store whereas for the correct BPM agent selects the recent measurement data. Similarly for handling the actuator failure cases the agent dynamically selects the value of m_i, depending upon the actuator normal/failure condition.

In the first step when operator defines the good injection event, the BPM data and set values of magnet power supplies are read. Then using TL-1 model, the beam position vector at the start TL-1, and end TL-1 are calculated using the DE-I by mutating the start orbit vector $\widetilde{X}_{start}$ and are stored as $\widetilde{X}_{start,opt}$ and $\widetilde{X}_{end,opt}$. These represent optimized beam parameters in good injection condition. In the second step whenever the injection degrades, the agent calculates the new beam position vectors at start TL-1 and end TL-1 as $\widetilde{X}_{start,mes}$ and $\widetilde{X}_{end,mes}$. In the third step using $\widetilde{X}_{start,mes}$ and $\widetilde{X}_{end,opt}$, it then calculates the suitable magnet settings $\widetilde{M}$ for matching the real beam parameters at the input with the optimized beam parameters at the output using DE-II.

$$\hat{A} = f(I_{inj,lim}, \widetilde{X}_{end,opt}, \widetilde{M}_{BR}) \tag{5.3}$$

$$I_{inj} > I_{inj,lim} \tag{5.4}$$

$$\hat{A}_{start} = Model_{TL}^{-1}(\hat{A}) \tag{5.5}$$

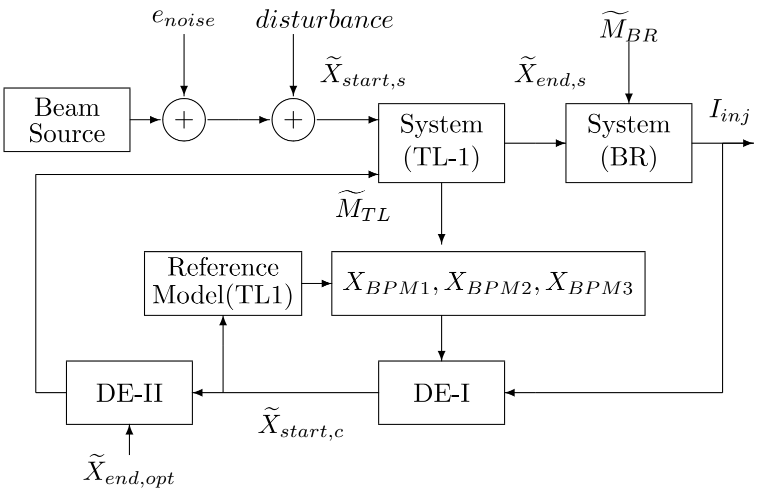


Figure 5.23: Control system block diagram for case 1.

$$\mathring{e}_{DE-I} = f(\mathring{e}_{BPM1}, \mathring{e}_{BPM2}, \mathring{e}_{BPM3}, \triangle t_1) \tag{5.6}$$

$$\triangle t_1 = t_{last\ corr} - t_{inj\ BL} \tag{5.7}$$

$$\mathring{E}_{start} = \mathring{e}_{DE-I} + e_{noise} + \frac{d\widetilde{X}_{start,s}}{dt}\triangle t_{cal} \tag{5.8}$$

$$\mathring{E}_{start} < \hat{A}_{start} \tag{5.9}$$

Tuning condition: To analyse the scheme of ABC system for this case the block diagram for overall control system is shown in Fig.5.23. To find the condition for successful tracking of beam and tuning of TL-1, let the rate of change of beam parameter at start TL-1 due to un-modelled dynamics/disturbance be given by $\frac{d\widetilde{X}_{start,s}}{dt}$. Let $\hat{A}$ given by Eq. (5.3) be the area at end TL-1 in vector space that will tune the TL-1 as per Eq. (5.4) by applying magnet settings $\widetilde{M}_{TL}$. Let $\hat{A}_{start}$ be the corresponding tuning area transformed to start TL-1 using TL-1 inverse model given by Eq. (5.5). Let X_{BPM1}, X_{BPM2}, X_{BPM3} be the beam position vector at corresponding BPM locations (obtained by direct measurement or from model-based tracked data) with errors $\mathring{e}_{BPM1}$, $\mathring{e}_{BPM2}$, $\mathring{e}_{BPM3}$ respectively, then the error ($\mathring{e}_{DE-1}$) in estimation of $\widetilde{X}_{start,c}$ by DE-I is given by Eq. (5.6) when $\triangle t_1$ is given by Eq. (5.7) where $t_{last\ corr}$ represents the time instance when last correction was applied and $t_{inj\ BL}$ is the time instance when injection degraded below limiting value. The $\triangle t_{cal}$

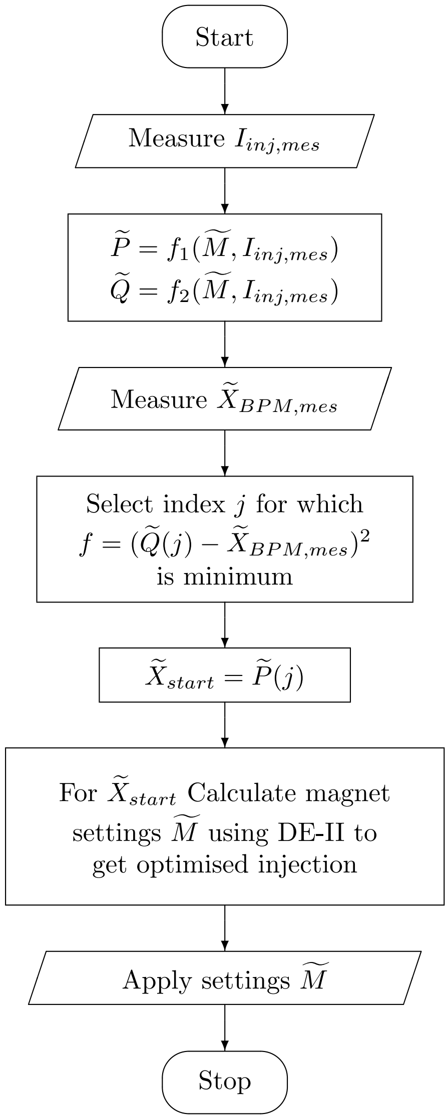


Figure 5.24: Flow chart for TL-1 tuning with only one BPM data.

is the calculation time taken by DE-I and DE-II. Then the error ($\mathring{E}_{start}$) in the estimation of beam parameter at start TL-1 at the time of applying the correction will be given by Eq. (5.8). The condition for successful tuning will be given by Eq. (5.9).

TL-1 tuning with only one BPM data

For tuning of TL-1 in case when only one BPM is available, the TL1CA uses the extended system model, which incorporates the booster model along with the TL-1 model. This extended model provides the calculated injection current value. In this case the agent

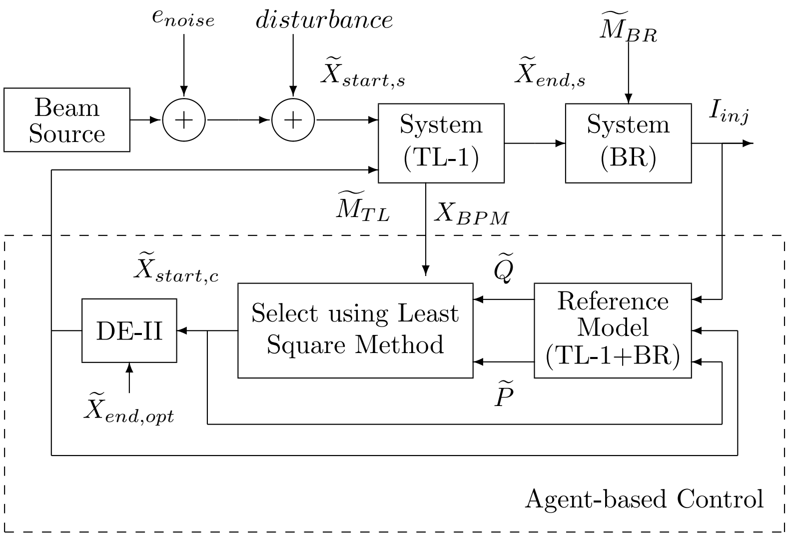


Figure 5.25: Control system block diagram for case 2.

calculates beam position vectors at start TL-1 ($\widetilde{P}$) and all the available BPMs ($\widetilde{Q}$) for injection current $I_{inj,mes}$. From $\widetilde{P}$ it then selects one position vector say $\widetilde{X}_{start}$ which gives the calculated values at the available BPM, nearest to the measured value. It then uses $\widetilde{X}_{start}$ for calculating the magnet settings ($\widetilde{M}$) for optimising the injection current in the same way as in case 1. The flow chart for the overall process is shown in Fig.5.24

Tuning condition: To analyse the scheme of ABC system for case 2 the block diagram for overall control system is shown in Fig.5.25. Let $\mathring{e}_{BPM}$ be the error in measured beam position vector X_{BPM} and let $\mathring{e}_{model}$ be the error in estimation of $\widetilde{X}_{start,c}$ using reference models of TL-1 and booster. Let $\triangle t_{cal}$ be the calculation time taken by ABC. The error in the estimated beam parameter at start TL-1 at the time of applying the corrections will be given by Eq. (5.10). The condition for successful tuning is given by Eq. (5.9). Here it is to be noted that theoretically it is possible to generate $\widetilde{Q}$ and $\widetilde{P}$ using $\widetilde{M}_{TL}$, $\widetilde{M}_{BR}$, and I_{inj} but the use of tracked $\widetilde{X}_{start}$ decreases t_{cal} and therefore it decreases $\mathring{E}_{start}$.

$$\mathring{E}_{start} = \mathring{e}_{BPM} + e_{noise} + \mathring{e}_{model} + \frac{d\widetilde{X}_{start,s}}{dt}\triangle t_{cal} \tag{5.10}$$

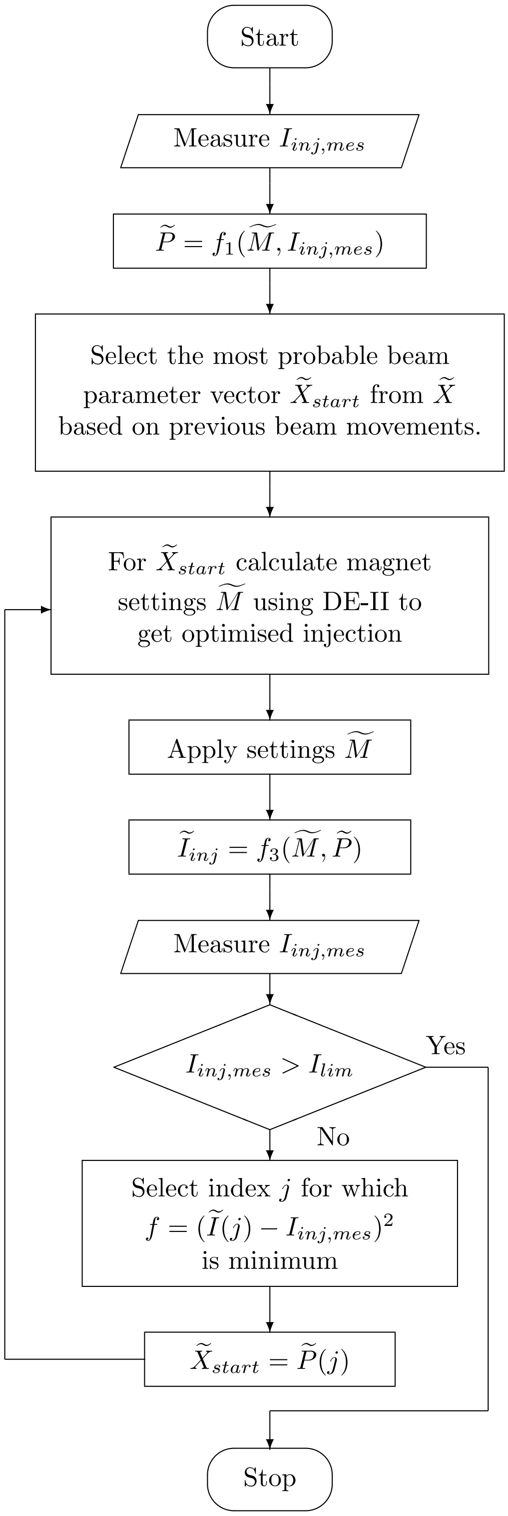


Figure 5.26: Flow chart for TL-1 tuning in total absence of BPM data.

TL-1 tuning in total absence of BPM data

For tuning of TL-1 in total absence of BPM data TL1CA calculates all the possible beam position vectors at start TL-1 $\widetilde{P}$ for the currently measured injection current value. Then based on the historical trend of beam position movement, it selects the most probable beam position vector $\widetilde{X}_{start}$. Using $\widetilde{X}_{start}$ it then calculates the magnet settings $(\widetilde{M})$ with DE-II and applies it to system. Using $(\widetilde{M})$ it then calculates the injection current $(\widetilde{I})$ for all elements of $(\widetilde{P})$. Using $\widetilde{I}$ and the newly measured injection current value $(I_{inj,mes})$ it then refines its decision of selecting the most probable beam position $\widetilde{X}_{start}$. The overall process for this is shown by the flow chart given in Fig.5.26.

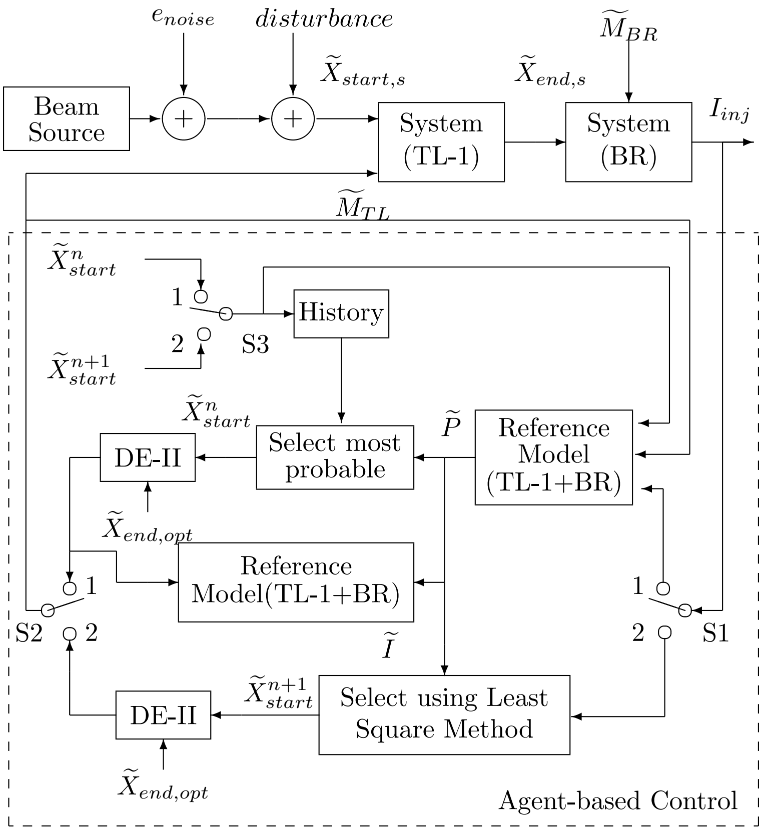


Figure 5.27: Control system block diagram for case 3.

Tuning condition: To analyse the scheme of case 3, the control system block diagram is shown in Fig.5.27. Let $\triangle t_{cal,1}$ be the calculation time in step 1, i.e. when all the three switches (S1, S2, S3) are in position 1 and the flow of control passes through various blocks in the order - *Reference model(TL+BR)* block, *Select most probable* block, *DE-II* block and *System(TL-1)* block. Let $\triangle t_{cal,2}$ be the calculation time in step 2, i.e. when all the three switches (S1, S2, S3) are in position 2 and the flow of control passes through various blocks in the order - *Select using least square method* block, *DE-II* block and *System(TL-1)* block. For calculating the condition for successful tracking, assume that the TL-1 is

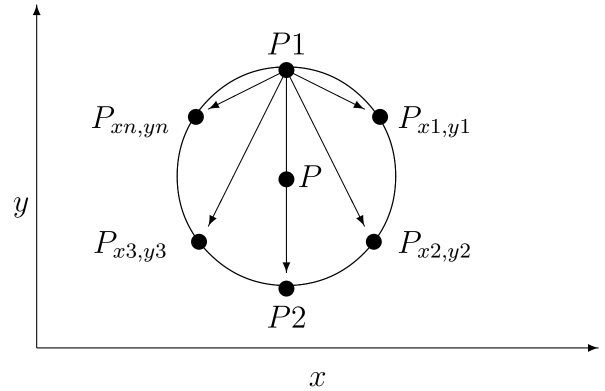


Figure 5.28: TL-1 operating points corresponding to $I_{inj,lim}$.

tuned to operate at point P (see Fig.5.28) by the last successfully applied correction. Then say P1, P2, $P_{x1,y1}$, $P_{x2,y2}$, ..., $P_{xn,yn}$ elements of $\widetilde{P}$ be the points corresponding to $I_{inj,lim}$. Let P1 be the point corresponding to actual beam position at the time when correction is demanded. Considering the worst case that the algorithm of selecting the most probable operating point wrongly selects the point P2 which is at the maximum distance from P1 (see Fig.5.28). The I_{inj} can now be plotted for points P, P1, P2 as shown in Fig.5.29.

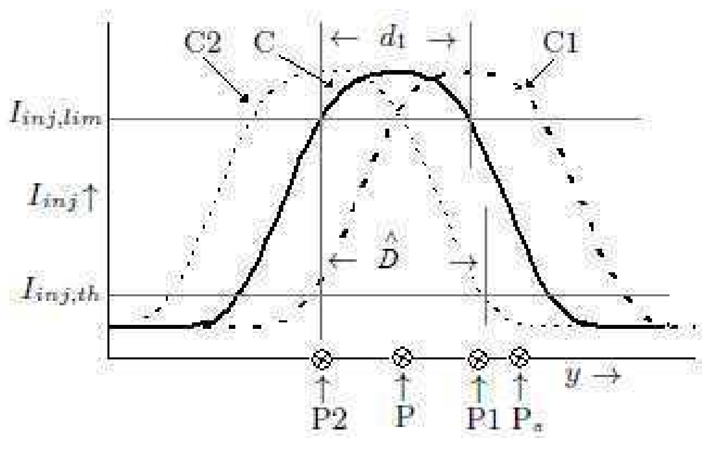


Figure 5.29: I_{inj} for TL-1 tuning corresponding to points P, P1, P2.

The curves C, C1, C2 corresponds to the I_{inj} when the TL-1 is tuned at point P, P1, P2 respectively and the electron beam at TL-1 input move in vertical plane. Let $\hat{D}$ be the maximum distance from the tuned position corresponding to the $I_{inj,th}$, when $I_{inj,th}$ is the threshold limit of I_{inj}, using which the point P1 can be successfully identified by the system in presence of noise and errors. Let d_1 be the distance between P1 and P2 given by Eq. (5.11) and d_2 be the distance moved by the beam during the calculation time $\triangle t_{cal,1}$ because of disturbance/un-modelled dynamics given by Eq. (5.12) then the condition for successful tracking will be given by Eq. (5.13). Now let P_s be the point corresponding to actual position of beam at the time when the magnet setting ($\widetilde{M}$) are applied to the system as a result of step 2, then the condition of successful tuning will be given by Eq. (5.15).

$$d_1{=}f(\hat{A}_{start}) \tag{5.11}$$

$$d_2{=}\tfrac{d\widetilde{X}_{start,s}}{dt}\triangle t_{cal,1} \tag{5.12}$$

$$d_1 + d_2 < \hat{D} \tag{5.13}$$

$$\triangle t = \triangle t_{cal,1} + \triangle t_{cal,2} \tag{5.14}$$

$$e_{noise} + \mathring{e}_{model} + \frac{d\widetilde{X}_{start,s}}{dt}\triangle t < \hat{A}_{start} \tag{5.15}$$

TL-1 tuning when actuators fail

For handling the case of actuator failure scenarios TL1AC dynamically selects the values of m_i depending upon the actuator normal/failure condition using the simple rule set.

Rule 1: find the two normal correctors ($\hat{C}1$ and $\hat{C}2$) coming first in the direction of beam traversal.

Rule 2: use $\hat{C}1$ to correct the beam position at $\hat{C}2$ start location.

Rule 3: use $\hat{C}2$ to correct the beam angle at $\hat{C}2$ end location.For simplification the following assumptions are made in implementation and simulation of the above stated plans.

Assumption 1: It is assumed that the device failures can be identified by the system.

Assumption 2: Although the BPM may fail in different ways, in this work it is assumed that the BPM failure only causes the loss of BPM data and does not affect the operation of system in any other way.

Assumption 3: Although the correctors may fail in different ways, in this work it is assumed that the failure of corrector is equivalent to switching off of power from the corrector power supply and does not affect the system in any other way.

Assumption 4: For case 3 it is assumed that initially either the system is in tuned condition or the initial beam position (only rough estimate) is known.

5.1.7 Methods for Stability

The conventional feedback control loops may become unstable if controller parameters are not chosen properly or in the influence of external dynamics. The agent based control provides opportunity for maintaining system in stable states through limit based event evoking (system alarms) where the plans can be made to handle these system events that indicates the likelihood of system approaching towards potentially unstable system states much before the system actually acquires such state. Also for recoverable systems, by maintaining the previous action history and state history with the state qualification methods the agents can easily recover the system from unstable state through action reversal. Although the cases discussed in this section are all of recoverable type, for non-recoverable systems the agent plans can be built that allows the agent to control the system for limited states.

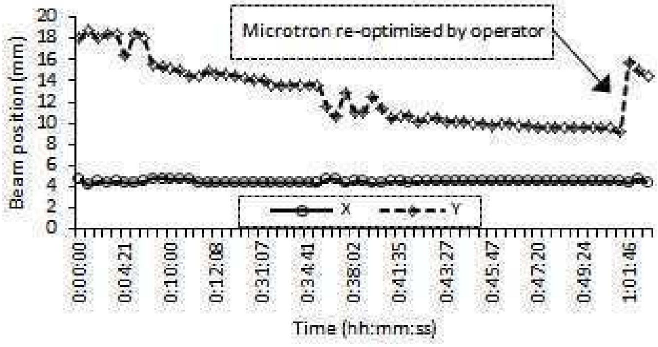


Figure 5.30: Beam position variation measured near TL-1 start.

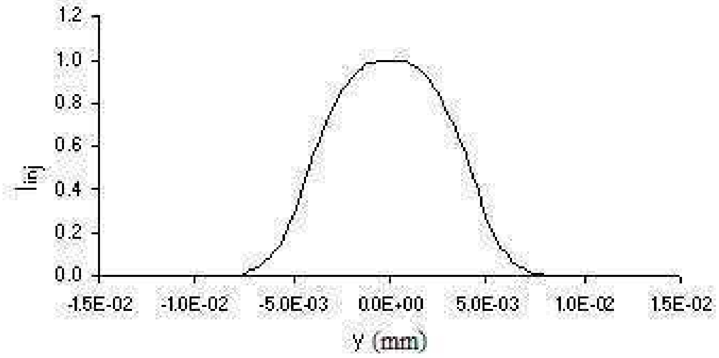


Figure 5.31: Normalised I_{inj} for beam position variation in vertical plane(y) at $x = 0$.

5.1.8 Results and discussion

Fig.5.30 shows the beam position variation measured near TL-1 beginning (at Beam Slit Monitor) during one hour operation. Fig.5.31 shows the simulated value of I_{inj} for beam movement along y. Table 5.6 lists the calculated limiting value (maximum value) of calculation time for successful tuning, and the actual calculation time values (observed on Windows Vista^{TM},Q9400 @2.66GHz). Fig.5.32 shows simulation result graphs for booster injection current (I_{inj}), tracked beam position and tracking error plotted for different BPM failure scenarios. During this simulation the beam at the TL-1 input is varied sinusoidally with an amplitude of $10mm$ along y. Fig.5.33 shows simulation results for the injection current (I_{inj}) for beam position variation along y between $-10mm$ to $+10mm$ under different actuator failure conditions. For all the simulations the lower limit of injection current in booster is taken as 95%(i.e. whenever the injection current

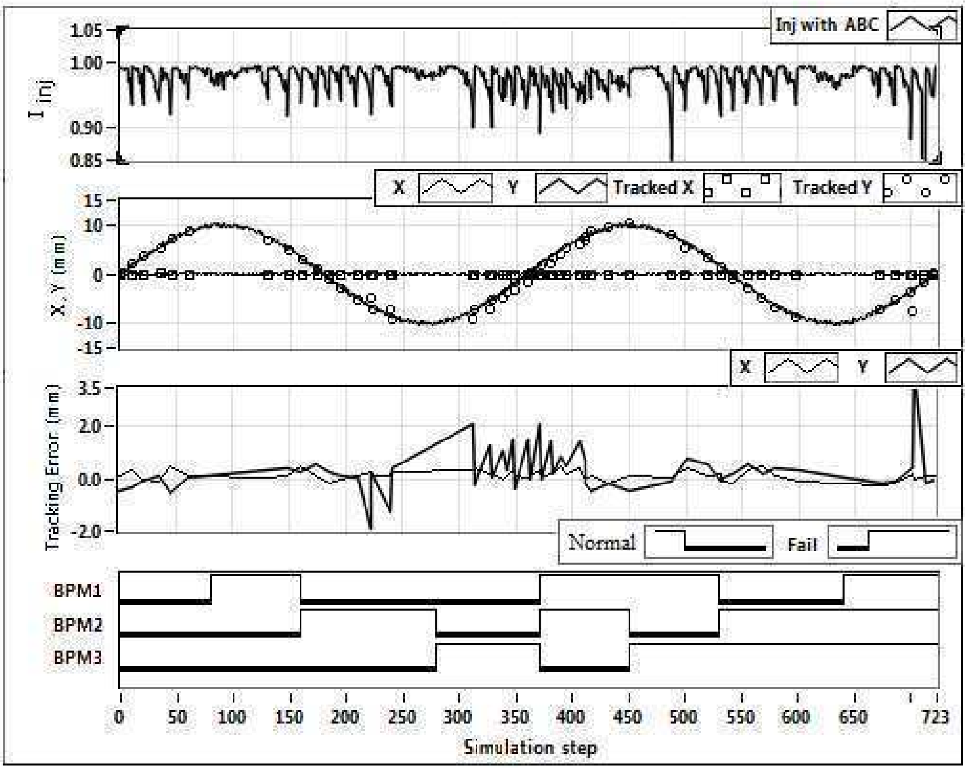


Figure 5.32: Normalised injection current, tracked beam position, and tracking error for different BPM failure conditions when beam position at TL-1 input along y is varied sinusoidally with amplitude of $10mm$.

in booster will fall below 95% the corrective action by the TL-1 agent will be initiated) and $I_{lim} = 98\%$ is used by the model based tracking plan discussed for case 3. From Fig.5.30 it can be seen that the beam position along x does not show larger variation from the operating point, whereas the variation along y is large and needs correction. From Fig.5.30 the value for $|\frac{d\widetilde{X}_{start,s}}{dt}|$ along y is calculated as $2.2\mu m/s$. From Fig.5.31 the value of $\hat{A}_{start}$ (for $I_{inj,lim} = 95\%$) along y, d_1, and $\hat{D}$ (for $I_{inj,th} = 10\%$) is calculated as 1.5mm, 3mm, and 6.1mm respectively. Then using these for the stated conditions the limiting value of $\triangle t_{cal}$, $\triangle t_{cal,1}$, $\triangle t$ are calculated for successful tuning of TL-1 as shown in Table 5.6. From Table 5.6 it can be concluded that since the measured value of calculation time is less than the limiting value of calculation time for all the cases discussed, the proposed TL tuning method can be used successfully for tuning of TL-1 under different sensor/actuator failure scenarios. Further it can be seen that limiting value of $\triangle t_{cal,1}$

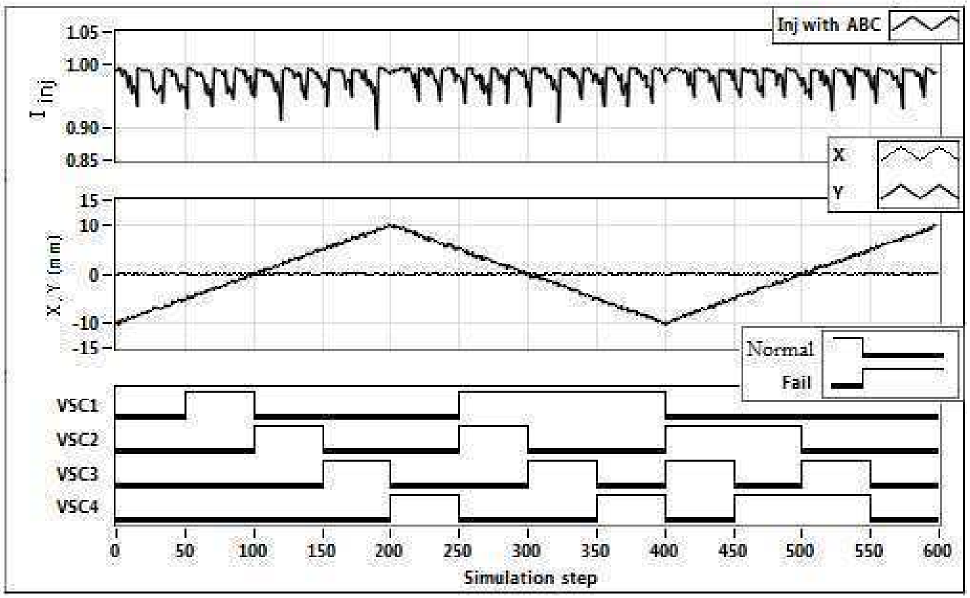


Figure 5.33: Normalised injection current under different steering coil (actuator) failure conditions (the beam at TL-1 input along y is varied between $-10mmto+10mm$ with $100\mu m$/s.)

needed for successful tracking is more than the limiting value $\triangle t$ for successful tuning, therefore it can be said that if the condition of successful tuning is met than the condition for successful tracking will also be satisfied. From Fig.5.32 and 5.33 it can be seen that using the proposed TL-1 tuning methods, the injection current can be successfully retained within the specified limit of 95% criteria under different sensor/actuator failure scenarios. Thus with the addition of new plans based on model based tracking concepts to the TL-1 intelligent agent. The agent based control implementation specifically at Indus-1 complex

Table 5.6: Limiting value of calculation time.

Scenario	limiting value	Actual value
Case 1	$\triangle t_{cal}$=500s	$\approx$ 13s
Case 2	$\triangle t_{cal}$=409s	$\approx$ 40s
Case 3	$\triangle t$=500s	$\approx$ 81s
	$\triangle t_{cal,1}$=1409s	$\approx$ 43s

$When\ \mathring{e}_{DE-I}, e_{noise}, \mathring{e}_{model}, \mathring{e}_{BPM} = 200\mu m$

will not only improve the beam quality by continuously maintaining the high level of injection current in booster but will also be able to do this even in the case of limited sensor/actuator failure cases thus improving the overall system availability. Which is an important parameter for machines like Indus to fulfil the increasing demand of synchrotron beam usages posed by beamline users. And addition of ability to run the machine even in the unlikely event of limited device failure, till the next proposed shutdown is highly desirable. As any unexpected early machine shutdown will adversely affect all the beam line users who has already been allotted with the beam time for use during this period.

5.1.9 Summary and conclusion

In accelerator the transport lines are used for transporting the electron / charged particle beams from source to destination. During transportation they modify the beam characteristics to match with the beam acceptance characteristics of destination . Various AI based methods proposed by many researchers in past have successfully improved the system availability by reducing the tuning time. In this section the method for controlling transport lines, based on intelligent agent concept is presented. The TL-1 intelligent agent is formulated with a model based goal based modular architecture. The goals and plans for TL-1 agent are formulated. The framework for the formulated agent is developed and the TL-1 intelligent agent is implemented with some of the formulated goals. Simulation program is developed and the overall systems characteristics in open loop as well as in closed loop with agent based control is captured. The method for controlling transport lines, based on model-based tracking concept, for agent-based control systems is presented, and the condition for its successful operation is derived. The proposed method increases the overall system availability under different sensor/actuator failure conditions by tuning of TL if beam position at source varies. The ability to work under total loss of BPM data can be further exploited to optimize injection current for transport lines having only interceptive type of BPM (for example fluorescent screen type BPM). The simulation results of the proposed scheme on the INDUS-1 accelerator subsystems models showed their effectiveness towards the automatic tuning of transport line under different sensor/actuator failure scenarios.

5.2 Microtron Intelligent Agent

A Microtron is a circular electron accelerator in which the kinetic energy of electrons is increased by a constant amount per field change (one half or a whole revolution). They are designed to operate at constant field frequency and magnetic field strength.

In a Microtron due to their different relativistic mass the electrons in different passes of the acceleration travel on different paths through the bending magnet. Thus the time needed for the electron is proportional to the turn number. The slow electrons need one electric field oscillation, the faster electrons an integer multiple of this oscillation.[47, 127]

To develop a model that represent variation of electron beam current and electron beam position dependence on different Microtron control and read-back parameters it was decided to find the inter dependence of different Microtron parameters experimentally. Towards this aim, experiments has been carried out and the data for extracting the dependence of FCT signal, emission signal and beam position on magnetic field, cathode current and RF frequency has been collected. Using these data the model of the Microtron is calculated for both static and dynamic behaviour.

5.2.1 Requirements of the Microtron intelligent object

The proposed research work needs the synthesis of multiple intelligent objects for efficient autonomous controlling of the accelerator machine with an emphasis on reducing the manual (operator) interventions primarily to the supervisory role. Microtron intelligent object (Microtron agent) is one of these intelligent objects to be synthesised and thus the requirements of the identified Microtron model will depend on the Microtron intelligent object to be synthesised. Table 5.7 list the requirements of Microtron intelligent object.

Considering the requirements of Microtron intelligent object listed in Table 5.7 and consulting the Microtron experts & shift operators for the Microtron parameters normally used by them for Microtron tuning, the system input are divided into two groups. First group lists the parameters which should not be changed/disturbed by the intelligent agent during its operation. Second group lists the parameters which can be changed/optimised (within the specified minimum and maximum limits) by the intelligent object during its operation. Table 5.8 lists all the parameters available in Microtron control system and figure 5.34 shows the corresponding black box representation of the system which is used in the calculation process. In the next section we will be presenting the Microtron model identification experiments and the data collected. Then in the next section the analysis performed on the collected data and the identified system model will be discussed.

Table 5.7: Requirements of Microtron intelligent object.

Sr. No.	Requirements
1	It should be able to automatically control the Microtron machine, to maintain it, to continuously operate at the given operating point
2	It should be able to successfully steer the machine from one operating point to another operating point by controlling the available machine control parameters
3	It should be able to predict maximum efficiency operating point for the machine
4	It should be able to predict maximum FCT current possible at the desired operating point for the machine
5	It should be able to predict FCT current possible at the desired beam position for the machine
6	It should be able to predict nearest optimum operating point of the machine for given beam position

Table 5.8: Different parameters available in Microtron control system.

Sr. No.	Parameter	Type	Uses	Comments
1	Cathode Current	Input	Allowed	Will be used by intelligent object
2	RF frequency	Input	Allowed	Will be used by intelligent object
3	Dipole Current	Input	Not allowed	
4	RHC Current	Input	Not allowed	
5	LHC Current	Input	Not allowed	
6	Mid plane Current	Input	Not allowed	
7	RF level	Input	Not allowed	
8	FCT	Output	Allowed	Available as CRO trace needs preprocessing
9	Emission	Output	Allowed	Available as CRO trace needs preprocessing
10	Reflected power	Output	Allowed	Available as CRO trace needs preprocessing
11	Forward power	Output	Allowed	Available as CRO trace needs preprocessing

5.2.2 Microtron Model identification

To identify the dependence of beam position over cathode current, the cathode current is varied in steps of $100mA$ while keeping all other parameters fixed. After providing the settling time of 5 minutes the beam image at beam slit monitor is captured. From this

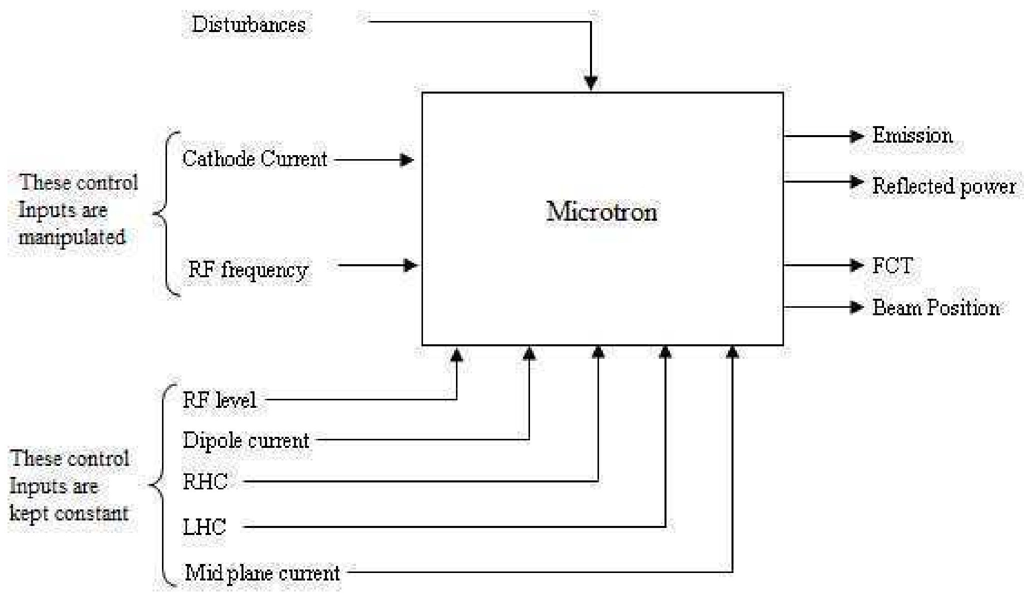


Figure 5.34: Black box model of the system.

image the beam position in horizontal and vertical plane are calculated as following.

1. Subtract the background image from the captured image.
2. Extract out the region of interest(ROI) from this processed image (subset of the processed image) with i number of pixels horizontally and j number of pixels vertically.
3. For ROI the beam centroid is calculated as:

$$X = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} i \times I_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{m} I_{i,j}} \tag{5.16}$$

$$Y = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} j \times I_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{m} I_{i,j}} \tag{5.17}$$

   where $I_{i,j}$ is the intensity of the pixel(i,j) in ROI and calculated X and Y are in pixel units.

4. From the calculated centroid positions X and Y the beam positions x and y in units of mm are calculated using the calibration data as:

$$x(mm) = x_0 - aX \tag{5.18}$$

$$y(mm) = y_0 + bY \tag{5.19}$$

   where a and b are the calibration constants for conversion of pixel to mm in horizontal direction and vertical direction respectively. Calibration constants x_0 and y_0 are for the reference point of centre coordinate with respect to the ROI.

Figure 5.35 shows the measured beam position variation with respect to cathode current setting.

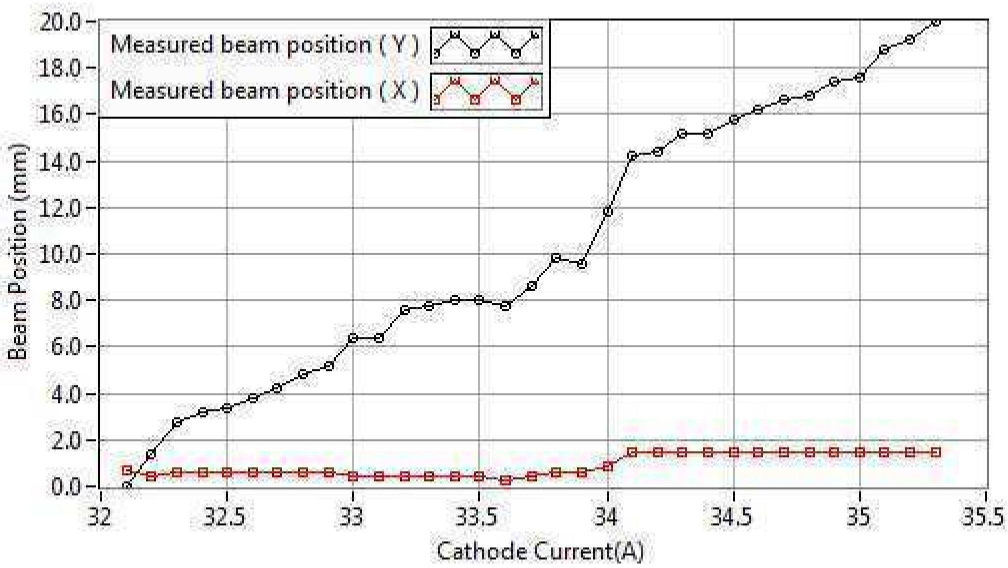


Figure 5.35: Beam position variation with respect to cathode current.

To identify the dependence of beam position over RF frequency, the RF frequency is varied in steps of $100KHz$ while keeping all other parameters fixed. After providing the settling time of 5 minutes the beam image at beam slit monitor is captured. From this image the beam position is calculated using Eq.5.17 to 5.19. Figure 5.36 shows the measured beam position variation with respect to RF frequency set values.

To determine the dependence of emission and FCT over cathode current, the cathode current is varied in steps of $100mA$ while keeping all other parameters fixed. After providing the settling time of 5 minutes the CRO traces of FCT and emission current are captured. Microtron operates in pulsed mode and thus the FCT and emission signals are observed in the form of CRO trace representing the pulse. To extract the emission and FCT signal values from these CRO traces, first from each CRO trace the respective background trace is subtracted and then the pulse hight is measured by fitting the best pulse to the observed CRO traces, this process is shown in figure 5.37. Figure 5.38 shows the observed variation of FCT and emission current with respect to cathode current.

For finding the dependants of FCT, emission current and RF reflected power over RF frequency, the RF frequency is varied in steps of $100KHz$ while keeping all other parameters fixed. After waiting for 5 minutes of stabilisation time the FCT, emission current and reflected power signals are measured from CRO traces. Figure 5.39 shows the variation of FCT, emission and reflected power with respect to the RF frequency.

In the absence of first principle models of each component of Microtron that can be interconnected and parametrised to get the systems dynamic characteristics. The black-box identification method is used. For identifying the dynamic characteristics of emission current and RF reflected power over cathode current, Microtron is first tuned at the best operating point and then the cathode current is varied to get step response of emission and RF reflected power. In the preliminary experiment stage it is observed that the emission and reflected power shows different transition rates for the cases, a: when cathode current is increased and b: when cathode current is decreased. Therefore the experiment is conducted to identify the transfer function for both the cases. For identifying emission 170 step response data sets are collected for each case. Out of 170 step

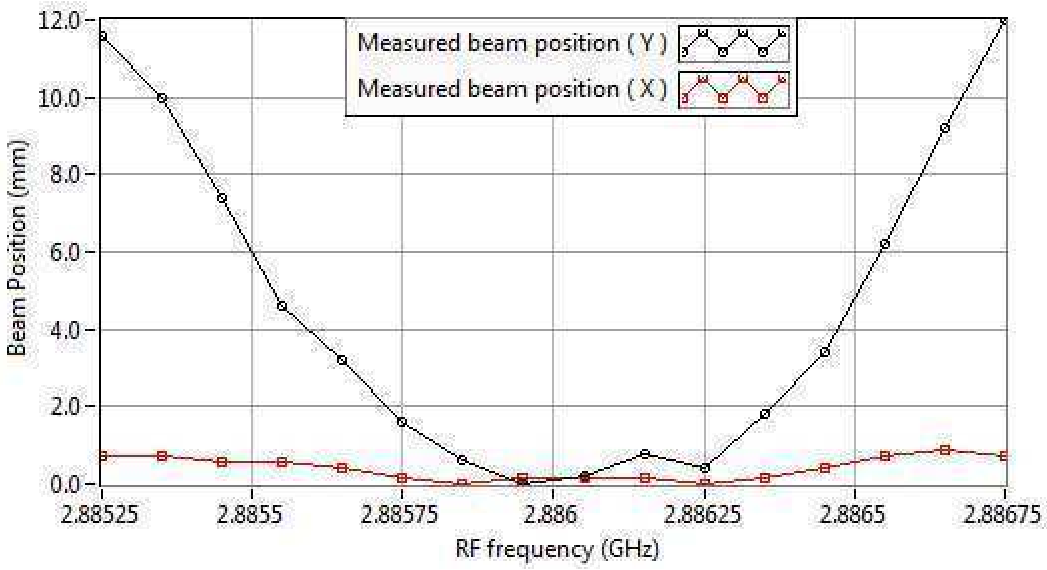


Figure 5.36: Beam position variation with respect to RF frequency.

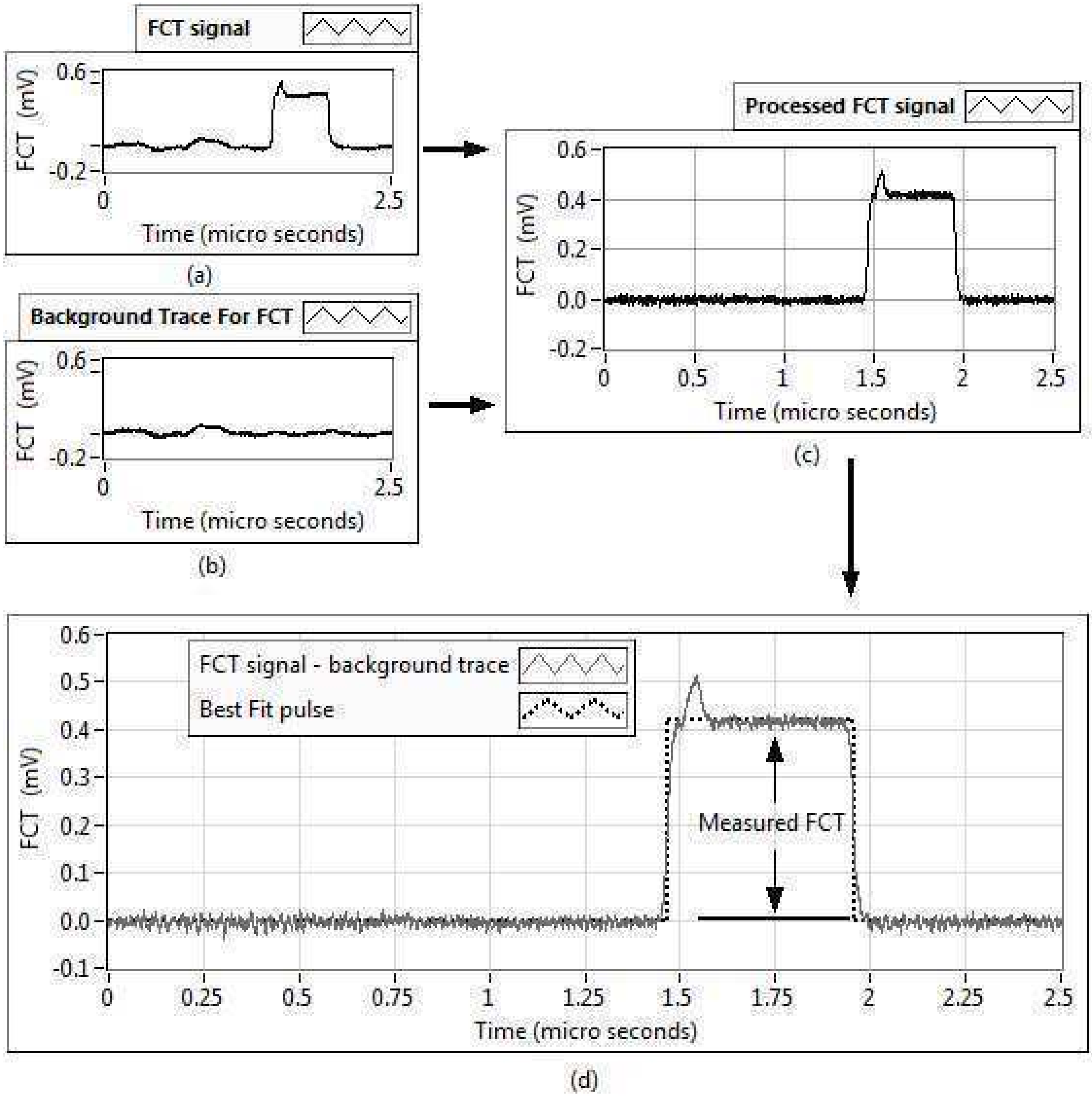


Figure 5.37: Process of calculating FCT signal from CRO trace.

response data sets, 100 data sets are selected after examining for outliers and distortion of step data due to change in parameter other than cathode current. These are then divided into two groups for system identification and validation. Figure 5.40 shows the emission identification data sets along with the step curve obtained by taking the average of these 50 data sets for both the cases, a)when cathode current is increased and (b) when the cathode current is decreased. Similarly for the RF reflected power best 60 data sets are grouped into identification data set and validation data set. Figure 5.41 shows the RF

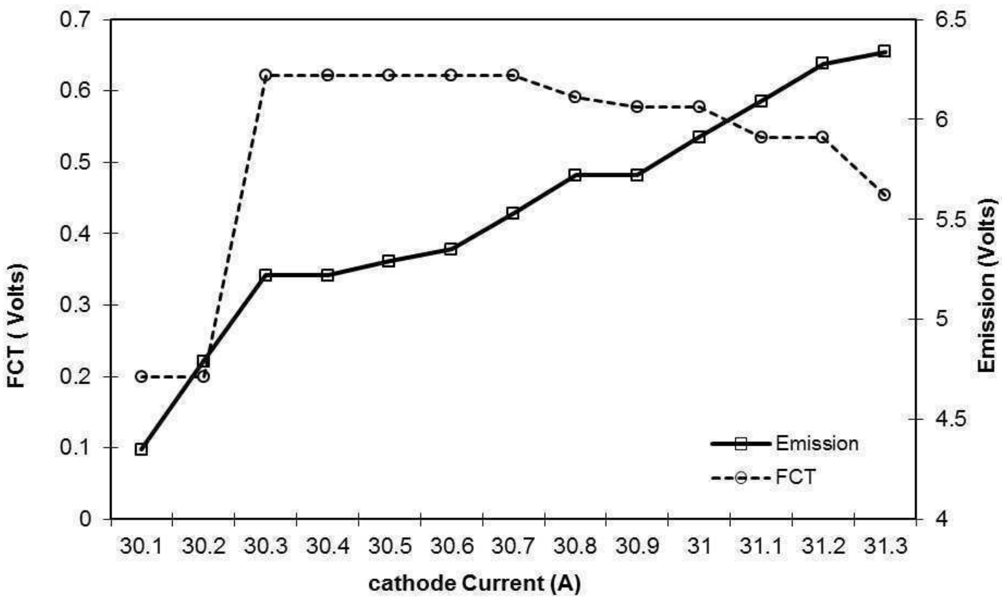


Figure 5.38: Variation of FCT and emission with respect to cathode current.

reflected power identification data sets along with the step curve obtained by taking the average of these 30 data sets for both the cases, a)when cathode current is increased and (b) when the cathode current is decreased.

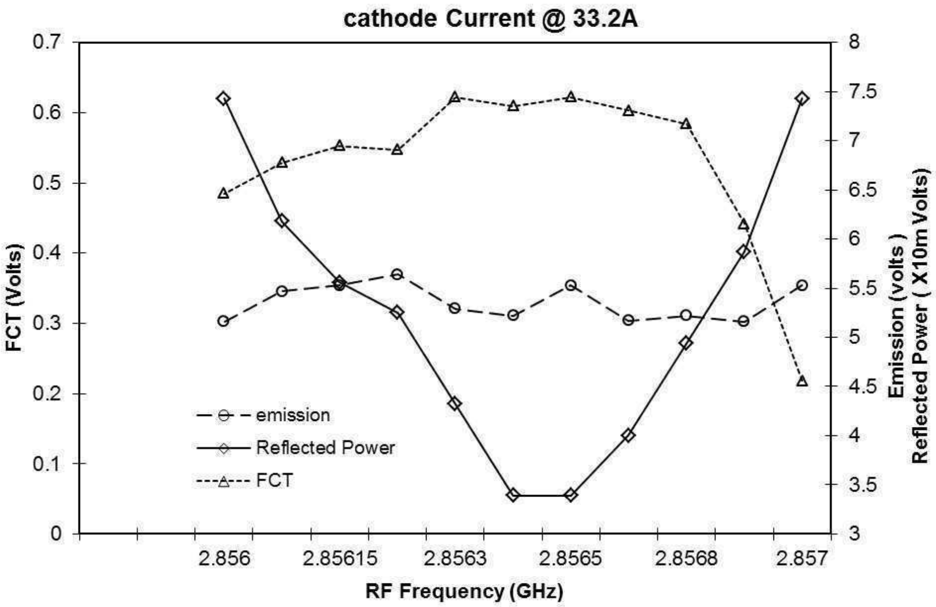
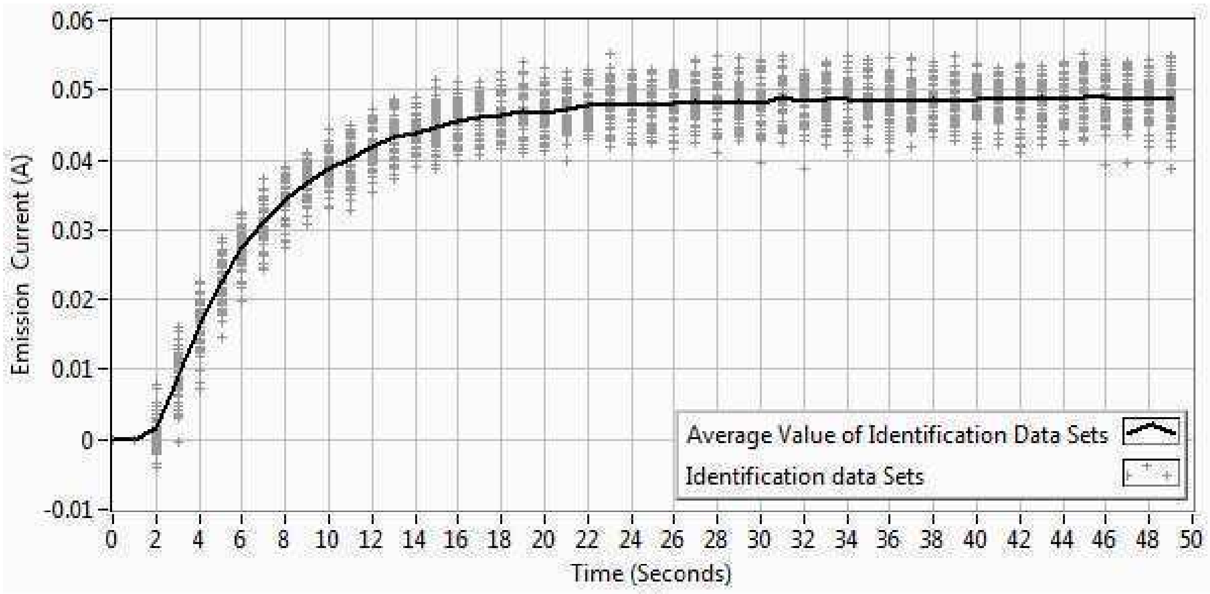


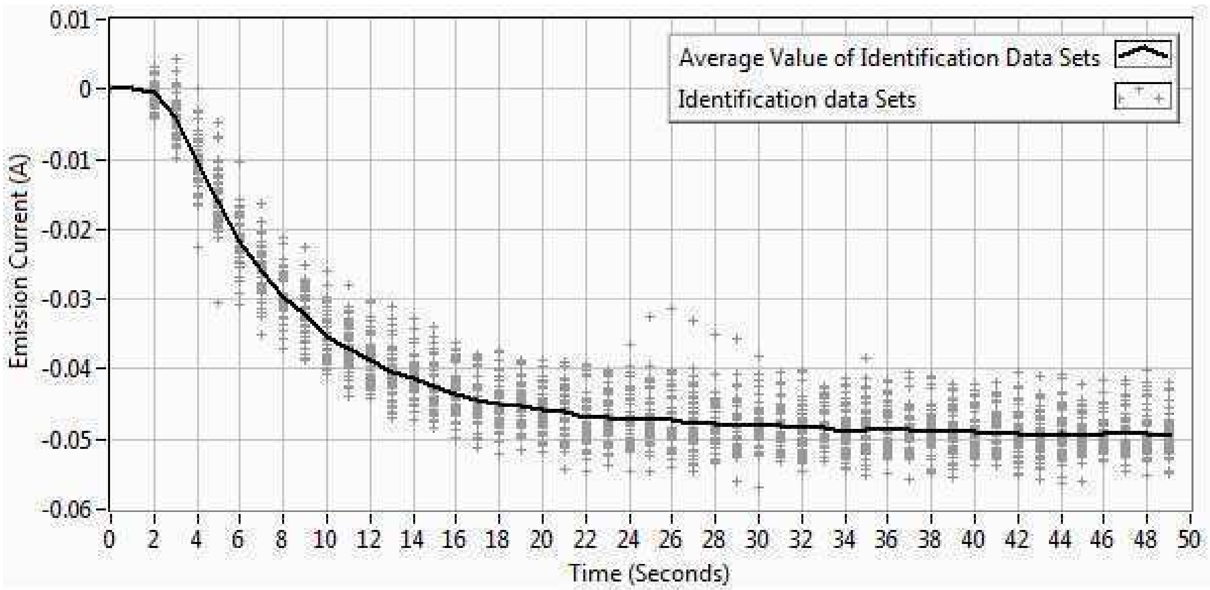
Figure 5.39: Variation of FCT, emission and reflected power signal with respect to the RF frequency.

Figure 5.42 shows the variation of cathode current(value for normal operation) and emission with respect to the cathode run time (i.e. the time from last cathode replacement for which the cathode is in operated)

To observe the effect of change in cathode current on cavity resonant frequency and the reflected power the sweep is generated in RF frequency at three different cathode



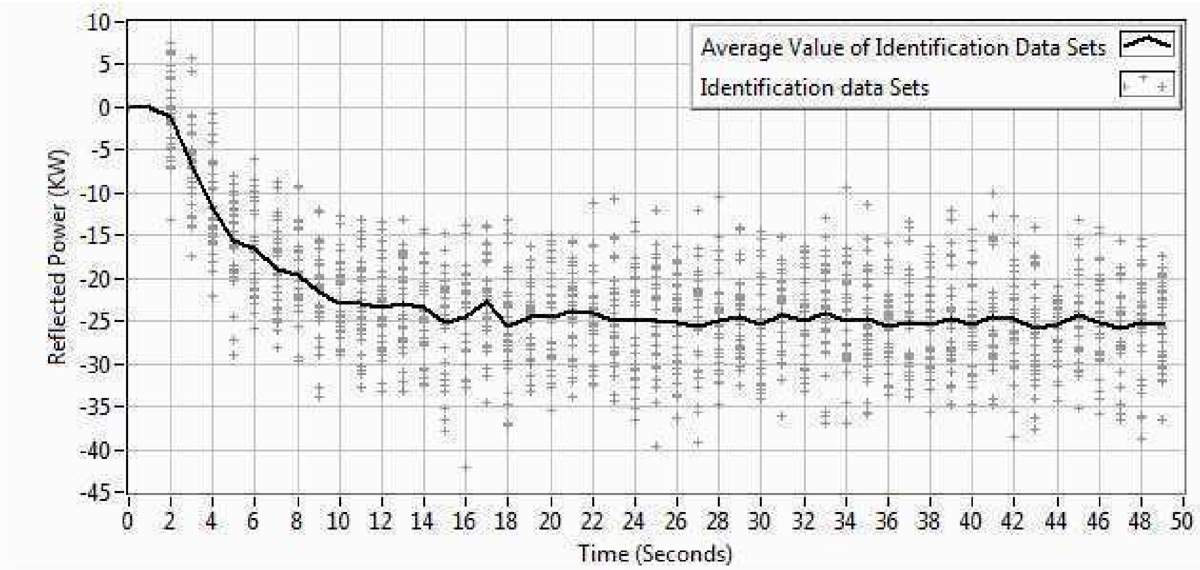
(a)



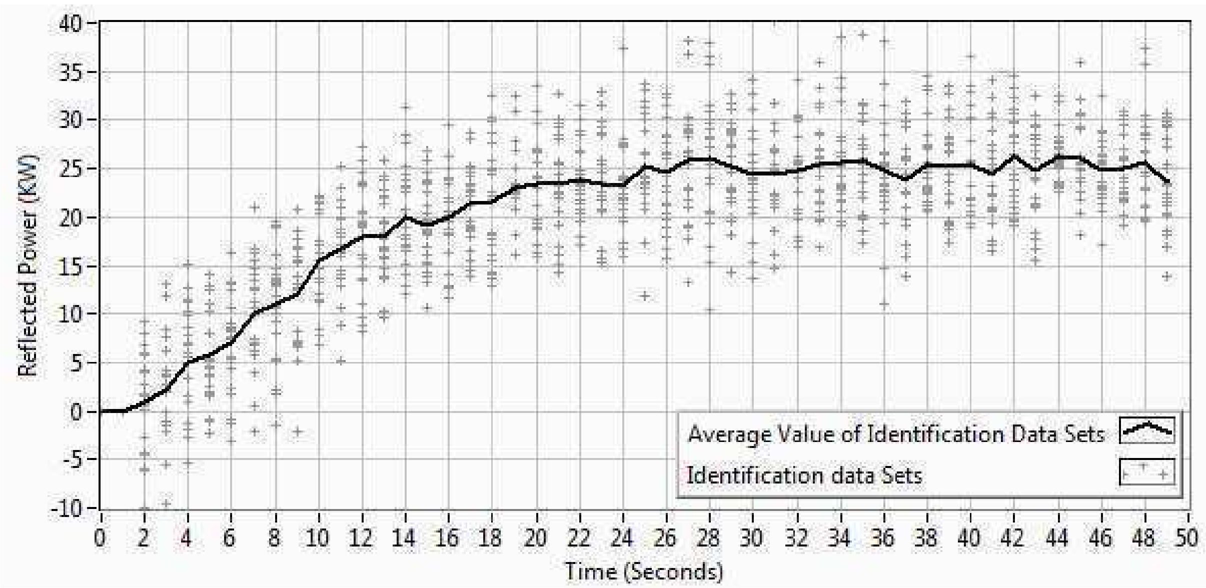
(b)

Figure 5.40: Cathode current step response for identifying the TF of emission, (a) when cathode current is increased from normal operating value, (b) when cathode current is decreased from normal operating value.

current values and the result of this is shown in Figure 5.43.



(a)



(b)

Figure 5.41: Cathode current step response for identifying the TF of reflected power, (a) when cathode current is increased from normal operating value, (b) when cathode current is decreased from normal operating value.

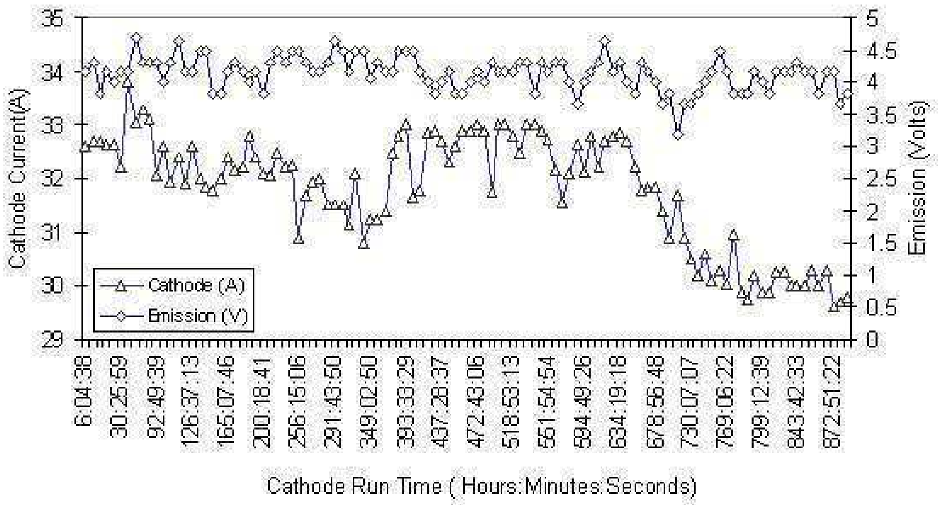


Figure 5.42: Variation of cathode current and emission with cathode runtime.

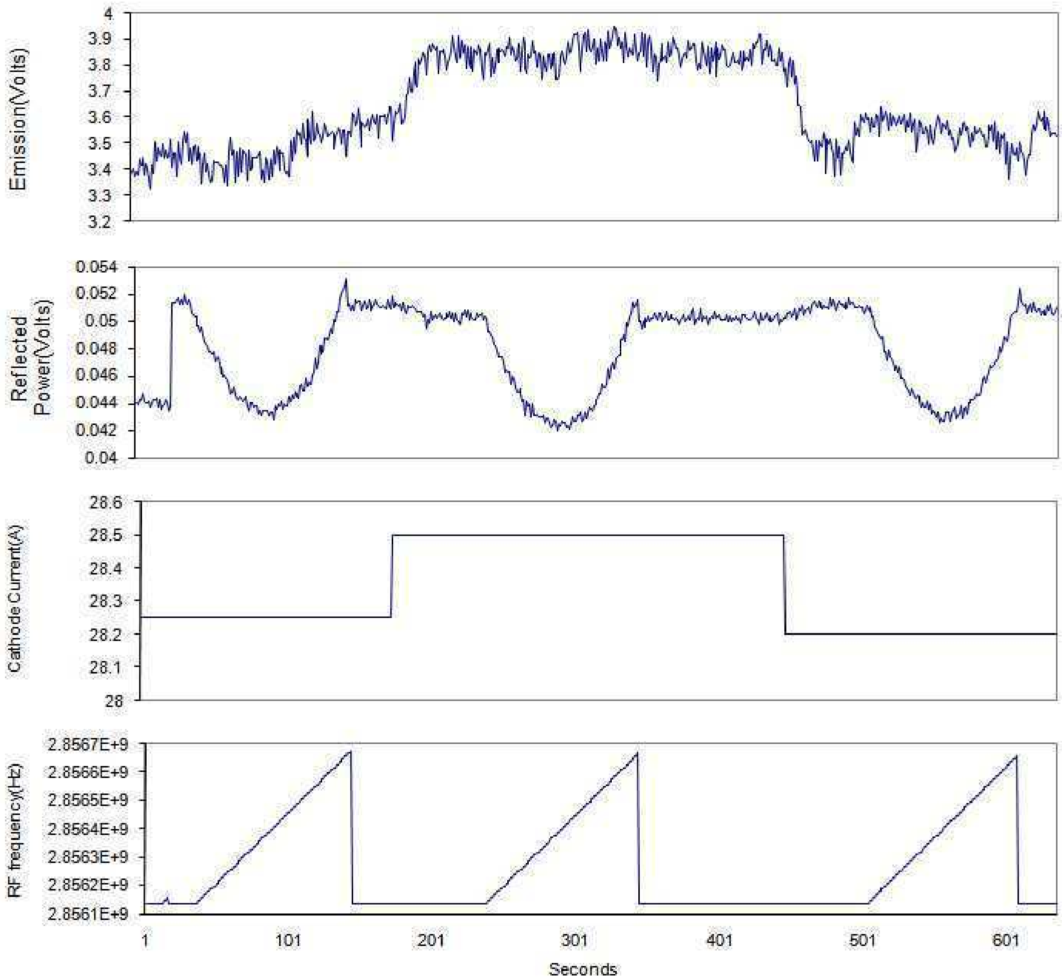


Figure 5.43: Data for observing effect of cathode current on cavity resonant frequency and reflected power.

5.2.3 Discussion

Static response

From figure 5.35 and 5.36 it can be observed that the sensitivity of beam position in vertical plane is about 10 to 20 times larger than that in the horizontal plane. further the over all variation of beam position in horizontal plane is observed to be limited within $1mm$.

Further it can be seen that the vertical beam position shows strong dependence on both the parameters namely cathode current and Rf frequency.

From figure 5.36 it can be seen that the position does not directly depend on the RF frequency but it depends on another parameter, we will name it as RF frequency detuning (δf) i.e. how far is the operating RF frequency from the cavity resonant frequency. As the operating RF frequency moves away from the cavity resonant frequency the beam starts moving in the +ve direction.

From figure 5.35 the linear model of beam position with respect to cathode current

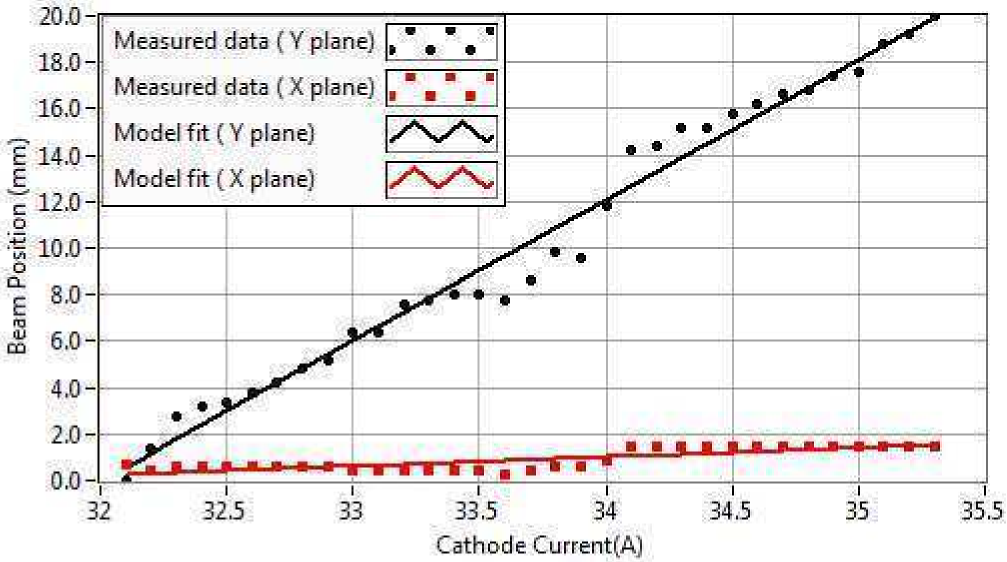


Figure 5.44: Goodness of fit for the beam position variation observed in horizontal plane and vertical plane with respect to the cathode current.

is computed as:

$$X = 0.3887 I_{cathode} - 12.22 \qquad \text{When X is in (mm) and } I_{cathode} \text{ is in (A)} \tag{5.20}$$

$$Y = 6.053 I_{cathode} - 193.8 \qquad \text{When Y is in (mm) and } I_{cathode} \text{ is in (A)} \tag{5.21}$$

using the linear least square fitting method (Appendix A.1). Figure 5.44 shows the goodness of fit for this computed model with the observed data.

Using trust-region-reflective non-linear least squares curve fitting method(Appendix A.3) and the fitting function given by Eq5.22 as function options.

$$y = a(1 - e^{-\frac{(x)^2}{b}}) \tag{5.22}$$

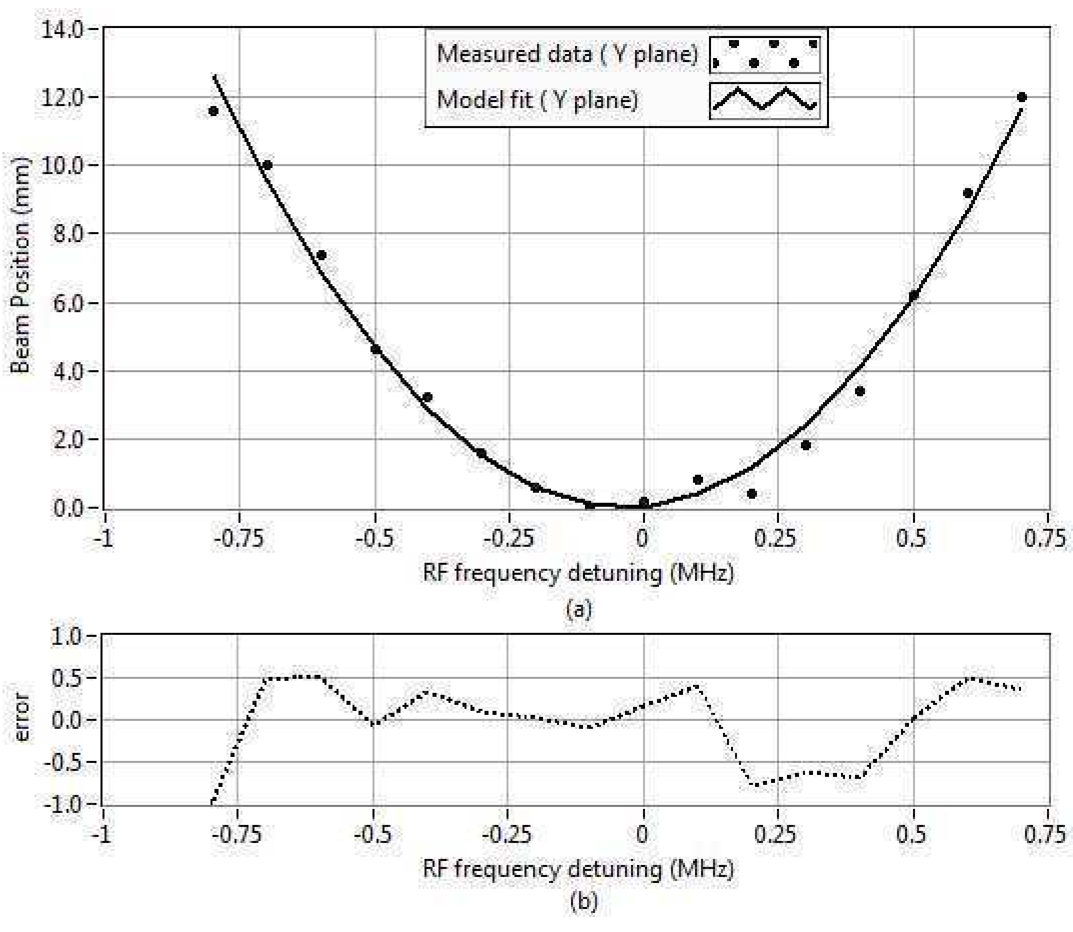


Figure 5.45: Goodness of fit for the beam position variation observed in vertical plane with respect to the RF detuning.

The relationship between beam position variation in vertical plane and RF detuning is computed using data from figure 5.36 as:

$$Y = 1300(1 - e^{-\frac{(\delta f)^2}{60.16}}) \qquad \text{When Y is in (mm) and } \delta f \text{ is in (MHz)} \tag{5.23}$$

Figure 5.45 shows the goodness of fit for this computed model. The R-square value for the fitted model is calculated as 0.98 which indicates a good model fit over observed data.

Similarly the relationship between beam position variation in horizontal plane and RF detuning is computed using data from figure 5.36 as:

$$X = 0.92(1 - e^{-\frac{(\delta f)^2}{0.25}}) \qquad \text{When X is in (mm) and } \delta f \text{ is in (MHz)} \tag{5.24}$$

Figure 5.46 shows the goodness of fit for this computed model. The R-square value for the

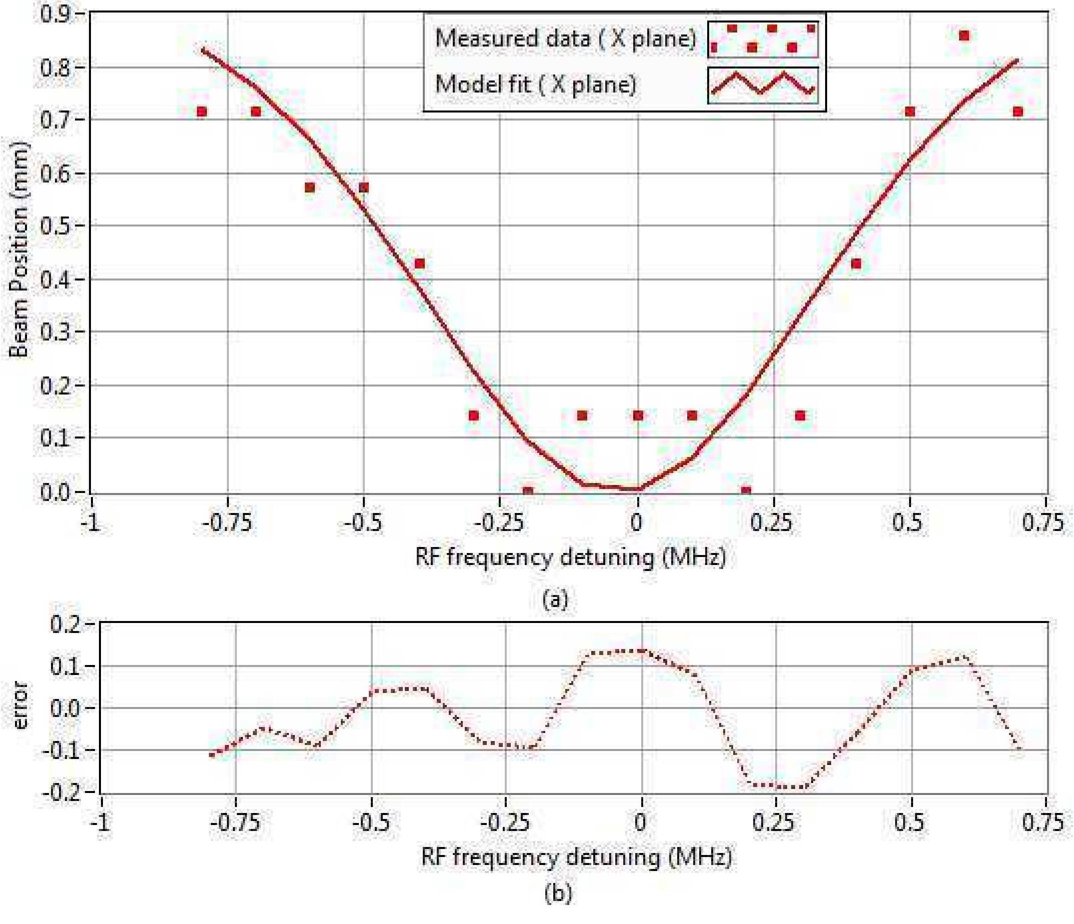


Figure 5.46: Goodness of fit for the beam position variation observed in horizontal plane with respect to the RF detuning.

fitted model is calculated as 0.85 that indicates an average model fit over observed data. Further one can see that the error between the fitted model and the observed values for the case of horizontal plane shown in the figure 5.46(b) shows an oscillation pattern which is not observed in case of vertical plane (figure 5.45(b)). This indicates the presence of some unknown phenomenon (that needs to be investigated separately and is not the part of this thesis work) that is overriding the horizontal plane beam position variations. From figure 5.38 it can be seen that with increase in the cathode current, the emission signal increases, but the rate of increase of emission signal experiences a sudden change. This point of sudden change in the rate of change of emission signal shows a sharp dependence on the FCT signal and is at the same point where the FCT shows a sudden increase (i.e. it experiences a sudden +ve slope), let us call this point as acceleration start point (ASP)(i.e. the point at which the electron acceleration process in Microtron starts). On

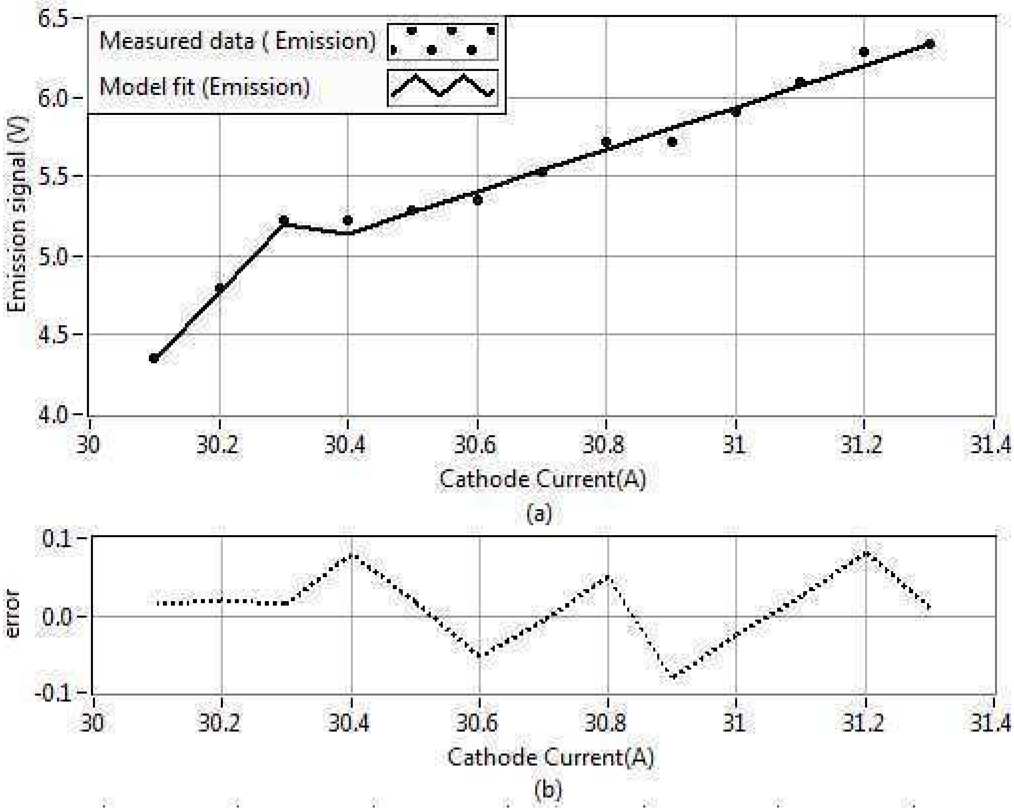


Figure 5.47: Goodness of fit for the Emission signal with respect to the cathode current.

further increase in the cathode current the FCT signal decreases (i.e. it experiences a -ve slope)

The emission signal data from figure 5.38 is divided into two groups based on the ASP ($I_{cathode}^{ASP} = 30.3$) such that for group 1 $I_{cathode} < ASP$ and for group 2 $I_{cathode} \geq ASP$. Now for each group, using the linear least square fitting method (Appendix A.1) the linear fit is calculated. Using these individual fitting equations the emission dependence on cathode current is modelled as:

$$E = \begin{cases} 4.350 I_{cathode} - 126.60 & \text{for } I_{cathode} \leq ASP \\ 1.324 I_{cathode} - 35.11 & \text{for } I_{cathode} > ASP \end{cases} \quad (5.25)$$

Figure 5.47 shows the goodness of fit for this computed model. The R-square value for the fitted model is calculated as 0.99 that indicates a good model fit over observed data. Similarly from the data of figure 5.38 the dependence of FCT signal is computed using general polynomial least square fitting method (Appendix A.2) and polynomial order = 3, as function options. The model is computed as:

$$FCT = \begin{cases} 0.2 & \text{for } I_{cathode} < ASP \\ -0.2511 I_{cathode}^2 + 15.33 I_{cathode} - 233.4 & \text{for } I_{cathode} \geq ASP \end{cases} \quad (5.26)$$

Figure 5.48 shows the goodness of fit for this computed model and the R-square value of 0.99 for the fitted model indicates a good model fit over observed data.

From figure 5.39 it can be seen that the FCT is maximum at the point where the reflected power is minimum (i.e. $\delta f = 0.0$). As we move away from the cavity resonant frequency the FCT decreases. The rate of decrease in FCT signal is less for $-\delta f$ as compared to $+\delta f$. Further to this a flat top region is observed in FCT for a band of about $0.40 MHz$ centred at cavity resonant frequency. Using data from Figure 5.39, the dependence of FCT with respect to the RF detuning (δf) is computed with general polynomial least square fitting method. With polynomial order = 5, the model is calculated as:

$$FCT = -5.007(\delta f)^5 - 2.889(\delta f)^4 + 0.034(\delta f)^3 - 0.375(\delta f)^2 + 0.034(\delta f) + 0.624 \quad (5.27)$$

$$Note: FCT \text{ is in } Volts \text{ and } (\delta f) \text{ is in } MHz$$

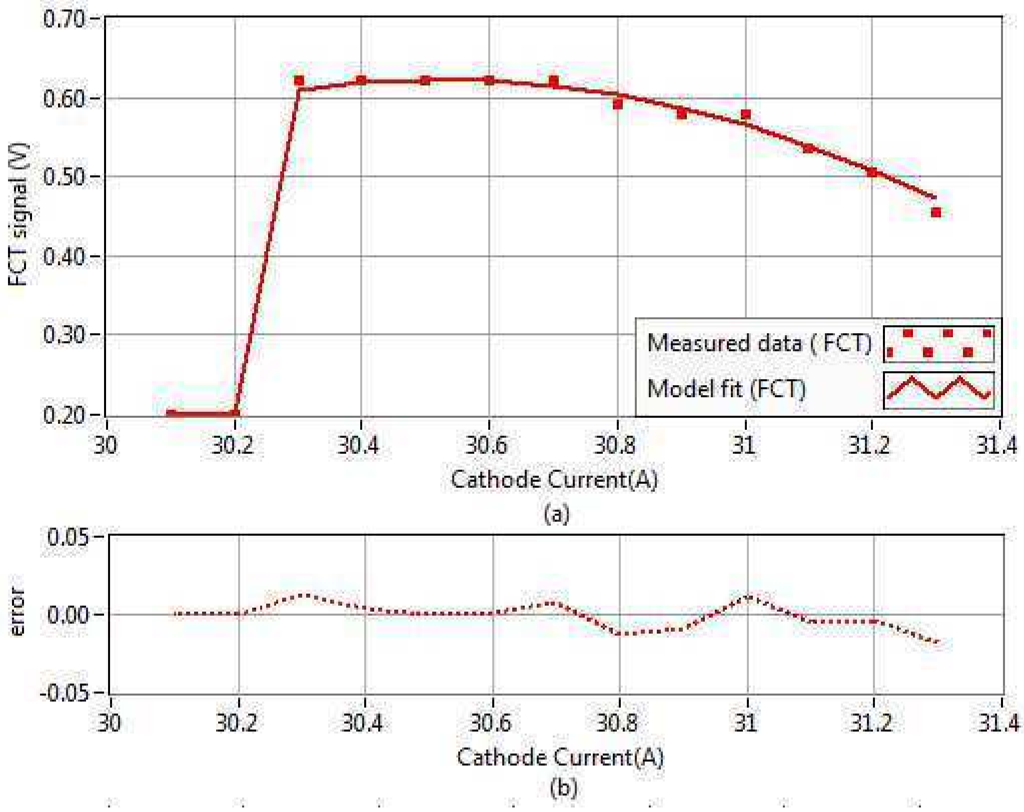


Figure 5.48: Goodness of fit for the FCT signal with respect to the cathode current.

for this computed model the goodness of fit is shown in figure 5.49. The R-square value of 0.98 for this fitted model indicates a good model fit over observed data.

From figure 5.39 it can be seen that the RF reflected power is minimum at the point where the $\delta f = 0.0$ and as we move away from the cavity resonant frequency, the reflected power increases. The minimum point of reflected power can be explored further to identify the cavity resonant frequency at any instance of time. Using data from Figure 5.39 the dependence of reflected power with respect to the RF detuning (δf) is computed using trust-region-reflective non-linear least squares curve fitting method(Appendix A.3) and the fitting function given by Eq5.22. The computed model is given by:

$$RP = 0.115(1 - e^{-\frac{(\delta f)^2}{0.576}}) + C_0 \text{ When } RP \text{ is in } (V), \text{ and } \delta f \text{ is in} (MHz) \quad (5.28)$$

$@I_{cathode} = 33.2A\ ,\ C_0 = 0.033$

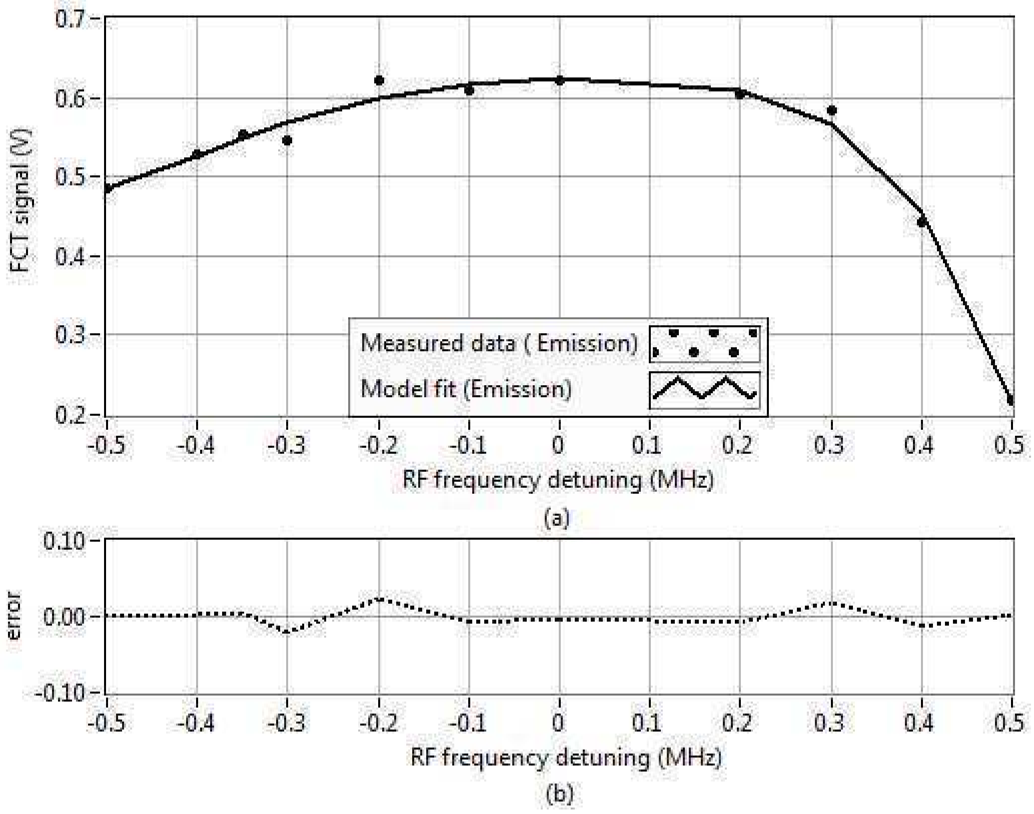


Figure 5.49: Goodness of fit for the FCT signal with respect to the RF detuning.

For this model the goodness of fit shown in figure 5.50 and the calculated value of 0.98 for R-square indicates a good model fit over observed data.

Further from Figure 5.43 it is observed that the change on the cathode current only changes the C_0 and thus shifts the curve up or down, without changing the width and amplitude of the curve. Moreover it seems that the value of C_0, also dependent on some other unidentified factors.

Dynamic response

From Cathode step response data shown in figure 5.40 and 5.41, it is observed that a change on the cathode current, introduces a change in the emission as well as RF reflected power. Using system identification method, first order plus dead time (FOPDT) type transfer function of emission is calculated for both the cases, a: when cathode current

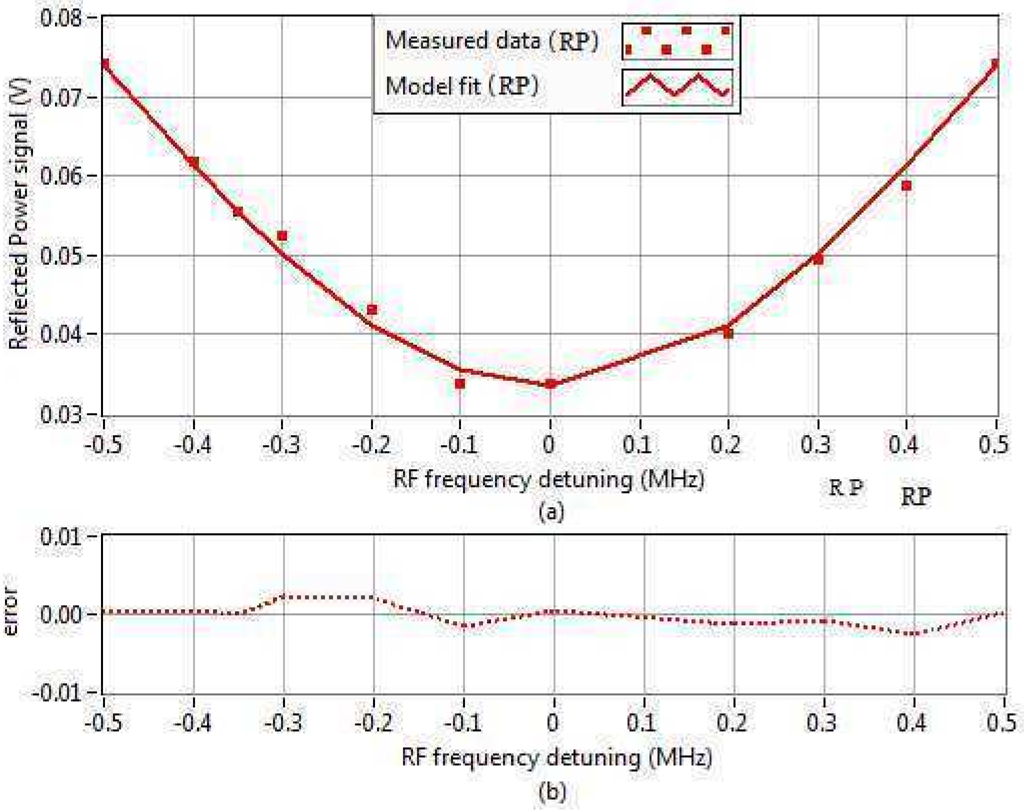


Figure 5.50: Goodness of fit for the reflected power signal with respect to the RF detuning.

is increased and b: when cathode current is decreased using the average step response curve of identification data shown in the figure 5.40 as:

$$TF_E = e^{-1s}\frac{0.098}{(1+5.0s)} \; for\ case\ (a)\ when\ cathode\ current\ increases \tag{5.29}$$

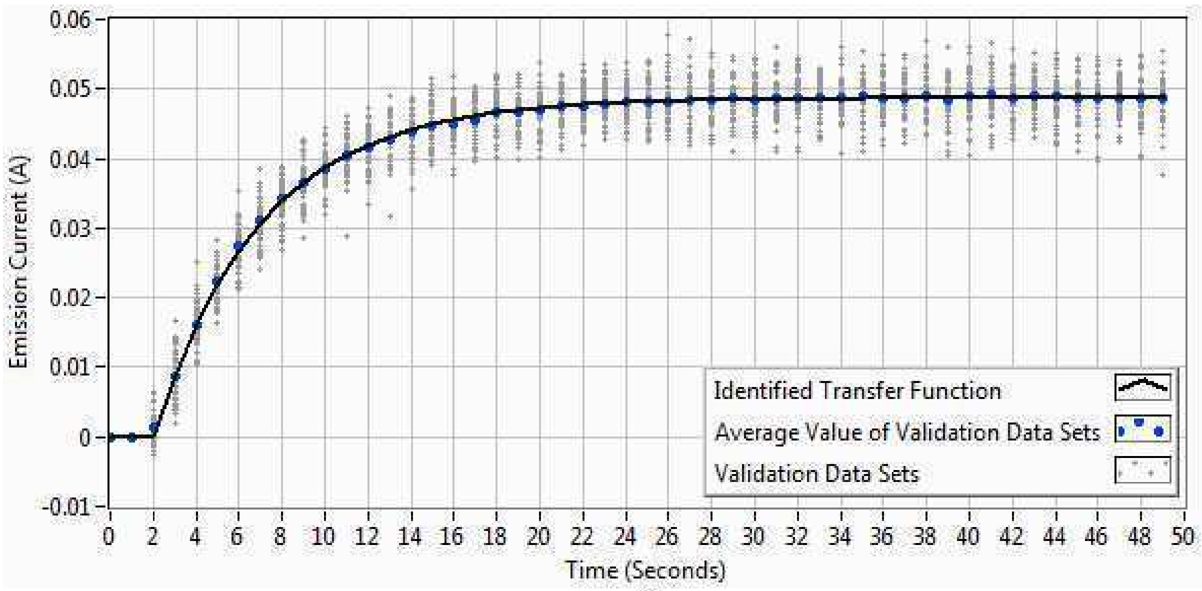
$$TF_E = e^{-1s}\frac{0.098}{(1+6.0s)} \; for\ case\ (b)\ when\ cathode\ current\ decreases \tag{5.30}$$

Figure 5.51 shows goodness of fit for these calculated transfer functions with the validation data. Similarly for the reflected power the TF are calculated using the average step response curve of identification data shown in the figure 5.41 as:

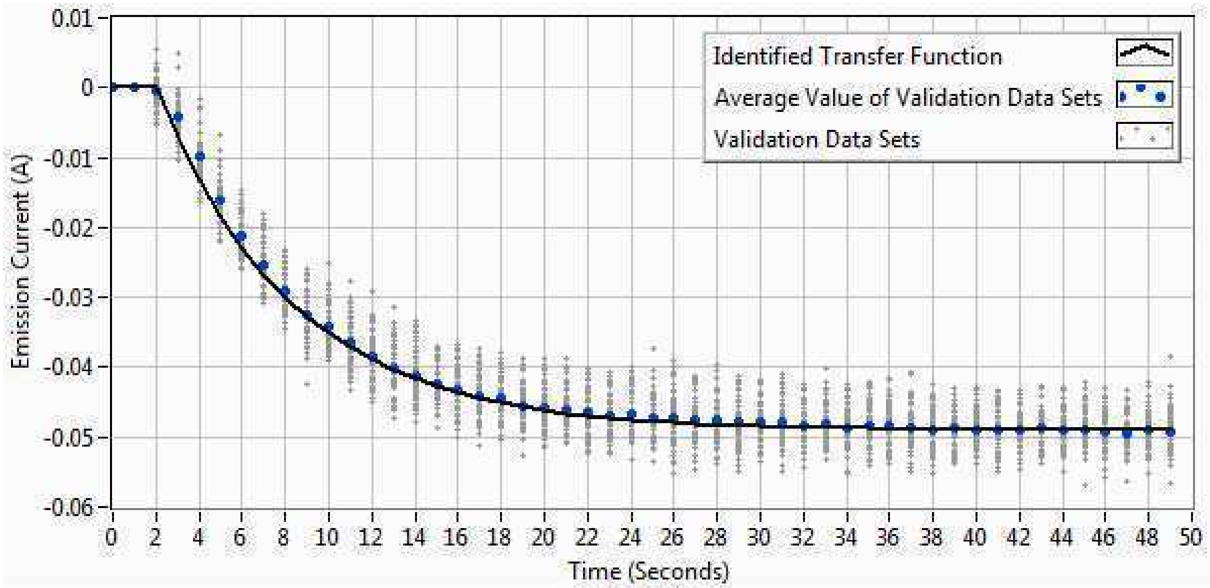
$$TF_{RP} = e^{-1s}\frac{-50}{(1+3.0s)} \; for\ case\ (a)\ when\ cathode\ current\ increases \tag{5.31}$$

$$TF_{RP} = e^{-1s}\frac{-50}{(1+8.0s)} \; for\ case\ (b)\ when\ cathode\ current\ decreases \tag{5.32}$$

Figure 5.52 shows goodness of fit for these calculated transfer functions with the validation data.



(a)



(b)

Figure 5.51: Validation of TF for emission, (a) when cathode current is increased from normal operating value, (b) when cathode current is decreased from normal operating value.

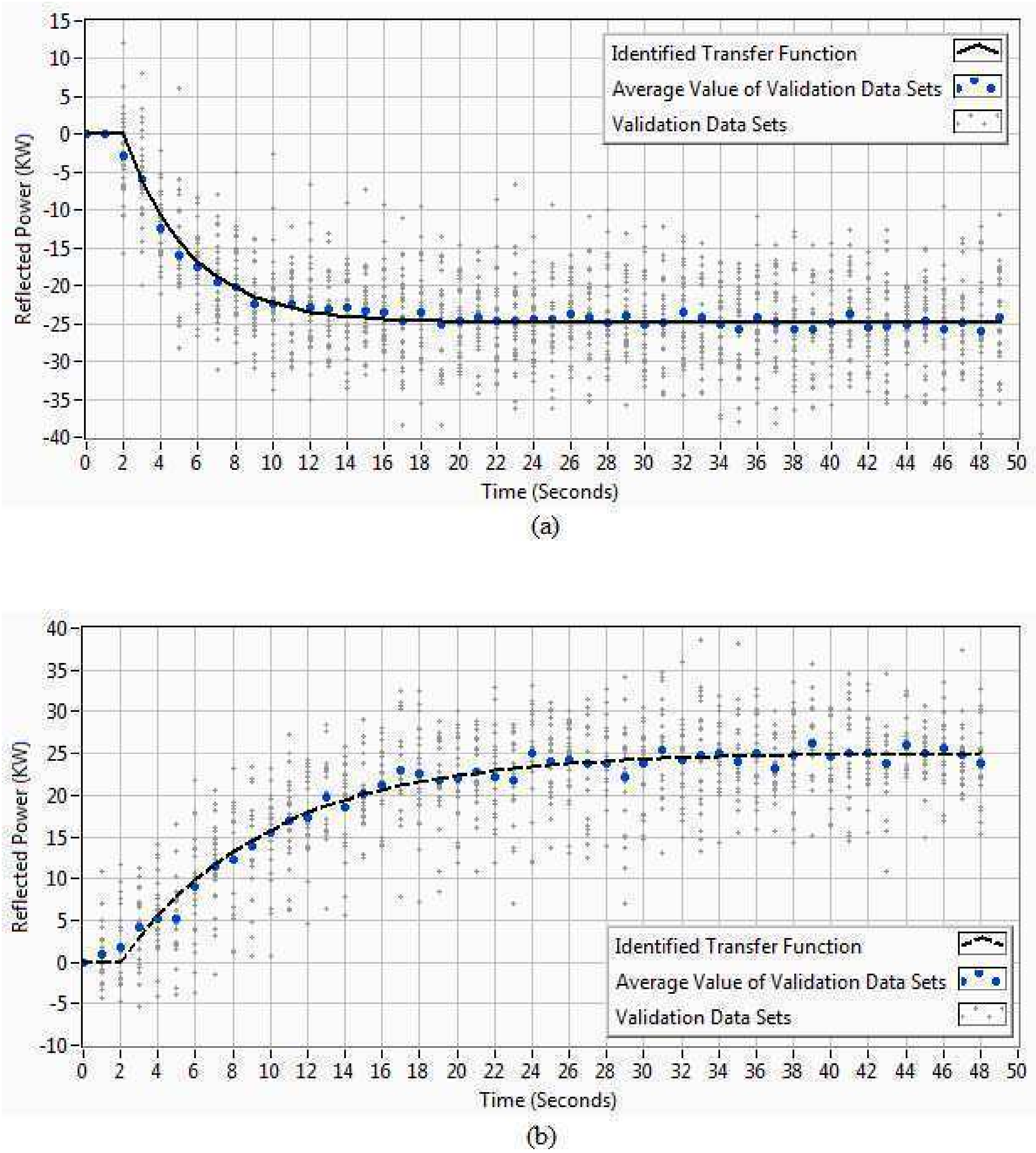


Figure 5.52: Validation of TF for reflected power, (a) when cathode current is increased from normal operating value, (b) when cathode current is decreased from normal operating value.

Equation (5.29) and (5.30) shows that the time constant for emission signal for both the cases, a)when cathode current increases and b) when cathode current decreases do not differ by large amount and thus for practical cases can be taken as equal. Whereas for RF reflected power (Eq.(5.29) and (5.30))the time constant for both the cases differs significantly thus making the system non-linear. Therefore at the time of controller design for actual implementation one has to take care about this.

Figure 5.42 shows the variation of cathode current (value for optimised operation) and emission with respect to the cathode run time. From Figure 5.42 it can be seen that the cathode current setting for optimum operation changes continuously throughout the cathode lifetime. Further it is observed that the emission seems to remain maintained about $4Volts$ throughout the cathode life time for optimum operation of the machine. Thus from this data the important conclusion can be drawn that emission is the parameters which defines the system internal state rather than cathode current. Whereas cathode current is only used as the control input to attain the desired emission level in the machine.

From Figure 5.43 it can be concluded that emission does not depends upon RF frequency change or RF frequency detuning (δF). The change in cathode current only changes the minimum value of reflected power at cavity resonant frequency (C_0 in equation Eq.(5.28)) and thus shifts the curve up or down, without changing the width and amplitude of the curve. Moreover it seems that the value of C_0, also dependent on some other unidentified factors. From this figure (by zooming on the selective portion of the figure) it can also be seen that the response of reflected power is very fast to the changes applied to RF frequency, and thus the reflected power always settles to its final value within one second of sampling time.

Summary

The system identification experiments performed over the Microtron has enabled us to formulate the Microtron model that was not available before this thesis work. This model is specifically aimed towards explaining the static and dynamic dependence of FCT, Emission, beam position (x,y) and reflected power over the control parameters cavity RF frequency deviation, cathode current and acceleration start point. The analysis of the results also brought out some of the observations that are important for designing the suitable controller for Microtron, like the emission current is the parameter that defines the systems internal state, for example for the Microtron at Indus complex the emission value of 4 volts defines the systems optimum operating point. This emission signal does not depends on the cavity RF frequency detuning. The beam position (x,y) depends on the reflected power. The reflected power depends on the cavity RF frequency detuning and cathode current. Emission current depends upon the cathode current. The FCT current depends upon the emission current and the cavity RF frequency detuning. This shows that the suitable controller can be designed for independently controlling the beam position and emission current simultaneously by using cathode current and cavity RF frequency detuning as control variables. Also as the FCT depends on the emission current as well as the RF detuning this independent controllability will be achieved at the expanse of decreased Microtron efficiency.

5.2.4 Microtron model

Based on the experimentally identified interdependence between different parameters discussed in previous section, the Microtron is modelled as a four-input five-output non-linear model. The inputs to the Microtron system are Cathode current (I_{ca} in Amperes) that controls the temperature of LaB_6 cathode inside Microtron RF cavity, RF frequency ($\check{r}_{fre}$ in GHz) that provides the basic RF signal which is amplified by amplifier stage and fed to the RF cavity for producing the required electric field in the cavity, acceleration start point in terms of I_{ca} (I_{ca}^{ASP} in Amperes) which defines the system state and depends on various factors, Cavity resonant frequency ($\check{r}_{cav-fre}$ in GHz) it is the resonant frequency of the cavity at the particular time and depends primarily on the cavity temperature and electron emission level in cavity. The outputs from the Microtron system are Emission ($\hat{e}$ in Volts) that gives the measure of electrons emitted from cathode, Fast current transformer signal ($\check{f}$ in Volts) that gives the measure of electrons actually accelerated to 20MeV level, beam position (X and Y in mm) at the extraction point, reflected power signal (g in Volts) that gives the measure of power reflected by the cavity. The different model blocks are codded using

$$\hat{e} = \begin{cases} 4.350 I_{ca} - 126.60 & for\ I_{ca} \leq I_{ca}^{ASP} \\ 1.324 I_{ca} - 35.11 & for\ I_{ca} > I_{ca}^{ASP} \end{cases} \tag{5.33}$$

$$\check{f}_E = \begin{cases} 0.2 & for\ I_{ca} \leq I_{ca}^{ASP} \\ -0.1318\hat{e}^2 + 1.389\hat{e} - 3.036 & for\ I_{ca} > I_{ca}^{ASP} \\ 0 & for\ \hat{e} < 3.1 \\ 0.1 & for\ \hat{e} > 7.4 \end{cases} \tag{5.34}$$

$$\breve{f} = \begin{cases} 0.1 & for\ \delta f < -0.6\ or\ \delta f > 0.6 \\ -5.007(\delta f)^5 - 2.889(\delta f)^4 + 0.034(\delta f)^3 & \\ \qquad - 0.375(\delta f)^2 + 0.034(\delta f) + \breve{f}_E & for\ -0.6 < \delta f < 0.6 \\ 0 & for\ \breve{f} < 0 \end{cases} \tag{5.35}$$

$$X_E = \begin{cases} 0.089\hat{e} & for\ I_{ca} \leq I_{ca}^{ASP} \\ 0.293\hat{e} & for\ I_{ca} > I_{ca}^{ASP} \end{cases} \tag{5.36}$$

$$Y_E = \begin{cases} 1.391\hat{e} & for\ I_{ca} \leq I_{ca}^{ASP} \\ 4.571\hat{e} & for\ I_{ca} > I_{ca}^{ASP} \end{cases} \tag{5.37}$$

$$X_{\delta f} = 0.920(1 - e^{-\frac{(\delta f)^2}{0.251}}) \tag{5.38}$$

$$Y_{\delta f} = 1300(1 - e^{-\frac{(\delta f)^2}{60.160}}) \tag{5.39}$$

$$g = 0.115(1 - e^{-\frac{(\delta f)^2}{0.576}}) + C_0 \quad Where\ C_0 = 0.067\frac{(\breve{F}_{max} - \breve{F})}{\breve{F}_{max}} \tag{5.40}$$

Where $\delta f = (\breve{r}_{fre} - \breve{r}_{cav-fre})$ is the deviation of RF generator frequency from the cavity resonant frequency expressed in MHz and the beam position X and Y using Eq. (5.36) to (5.39) are calculated as below.

$$Beam\ Position(X, Y) = \begin{cases} X = X_E + X_{\delta f} - 1.6 \\ Y = Y_E + Y_{\delta f} - 24.8 \end{cases} \tag{5.41}$$

Noise at emission signal $Noise(E)$ and noise at the reflected power signal $Noise(RP)$ are coded by autoregressive model using

$$Noise(E) = \frac{e(t)}{1 - 0.9982z^{-1} - 0.0007436z^{-2}} \tag{5.42}$$

$$Noise(RP) = \frac{e(t)}{1 - 0.9983z^{-1} - 0.000969z^{-2}} \tag{5.43}$$

$$Where\ e(t) = White\ noise$$

and the dynamic responce transfer function (TF) for emission signal (E) and reflected power (RP) are modelled as Eq. (5.44) and (5.45).

$$TF(E) = \frac{e^{-2.0s}}{7.0s+1} \tag{5.44}$$

$$TF(RP) = e^{-2.0s} \tag{5.45}$$

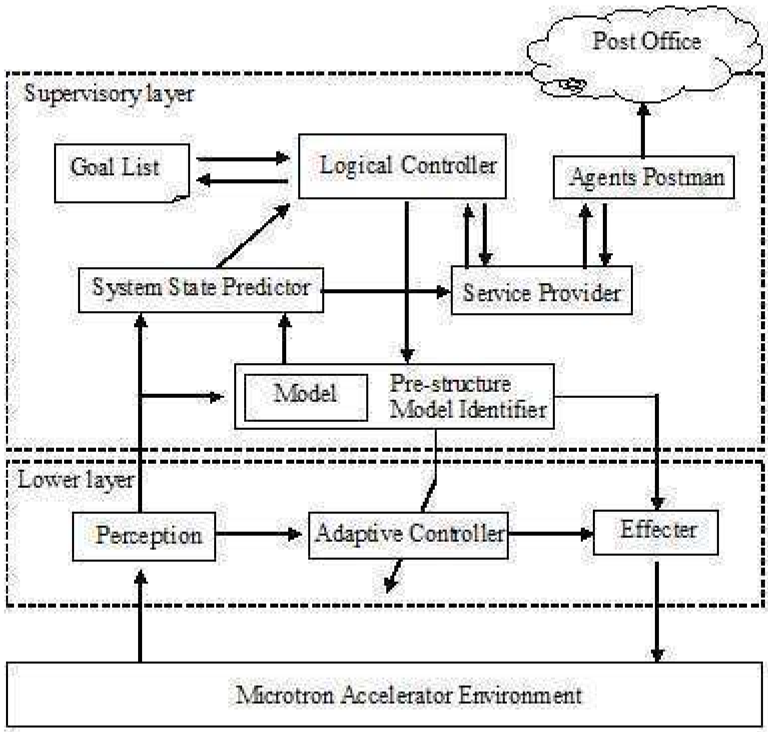


Figure 5.53: Microtron Agent Architecture.

5.2.5 Microtron Intelligent agent architecture

For system like Microtron accelerators, that exhibits dynamic nonlinear input-output behavior, agent architecture requires augmenting functionality for adaptive feedback controller, dynamic and static model identifiers, and system state predictors based on historical data alongwith the supervisory level optimisation, communication, coordination and planning functionalities.

The abstract architecture of Microtron agent is made with the architecture shown in figure 5.53. This agent architecture comprises of two loops, the first loop: comprised of perception, adaptive controller and effecter blocks. This loop is responsible for continuously maintaining the machine operating point under dynamic conditions. The loop works on the principle of sense-think-control cycle where the accelerator environment is continuously sensed and if some drift in the operating point is observed the corresponding corrective action is calculated by the adaptive controller and applied to the accelerator environment through effecter. The second loop is the supervisory loop responsible for autonomously controlling the agent actions and the interaction with other agents. The pre-structure model identifier block when required /asked by the logical controller, identifies the plant model in the pre-structured model form, by directly taking the control of effecter and perception blocks and using the predefined action recipe. This block also provided this identified model to other blocks like system state predictor block, adaptive controller block and logical controller block for their functions. The system state predictor block continuously tries to learn the system dynamics and predicts the system dynamics for future n steps using the auto-regressive moving average(ARMA) algorithm. This block also provides the functionalities of predicting the future machine states/parameters under the influence of dynamics using the currently identified Microtron model. Service provider block is the communication interface of the agent with the other agents. This block is responsible for serving the requests obtained from different agents and from logic controller which requires some data from other agents. The postman is the communication medium between the agent and the post office for exchange of messages between different agents. The logic controller is the brain of the agent and is responsible for managing and synchronizing all the activities of the agents towards the achievement of goals.

Algorithm 5: Capability identify I_{ca}^{ASP}, $\breve{f}(I_{ca})$ and $\hat{e}(I_{ca})$.

```
Data: percept ⟨ emission signal ê, FCT signal f̆ ⟩, actuator ⟨ I_ca⟩
Result: calculated value of I_ca^ASP, functions f̆(I_ca) and ê(I_ca)
/*Declare self state as identification state */
initialise self state ← identification state ;
repeat
    /*decrease cathode current by small value and apply */
    I_ca = I_ca − 0.1A ;
    APPLY I_ca ;
    /*repeat till FCT fall below 0.3V */
until (f̆ > 0.3V);
/*move system state to identification start point */
/*decrease cathode current to provide sufficient margins */
I_ca = I_ca − 0.4A ;
APPLY I_ca ;
/*The system has reached at the state, to start identification */
repeat
    /*increase cathode current by small value and apply */
    I_ca = I_ca + 0.1A ;
    APPLY I_ca ;
    /*read Emission and FCT signal values from CRO traces  */
    READ f̆ and ê ;
    /*log Emission and FCT values against cathode current values */
    LOG f̆ and ê for the I_ca ;
    /*repeat till Emission increases above 7.0V */
until (ê < 7.0V);
/*compute I_ca^ASP as the acceleration start point(ASP) where
  function ê(I_ca) shows the sharp slope change */
COMPUTE I_ca^ASP ;
/*compute f̆(I_ca) and ê(I_ca) with least square fitting method */
COMPUTE functions for f̆(I_ca) and ê(I_ca);
/*update I_ca^ASP, f̆(I_ca) and ê(I_ca) in agent beliefs */
UPDATE I_ca^ASP, f̆(I_ca) and ê(I_ca);
/*restore system state to active state */
self state ← active state ;
```

Algorithm 6: Capability identify cavity resonant frequency ($\breve{r}_{cav-fre}$).

```
Data: percept ⟨ reflected power signal g, FCT signal f̆ ⟩, actuator ⟨ r̆_fre ⟩
Result: calculated value of r̆_cav-fre and function f̆(δr̆_fre)
/*Declare self state as identification state */
initialise self state ← identification state ;
repeat
    /*decrease RF frequency by small value and apply */
    r̆_fre = r̆_fre − 100KHz ;
    APPLY r̆_fre ;
    /*repeat till change in reflected power is more than 20mV */
until (g − g_start < 20mV);
/*this means we have reached on left side of the curve */
/*to get the course value of cavity frequency scane RF now
  towards right side */
repeat
    /*increase RF frequency by small value and apply */
    r̆_fre = r̆_fre + 100KHz ;
    APPLY r̆_fre ;
    /*read reflected power signal value from CRO and log  */
    READ g ; LOG g ;
    /*repeat till change in reflected power is more than 20mV */
until (g − g_start < 20mV);
/* compute r̆_cav-fre as r̆_fre for which g is minimum  */
COMPUTE cavity resonant frequency r̆_cav-fre ;
/*Now scan RF again to get the fine value of cavity frequency */
/*Move to lowest RF frequency value and apply */
r̆_fre = r̆_fre − 900KHz ;
APPLY r̆_fre ;
for (i=0; i<33; i++) do
    r̆_fre = r̆_fre + 50KHz ;
    APPLY r̆_fre ;
    READ g and f̆ ; LOG g and f̆ ;
end
/*compute f̆(δr̆_fre) with least square fitting method */
COMPUTE function for f̆(δr̆_fre);
/*update r̆_cav-fre andf̆(δr̆_fre) in agent beliefs */
UPDATE r̆_cav-fre andf̆(δr̆_fre);
/*restore system state to active state */
self state ← active state ;
```

Assuming that there exists a mechanism for identification of Microtron model by the "pre-structure model identifier"block using the algorithms 5 and 6. whenever the system response changes / agent decides to identify the system model. And assuming that there exists an adaptive controller that can successfully control the Microtron accelerator to the desired operating point given by $\breve{P}_{MIC} = \{\hat{e}, g, \breve{f}, x, y\}$ using control inputs $\breve{C}_{MIC} = \{i_{ca}, \breve{r}_{fre}\}$ and identified system model through control function Θ. Where $\hat{e} \in E$, $g \in G$, $\breve{f} \in \breve{F}$, $i_{ca} \in I_{ca}$ and $\breve{r}_{fre} \in \breve{R}_{fre}$. Now using the history of externally communicated (provided by TL-1 agent) beam position X and Y at Microtron output. The "System state predictor"module predicts the beam movement for next 'n'steps using the ARMA model fitted over last 40 beam position values of X and Y. The module computes the AR and MA coefficients using the ARMA function VI with AR order = 30, MA order = 29 and method = "Yule-Walker"at each iteration for newly obtained X and Y value.

Using the system model this block predicts the system state $s = \{\hat{e}, g, \breve{f}, i_{ca}, \breve{r}_{fre}, x, y\}$ at optimised operating point by minimising the cost function J_0 given by:

$$min \;\; J_0 \;=\; w_1(\frac{e}{\breve{f}}) + w_2 rp + w_3(1-\breve{f})^2 + w_4(x-\hat{x})^2 + w_5(y-\hat{y})^2 + \Upsilon \qquad (5.46)$$

$$\Upsilon \;=\; \begin{cases} 0 \;\; if \;\;\; i_{ca} - (I_{ca}^{ASP} + I_{ca}^{margin}) \geq 0 \\ 1000 \;\; if \;\;\; i_{ca} - (I_{ca}^{ASP} + I_{ca}^{margin}) < 0 \end{cases} \qquad (5.47)$$

where $w_1, w_2, w_3, w_4 \; and \; w_5$ are the weights selected according to the optimisation criteria, for example to get the operating criteria of best current from Microtron near beam position offset$(\hat{x}, \hat{y})$ from present operating point, the weight set is $W = \{w_1 = 0, w_2 = 0, w_3 = 1, w_4 = 1, w_5 = 1\}$. Whereas to get the operating criteria of best efficiency of Microtron near beam position offset$(\hat{x}, \hat{y})$ from present operating point, the weight set is $W = \{w_1 = 1, w_2 = 0, w_3 = 1, w_4 = 1, w_5 = 1\}$. Where $\hat{e} \in E$, $g \in G$, $\breve{f} \in \breve{F}$, $i_{ca} \in I_{ca}$, $\breve{r}_{fre} \in \breve{R}_{fre}$, $x \in X$, $y \in Y$, I_{ca}^{ASP} is the acceleration start point cathode current value obtained through model identification and I_{ca}^{margin} is the margin in cathode current set value required for stable operation. The cost function J_0 is formulated so as to selectively optimise the Microtron operating point based on the criteria of maximizing the Microtron efficiency, maximizing FCT (i.e. $\breve{f}$), minimizing RP (i.e g) and minimizing the beam position error with respect to the desired beam position. The function Υ is introduced in the equation to force the solution with sufficient cathode current margin.

For the case of individual optimisation goal, agent computes the optimised system state s_0 using Eq.5.46 with weight set $W = \{w_1 = 1, w_2 = 1, w_3 = 1, w_4 = 0, w_5 = 0\}$ that optimises the Microtron operating point irrespective of the beam position offset and focusses only on increasing the Microtron efficiency and beam current($\breve{f}$). Using s_0 the agent controls the Microtron with $\breve{C}_{MIC}$ obtained as:

$$\breve{C}_{MIC} = \Theta(\breve{P}_{MIC}) \ \ ; \ \ \breve{P}_{MIC} \subset s_0 \tag{5.48}$$

For the case of cooperative optimisation goal, on request by TL-1 agent /self trigger event, the agent generates a set $\mathbf{P}$ of possible beam position vectors $p = (\hat{x}, \hat{y})$, for the desired beam position $p_d = (x_d, y_d)$ and search distance d_1 and d_2 in horizontal and vertical plane such that:

$$\mathbf{P} = \{(\hat{x}, \hat{y}) \mid \ d_1 < (\hat{x} - x_d) < -d_1 \ \ and \ \ d_2 < (\hat{y} - y_d) < -d_2\} \tag{5.49}$$

Now against each element 'p 'of $\mathbf{P}$ the agent computes the optimised system state 's'according to the selected policy criteria to generate the set $\mathbf{S}$ given as:

$$\mathbf{S} = \{s \mid \ s \ = \ f_M(p) \ ; \ p \in \mathbf{P}\} \tag{5.50}$$

where f_M is the model function that maps 'p'to 's'. It then uses $\mathbf{S}$ for deciding the self operating point and the TL-1 agents operating point by maximizing the coalition value according to the selected policy.

5.3 Multiagent based control for accelerators

Agent-based control offers the ability to learn the patterns in system dynamics and use this information in determining the optimal, or near optimal control schema. Further to this by propagating this information among different agents in a multi-agent environment the global goals and global constraints could be easily handled. Thus in this section an agent-based methodology for controlling the pre-injector and transport line operations is discussed. Where agents learn the patterns observed in the system dynamics and optimize there individual operations as well as there joint goals based on the learned patterns. For controlling Microtron, the agent architecture with model assisted adaptive controller for realizing feedback control action at lower layer and goal based logic controller with pre-structure model identifier along with the pattern recognizer at supervisory layer discussed in previous section is used. For controlling the TL-1, the agent with a model-based, goal-based modular architecture discussed in section 5.1 is used that optimizes the TL-1 control using differential evolution based tuning plans.

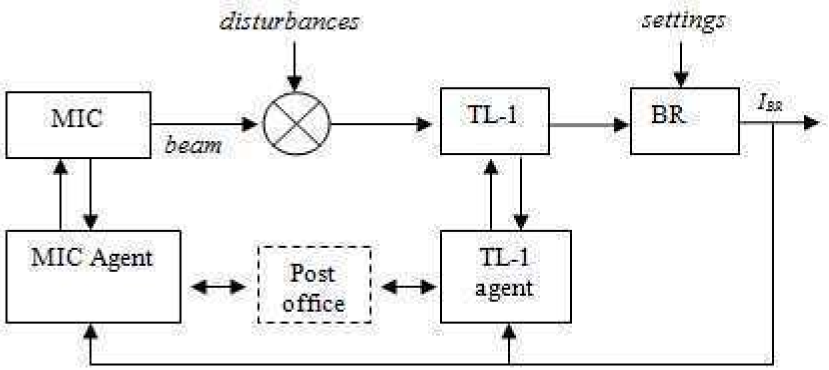


Figure 5.54: Block diagram of multi-agent based control of Microtron and TL-1.

5.3.1 Cooperative accelerator tuning

Figure 5.54 shows the multi-agent based control system block diagram for controlling Microtron and TL-1. The multi-agent based control of Microtron and TL-1 towards the cooperative tuning requires that both of the agents should try to maintain their individual operation to the optimum according to their local priorities on one hand and cooperatively decides their operating points such that their joint goal of increasing the overall injection current in the booster is achieved. As the beam current in booster depends on two factors first: the current produced by Microtron, and the second: the transfer efficiency (normalised beam current in TL-1) in TL-1, therefore the cost function J_1 is formulated to maximize the product of these two currents as:

$$maxJ_1 = I_{MIC}(x_t, x'_t, y_t, y'_t, \check{P}_{MIC}) \times I_{TL1}(x_t, x'_t, y_t, y'_t, \check{P}_{TL1}) \tag{5.51}$$

where the I_{MIC} and I_{TL1} are the beam current at Microtron output(which is a function of beam parameter and Microtron operating point($\check{P}_{MIC}$)) and normalised beam current at TL-1 output (which is a function of beam parameter and TL-1 operating point($\check{P}_{TL1}$)). For achieving the cooperative accelerator tuning, the agents jointly identifying the operating points that maximizes the cost function J_1 given by Eq.(5.51) for the measured beam parameters.

For the case of cooperative optimisation based on the dynamics learning with additional conditions that the demand of change in the TL-1 magnet settings is to be reduced while always maintaining the required level of injection current in booster. The Microtron agent at the time of deciding the new operating point for optimisation cooperatively maximises the cost function J_2 given by Eq.(5.52) considering the 'n'steps ahead future disturbances based on the past movement history provided by the "system state predictor"block.

$$maxJ_2 = \sum_{i=1}^{n} I_{MIC}(x_i, x'_i, y_i, y'_i, \check{P}_{MIC}) \times I_{TL1}(x_i, x'_i, y_i, y'_i, \check{P}_{TL1}) \tag{5.52}$$

The cost function J_2 is based on the fact that optimised operating point for the two agents subject to the above said criteria will be the one that not only maximizes the injection current in booster but also maintains this maximised level of current in booster for future durations under the influence of external disturbances. Thus J_2 is formulated to maximize the sum of product of I_{MIC} and I_{TL1} for the predicted future beam position movements.

For the cooperative optimisation case the Microtron agent generates optimised state set $\mathbf{S_{MIC}}$ for the search position vector set $\mathbf{P}$ (given by Eq.5.49) using Eq.5.50.

$$\mathbf{S_{MIC}} = \{\mu_1, \mu_2, \mu_3, ..., \mu_n\} \tag{5.53}$$

similarly the TL-1 agent generates the state set $\mathbf{S_{TL1}}$ with states 'λ 'as following:

$$\mathbf{S_{TL1}} = \{\lambda \mid \lambda = f_{TL1}(p)\ ;\ p \in \mathbf{P}\} \tag{5.54}$$

$$\lambda = \{x, x\prime, y, y\prime, i_{TL1}\} \tag{5.55}$$

where f_{TL1} is the mapping function from position vector 'p'to state 'λ '. Now the agent selects best coalition($\hat{\mu}$,$\hat{\lambda}$) for tuning of Microtron and TL-1 as the optimised states for applying to the system as the one that maximizes the coalition value $\mathbf{J_1}$ given by:

$$max\ \mathbf{J_1} = \{fct * i_{TL1} \mid (\mu, \lambda) \in \mathbf{S_{MIC}} \times \mathbf{S_{TL1}}\} \tag{5.56}$$

$$fct \in \mu\ \ ;\ \ i_{TL1} \in \lambda\ \ ;\ \ \mu \in \mathbf{S_{MIC}}\ \ and\ \ \lambda \in \mathbf{S_{TL1}} \tag{5.57}$$

$$\mathbf{S_{MIC}} \times \mathbf{S_{TL1}} = \{(\mu, \lambda)\ \mid\ \mu \in \mathbf{S_{MIC}}\ \ and\ \ \lambda \in \mathbf{S_{TL1}}\} \tag{5.58}$$

For the cooperative optimisation with system dynamics learning case the Microtron agents generates the predicted beam position set $\boldsymbol{\xi}$ for 'n'future steps based on the history of beam position movement as discussed in section 5.2.

$$\boldsymbol{\xi} = \{\varrho_1, \varrho_2, \varrho_3, ..., \varrho_n\}\ \text{ where }\ \varrho = (x, y) \tag{5.59}$$

now corresponding to each element ϱ of $\boldsymbol{\xi}$ the search beam position set $\mathbf{P}$ is computed using Eq.5.49 to make set $\hat{\mathbf{P}}$ as:

$$\hat{\mathbf{P}} = \{\mathbf{P_1}, \mathbf{P_2}, \mathbf{P_2}, ..., \mathbf{P_n}\} \tag{5.60}$$

Corresponding to each element of $\hat{\mathbf{P}}$ the optimised coalition state $\vartheta = (\hat{\mu}, \hat{\lambda})$ are calculated using Eq. 5.53 to 5.61 to get set $\hat{\boldsymbol{\varsigma}}$ as.

$$\hat{\boldsymbol{\varsigma}} = \{\vartheta_1, \vartheta_2, \vartheta_3, ..., \vartheta_n\} \tag{5.61}$$

From $\hat{\boldsymbol{\varsigma}}$ the agent selects best coalition ϑ for tuning of Microtron and TL-1 as the optimised states for applying to the system as the one that maximizes the cost function $\mathbf{J_2}$ computed as:

$$max\ \mathbf{J_2} = \{\sum_{i=1}^{n} fct * i_{TL1} \mid (\mu, \lambda) \in \mathbf{S_{mic}} \times \mathbf{S_{tl1}}\} \tag{5.62}$$

$$fct \in \mu \ \ ; \ \ i_{TL1} \in \lambda \ \ ; \ \ \mu \in \mathbf{S_{mic}} \ \ and \ \ \lambda \in \mathbf{S_{tl1}} \tag{5.63}$$

$$\mathbf{S_{mic}} \times \mathbf{S_{tl1}} = \{(\mu, \lambda) \ \mid \ \mu \in \mathbf{S_{mic}} \ \ and \ \ \lambda \in \mathbf{S_{tl1}}\} \tag{5.64}$$

$$\mathbf{S_{mic}} = \{\mu \mid \mu = f^{\vartheta}_{MIC}(\varrho) \ ; \ \varrho \in \xi\} \tag{5.65}$$

$$\mathbf{S_{tl1}} = \{\lambda \mid \lambda = f^{\vartheta}_{TL1}(\varrho) \ ; \ \varrho \in \xi\} \tag{5.66}$$

where the mapping functionf^{ϑ}_{MIC} and f^{ϑ}_{TL1} computes the states μ and λ of Microtron and TL-1 for $\boldsymbol{\xi}$, when they are tuned at operating point as per ϑ.

5.3.2 Simulation results

For checking the effectiveness of this scheme the system comprised of accelerator model, Microtron agent and TL-1 agent as shown in figure 5.54 is simulated. The results of the

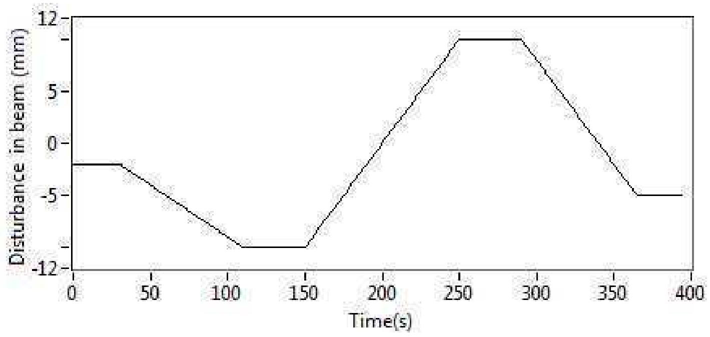


Figure 5.55: Disturbance in beam added at Microtron output.

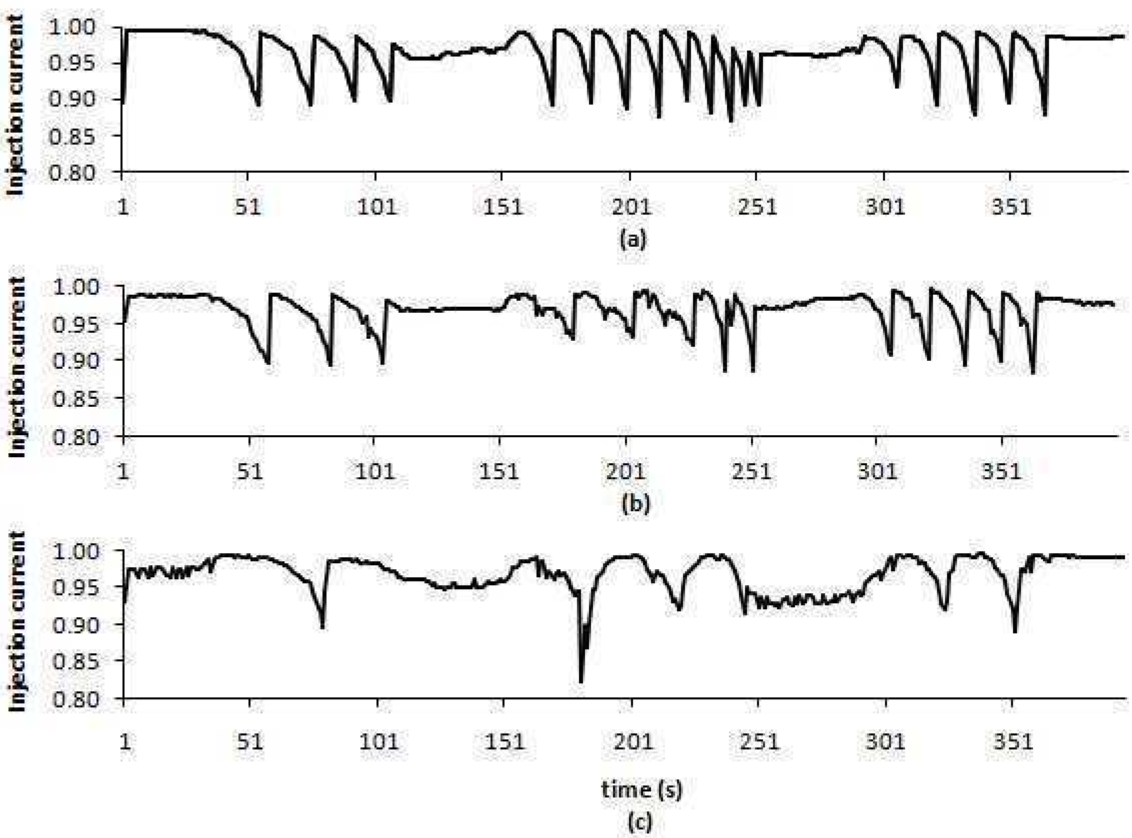


Figure 5.56: Normalized injection current in booster for, (a) when Microtron and TL-1 agents optimizes there operations individually, (b) when both of the agents work cooperatively for optimizing the injection current as well as reducing the no of changes in TL-1 settings, (c) when both of the agents work cooperatively with dynamics learning case.

agent based control when beam coming out of Microtron is subjected to the disturbance shown in figure 5.55 for three different scenarios, scenario1: when both the TL-1 and Microtron agent works independently to achieve their individual goal, scenario 2: when both of the agents work cooperatively to maximize the booster injection current, and scenario 3: when both of the agents cooperatively with dynamics learning capability works to maximize the booster injection current with 20 steps ahead predicted beam dynamic behavior were calculated for the beam disturbance shown in figure 5.55 Figure 5.56 shows the injection current in booster for the three scenarios. Figure 5.57 shows the different operating points for which the TL-1 was adjusted by the TL-1 agent for the three scenarios. Figure 5.58 shows the beam current provided by the Microtron for the

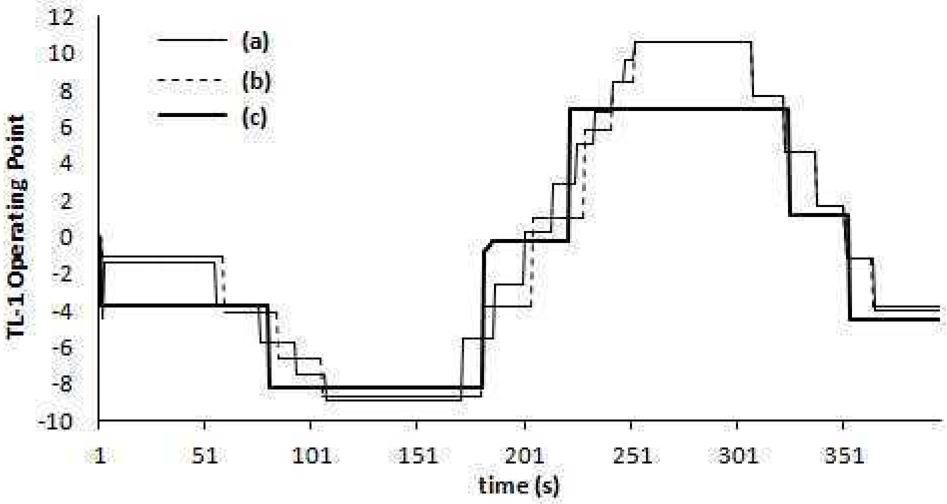


Figure 5.57: TL-1 operating points for which TL-1 was adjusted by TL-1 agent for, (a) scenario 1, (b) scenario 2, (c) scenario 3.

three different scenarios.

From Figure 5.56(b) it can be seen that the booster current in case of cooperative optimization is showing lesser number of variation in Booster current with respect to the case when agents work individual (figure 5.56(a)). This variation further reduces with the application of the dynamics based learning algorithm (figure 5.56(c)). From figure 5.57 it can be seen that for the case of dynamics based learning the changes in operating point of TL-1 is minimum as compared to other two cases. From Figure 5.58 the effect

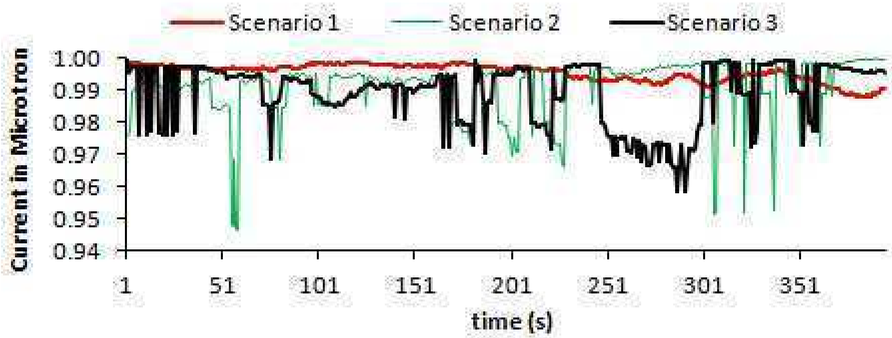


Figure 5.58: Beam current at Microtron output when Microtron is operated at different operating points by Microtron agent under different scenarios.

of cooperative optimization in choosing the operating points by Microtron agent can be seen clearly. Where for the scenario 1 the Microtron current remains always at its best operating value but for the other two scenarios the agent gives priority to the common goals and thus opted for slightly sub optimal operating points.

5.3.3 Summary and discussion

In this section the application of a multi-agent based approach in control of pre-injector and transport line at synchrotron accelerator facilities was discussed. The novel concept of cooperative optimization with system dynamics learning capability for multi-agent based control approach was presented. The individual agent architecture for controlling Microtron and Transport line and their organization as multi-agent for cooperative control was designed. The simulation results of the presented concept for controlling the pre-injector accelerator Microtron and Transport line under the influence of disturbance on beam shows that this scheme can be used successfully for their optimal control without operator interventions. With the implementation of agent based controller for individually controlling TL-1 and Microtron the effective operation parameter limits can be increased. Further with the inclusion of proposed cooperative tuning goals and related formulated plans for the individual intelligent agents this effective tuning can be achieved while honouring the soft constraints imposed by means of prioritized individual goals to agents for some of the operating parameters such as efficiency of Microtron operation, limiting cathode current, minimizing the operating point changes to TL-1. And further this can be done on case to case basis thus effectively enhancing the operating flexibility to operator. Thus enabling the operator to select the optimization policy for individual agents. Like in the beginning of a new cathode life cycle the operator can choose to select the goal of cooperative optimization with maximum efficiency of operation for Microtron, whereas towards the end of the cathode life the operator may wish to select the goal of cooperative optimization with cathode current limited to some (I_{ca}^{max}) maximum value so as to avoid the possibility of arcing in cavity and also to support the elongation of possible life of cathode till next proposed shutdown. Thus this can avoid the unlikely event of cathode failure that may result in an unintended shutdown of the facility.

5.4 Distributed Intelligent agent based beam orbit control of synchrotron radiation sources

In the beam lines (BL) the SR position is highly dependent on the electron beam position and angle at the source point. The tuning of accelerator for getting the desired electron beam position and angle at the source point is a time consuming and regular job done during commissioning of new beam lines or when accelerator is operated at new operating point. For accelerator control and tuning this section presents a novel Intelligent Agent (IA) based operator support and beam orbit control scheme. The proposed multi-agent based scheme is well suited for the multilayer control system architectures of synchrotron radiation sources. The scheme successfully distributes the orbit control job to multiple low complexity reactive agents that work simultaneously and control the local orbit for individual beam lines and insertion devices in an optimized manner. The performance of the beam line control agents and insertion device control agents is evaluated on periodic basis by monitoring agents. The gradient based constrained trainer agents are proposed that coordinates the agents training to achieve the accelerator tuning goal in this multi-agent environment. The logic based fault assistance agents are proposed for providing the fault identification and isolation. At top of all these agents the model based goal based orbit control agent is proposed that interacts with the user and database to generate the lower layer agents and control their activities in a synchronized and systematic manner. The proposed scheme of beam orbit control in particular is very useful for machines like INDUS-2, where new beam lines are in the process of commissioning as this scheme reduces the operator efforts and accelerator tuning time for providing beam to new beam lines. Further it extends the beam availability to other beam lines (already installed and in use beam lines) as the agent tunes the accelerator in systematic way and under constraints on local orbit bump leakage thereby enabling the use of other beam lines for routine experiments which otherwise was not possible.

5.4.1 Introduction

Synchrotron radiation sources provide light of very high intensity, Photon flux of order of 10^{13} photons/s/0.1% bandwidth light is a common figure for SRS facilities [24]. This light is of wide bandwidth ranging from visible to hard X-rays. Further to this the light is emitted in a very narrow angle in the forward direction. Due to these appealing features the SR user community is growing day by day expanding the new areas of SR utilization. In electron storage rings the SR is emitted by the relativistic electrons (electron energy from hundreds of MeV to few of GeV is common) when they pass through dipoles or through insertion devices such as wigglers [118] and undulators [118] that are part of electron storage rings. For using the SR in experiments there are beam lines connected to the storage ring that transports the SR from electron orbit to the experimental station. The experiment stations are the part of the beam lines, designed specifically to perform the particular type of experiment on the sample. In the beam lines the SR position is highly dependent on the electron beam position and angle at the source point. At the time of commissioning of a new beam line, it is first coarsely aligned independently as per the design and then the alignment is fine tuned jointly by tuning along with the source point (electron beam position in storage ring). The alignment restrictions on the BL side require the residual misalignment to be compensated by correcting the electron beam position and angle at the source point. The electron beam position and angle are corrected by applying the local orbit bump at source point. In normal practice two methods are used for generating the local orbit bump. One is three corrector based closed orbit bump and the other is four corrector based closed orbit bump [46]. Further the four corrector based scheme can be represented by superposition of two distinct three corrector bumps [18]. In the process of manual tuning of accelerator by operators to obtain the desired beam position and angle at source point the operator first calculates the initial corrector values using three/four corrector based scheme and then applies it to the machine. Due to difference in the theoretically calculated values of machine parameters and the actual machine parameters at given operating point the applied orbit bump does not close properly and leaks out to the entire electron beam orbit thus producing disturbance throughout the ring. Operator now tries to close the bump to the required level by perturbing the different correctors used for generating the bump in an iterative manner. This task of finding the suitable corrector strength that satisfies the BL alignment and also closes the bump to within limits is a time consuming and routine job done during commissioning of new beam lines or when accelerator is operated at new operating point.

Intelligent agent is an autonomous software entity situated in an environment. It is capable of observing its environment (partially of fully) through sensors and can proactively or reactively act upon the environment through actuators. It autonomously directs its activities towards achieving its goals. Different agent architectures are proposed for implementing intelligent agents by researchers and are discussed in detail in chapter 2. The simple reactive architectures like subsumption architecture are the most popular architectures used in applications demanding the fast reactive behaviours in real time, like robotics. The Logic based, model based and goal based architectures are some other architectures used for applications demanding complex behaviour implementations. The multi-agent system is the system that solves the problem through work division among multiple agents of relatively low complexity and limited actuator/sensor coverage over environment.

For the distributed multilayer control architecture of synchrotron radiation sources, a novel intelligent agent based operator support and beam orbit control methodology is formulated. The proposed methodology distributes the orbit control job among five different types of agents. To take the advantage of layered control system architecture these different types of agents can be implemented at different layers with varying computation power, memory and communication resources. The beam line control agents and insertion device control agents are designed with the simple subsumption architecture suited for implementation at the lower layer controllers demanding very less computation power. These agents augment the basic behaviours for optimally controlling the local beam parameters in a reactive manner. Being simple and localized in environment these agents exhibit a fast reactive control cycle to attain position/angle and maintain position/angle type of goals. Since these lower level agents acquire only the local information available to them to attain their local goals and as the global goal of controlling the orbit is attained by the combined efforts of these low level agents, the performance degradation of any one agent can affect the overall global goal. For this, the performance of the beam line control agents and insertion device control agents are evaluated on periodic basis by monitoring agents situated at the higher level of control architecture layer. The monitoring agents periodically acquire the data from multiple agents and evaluate the performance of the individual agents and provide this feedback to them. A low level agent with average or poor performance takes assistance from trainer agents that are again at higher level in hierarchy and implements the complex behaviours for gradient based constrained reinforcement learning plans. These trainer agents coordinate the student agents training for skill development and actually refine/correct those beliefs whose knowledge content has degraded due to the dynamics of changing environment. The trainer agent forms the team of low level agents comprising of student agent (the agent which is trained to generate the close bump using correctors) and the cooperating agents which provide the information about the actions of student agents on the global beam orbit. The trainer agent in a coordinated fashion suggests the actions to be performed by the student agent to generate the closed bump. It then evaluates the utility of the suggested action by gathering the information from the other cooperating agents taking part in training process and decides the next action to be suggested to the student agent. This iterative process continues till the learning goal is completed. In case the agent finds that the performance is altogether not in accordance with the expected it can seek fault assistance from fault assistance agents. The fault assistance agent finds the faulty sensor and actuators by logical equation evaluation. At the highest hierarchy there is the model based goal based orbit control agent that interacts with the user and database to generate the lower layer agents and control their activities in a synchronized and systematic manner. Also at the same hierarchy there is the closed orbit minimization agent which intermittently checks the corrector strength and the beam positions and decides the optimized orbit for orbit maintenance goal.

The proposed methodology extends the use of multi-agent based control to the field of local beam orbit control in synchrotron radiation sources thereby improving the overall reliability and beam availability to the users. It further improves the available beam quality to users through prioritized optimization in case the multiple beam lines demand the beam correction simultaneously. Further to this the work presents a novel gradient based cooperative multi-agent learning scheme. The gradient based reinforcement learning schemes have been proposed in past by many researchers [8, 17, 19, 123]. The scheme presented here differs from all of these as all the previously proposed schemes are focused on identifying the suitable agents actions through their interaction with environment which transforms the system from current state to the desired state but they have not considered the cases where the agents actions performed for learning may lead the environment/system to a state such that the system can no longer be used further. For example if the learning activity by a beam line control agent disturbs the beam at nearby or other high priority beam lines such that the experiment data gets affected then the purpose of the orbit control is defeated, or if the perturbation introduced by the learning agent disturbs the overall beam to an extent that the complete beam gets killed then there is no beam in orbit to control. Therefore for such cases this section presents the new scheme of agents skill learning under constrained condition for improving the system reliability. The presented accelerator control scheme further reduces the operator efforts towards accelerator tuning by automatically performing the beam line tuning based on the learned tuning patterns from the past and present operator feedback about the operation quality.

5.4.2 Closed orbit bump and accelerator environment

In synchrotron radiation sources the highly energetic electrons are made to follow a nearly circular path through magnetic optics comprising of dipoles for bending of electron beam and quadrupoles for focusing/defocusing of beam [68]. The electron beam position along the beam path in transverse plane (i.e. the plane perpendicular to the beam motion) is controlled using corrector magnets. Depending upon the direction in which the corrector magnet gives kick to the beam they are categorized into vertical corrector magnets and horizontal corrector magnets. Electron beam position monitors (BPM) are used for measuring the beam position at different locations in the ring. For successful storage of the electrons in the ring only some specific set of dipole and quadrupole magnet settings are allowed which are governed by the accelerator beam dynamics [68] and specifies the accelerator operating point. In normal practice due to presence of different types of errors (alignment error, field error, manufacturing tolerances etc.) in components the electron beam deviates from its ideal design trajectory (normally called as golden orbit). The corrector magnets are used to control the orbit in order to bring it near to the golden orbit (ideally if possible it should match with the golden orbit). This is known as closed orbit distortion (COD) correction. After the beam is corrected for COD if one wants to purposefully change the orbit at desired location then the kicks to the beam are required to be applied using group of correctors. Different schemes for this are adopted in normal practice like two corrector closed bump, three corrector closed bump and four corrector closed bump [36]. Choice of using the particular scheme is normally dependent on the machine operating parameters (ψ and β) and machine design (location of correctors and BPMs). For simplicity we will restrict to three corrector bumps and four corrector bumps only

Three corrector bump

In this scheme three correctors are used to generate the local bump where the first corrector opens the orbit i.e. deflects the beam away from the orbit the second corrector deflects the beam towards the orbit and the third corrector closes the bump by correcting the angle as shown in figure 5.59. In figure 5.59 let $\theta_1, \theta_2, \theta_3$ be the kick angles generated by three corrector magnets CV_1, CV_2, CV_3, then for closed bump condition the $\theta_1, \theta_2, \theta_3$ must satisfy the Eq. (5.67) [46] where β_i is the β-function at the i_{th} corrector magnet, $\psi_{ij} = \psi_i - \psi_j$ is the phase advance from the i_{th} to j_{th} corrector magnet.

$$\frac{\theta_1\sqrt{\beta_1}}{\sin\psi_{2,3}} = \frac{-\theta_2\sqrt{\beta_2}}{\sin\psi_{1,3}} = \frac{\theta_3\sqrt{\beta_3}}{\sin\psi_{1,2}} \tag{5.67}$$

Four corrector bump

The four corrector bump can be considered as the combination of two independent three corrector bumps **a** and **b** as shown in figure 5.60 Let θ_a, θ_b be the strengths of three

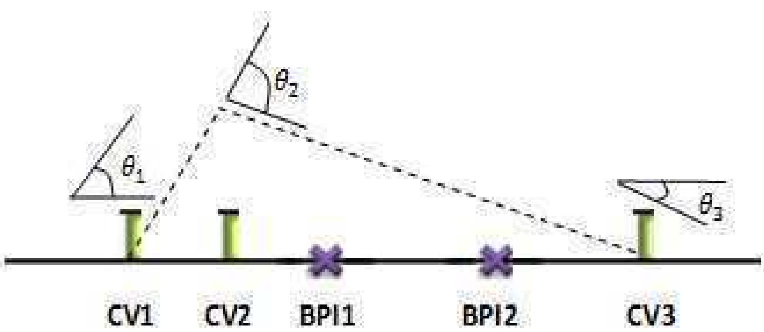


Figure 5.59: Three kicker bump scheme.

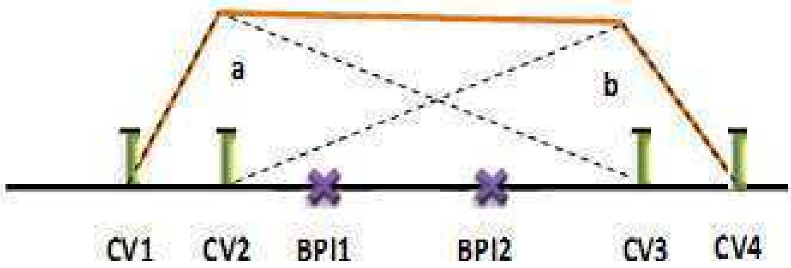


Figure 5.60: Four kicker bump scheme.

corrector bump **a** and bump **b** individually, then the corrector strengths $\theta_1, \theta_2, \theta_3, \theta_4$ will be given by Eq. (5.68) [18].

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = K \times \begin{bmatrix} \theta_a \\ \theta_b \end{bmatrix} = \begin{bmatrix} K_{1a} & 0 \\ K_{2a} & K_{1b} \\ K_{3a} & K_{2b} \\ 0 & K_{3b} \end{bmatrix} \times \begin{bmatrix} \theta_a \\ \theta_b \end{bmatrix} \tag{5.68}$$

For the bump to be closed the matrix coefficients K_{1a}, K_{2a}, K_{3a} and K_{1b}, K_{2b}, K_{3b} must satisfy the equation (5.67). The beam positions $[x1, x2]$ are then related to the bump strengths [θ_a, θ_b] through a 2×2 local response matrix R_l given by Eq. (5.69). The elements of response matrix R_l can be measured experimentally or calculated from the lattice function β and ψ at the BPI and corrector locations using Eq. (5.70) [18], where

ν is the tune and k_i and k_{cj} are the coefficient of sensitivity for BPMs and correctors, respectively.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = R_l \times \begin{bmatrix} \theta_a \\ \theta_b \end{bmatrix} \tag{5.69}$$

$$R_{ij} = k_i k_{cj} \frac{\sqrt{\beta_i \beta_{cj}}}{2\sin(\pi\nu)} \cos(\pi\nu - |\psi_i - \psi_{cj}|) \tag{5.70}$$

5.4.3 Organization of agents in multi-agent beam orbit control environment

The work in this section implements electron orbit control system for synchrotron radiation sources by means of cooperative working of multiple agents simultaneously to achieve the overall global goal of orbit control. The different types of agents are arranged in the form of layers as shown in the figure 5.61. The lower layer agents are the one that are directly interfaced with the accelerator machine components. There are two types of lower layer agents namely *beam line control agents* (depicted by DP-1 agents, DP-2 agent and DP-4 agent in figure 5.61) and *insertion device control agents* (depicted by ID-1 agent in figure 5.61). The lower layer agents encapsulate the behaviours necessary for controlling the beam position and angle for beam lines generated from dipoles and that from insertion devices. The behaviours are implemented through subsumption architecture to exhibit fast reactive cycle where in each execution cycle the agent reads the BPIs connected to it and according to the current state and desired state it calculates the corrector strengths using their skill and apply it to the correctors. Additional behaviour are incorporated at different priority levels to work in synchronization, seek help from trainer agents, assist trainer agents, seek help from fault assistance agents and to assist fault assistance agents. The lower layer agents are simpler in nature and can sense only the environment local to them i.e. they can only sense the beam position at the BPIs which are interfaced with the particular agent and similarly they can only take action on the environment through the actuators interfaced to them. Further the lower layer agents do not possess the capability to communicate and self collaborate with other similar lower layer agents. Thus the lower layer agents can only communicate with the middle layer or higher layer agents and that too with a slower rate than the rate of their execution cycle. In fact the lower layer agent exhibits a high degree of autonomy, performing actions in its environment based on information (sensors, feedback) received from the environment. The trainer agents and the fault assistance agents are situated at middle layer and are capable of gathering information from lower layer agents thus these agents can observe the relatively larger environment (but at a lower rate than what the lower layer agents can do) as compared to the individual lower layer agents. These agents implement the higher level of abstraction in their behaviour and thus are formulated to implement the system requirements that needs data collected from multiple lower layer agents such as training of lower layer agents for skill learning towards close bump generation and faulty device (sensor and actuator) identification. Middle layer agents are created on demand by orbit control agent whenever there is a request from lower layer agents for skill development and fault finding assistance. These agents, when become active, take control of overall beam orbit control activity and issue command to all lower layer agents to stop their ongoing activity and prepare them to participate in activities for higher level goal achievement. It then issues command to different participating agents according to their role in the team to interact with the environment in a coordinated manner and provide the necessary information required by middle layer agents in their decision making. These agents are implemented with modified subsumption architecture where the behaviour selection is done in a manner similar to the subsumption architecture explained in section 2 but here the behaviour functions are of much complex nature and implementation of some behaviours also utilizes the data storage between iterations. At the highest level there are three agents namely *monitoring agent, closed orbit distortion minimization agent* and *orbit control agent*. All of these agents are the main agents in this proposed multi-agent based electron orbit control system and always remain active. These are at the highest level of abstraction and control the overall orbit control job. The *monitoring agent* periodically (or when demanded) collects sensory data from lower layer agents and evaluates the individual agents performance. In case of performance degradation it informs the corresponding lower layer agent about its performance (*good, average, poor*and *bad*). The *closed orbit distortion minimization agent* contains the rough system model (previously measured systems response matrix) and on request it provides the optimized corrector strengths using the singular value decomposition SVD [36] method for achieving the desired golden orbit. This information is used for initialization at the time of starting of the overall orbit control system or intermittently if the overall organization of the agents is reinitiated. It also provides the tentative beam orbit for *orbit maintain* goal of different lower layer agents. The *orbit control agent* is responsible for creation of different lower layer and middle layer agents when required. It is implemented with a model based goal based modular architecture. It implements the graphical user interface for interaction with the operator. It reactively interacts with the operator to modify the beliefs in case of doubt, for example if *fault assistance agent* has executed all the related plans but still the faulty device is not found, or a timeout has occurred before a faulty device could be identified. It proactively interacts with the operator to seek suggestions about the present system performance to update the beliefs about the good beam quality at respective beam lines. It further accepts operator commands in the form of *attain position* and *attain angle* goals for the respective BL and send these goals to the respective *beam line control agents / insertion device control agents*. It also provides the updated

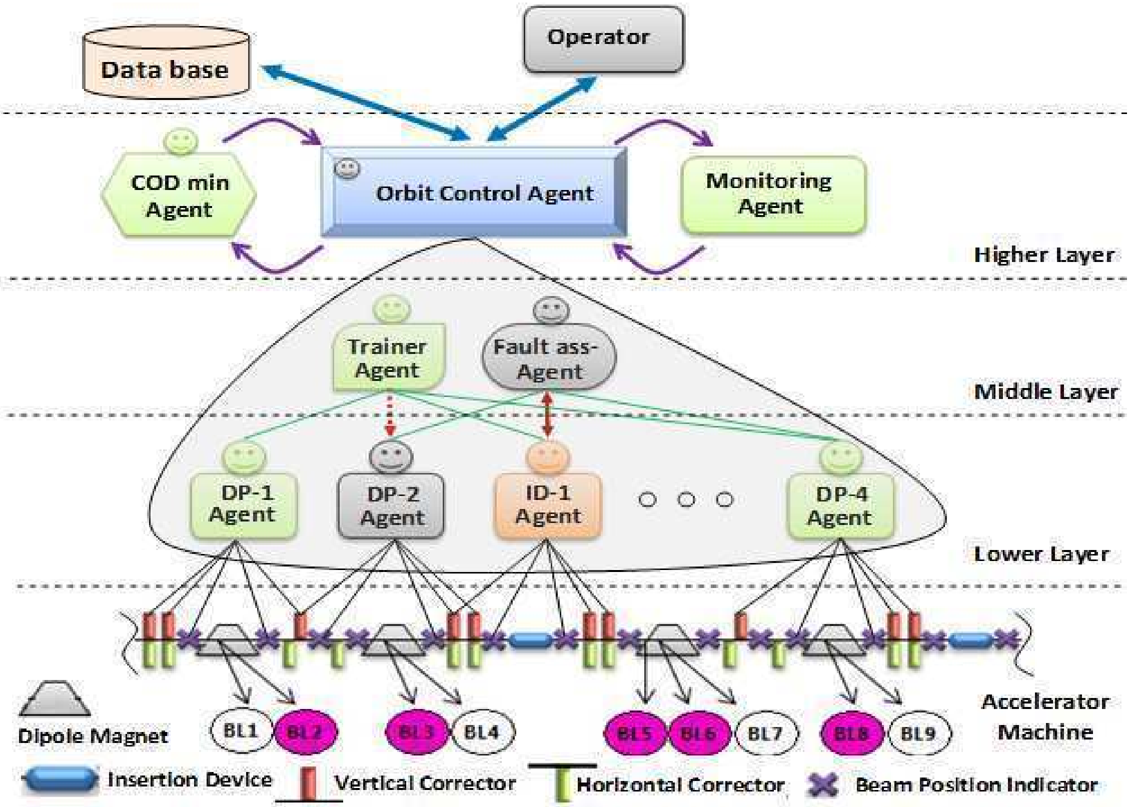


Figure 5.61: Organization of agents for multi-agent based beam orbit control system.

beliefs to lower layer agents whenever the beliefs are updated or agents are created. For example whenever a belief about the beam line operation ($[good: \ P, \theta]$, $[average: \ P, \theta]$, $[poor: \ P, \theta]$) changes the same is provided to the lower layer *beam line control agent* so that it can decide the optimum position and theta to be attained and maintained when the beam line is put to use by opening the safety shutter of that beam line. The graphical user interface served by this agent presents the pictorial representation of all the agents present in the multi-agent organization at any instant of time. The different execution states of agents are represented with the different colours in the mimic. The association of accelerator devices with the agent is shown with the lines connecting devices to the corresponding agents and the interconnection between different agents are represented with the coloured linkages between them where the line color shows the direction of message passing between collaborating agents. This live mimic of the overall system is very useful for debugging purpose.

Table 5.9: Beam line control agent design.

Percepts :: *BPI1, BPI2, CV1, CV2,CV3, SS1, SS2*

Beliefs :: *performance, self state, related bl,* $\{bl1 \Leftrightarrow [good : P, \theta], [average : P, \theta], [poor : P, \theta]\}$, $\{bl2 \Leftrightarrow [good : P, \theta], [average : P, \theta], [poor : P, \theta]\}$, $\{cv1 \Leftrightarrow [good : a, \Delta a], [average : a, \Delta a]\ \}$,$\{cv2 \Leftrightarrow [good : a, \Delta a], [average : a, \Delta a]\ \}$,$\{cv3 \Leftrightarrow [good : a, \Delta a], [average : a, \Delta a]\ \}$

Skill :: $cv1 : cv2 : cv3 : \theta1 : \theta2$

Constraints :: $\{cv1 \Leftrightarrow [free\ running : max, min], [restricted\ mode : max, min]\}$, $\{cv2 \Leftrightarrow [free\ running : max, min], [restricted\ mode : max, min]\}$, $\{cv3 \Leftrightarrow [free\ running : max, min], [restricted\ mode : max, min]\}$

Behavior :: *b0*: *work in synchronous way*
(*event : execute*)
Execute single cycle
Else
Remain ideal

Behavior :: *b1*: *remain ideal*
Do nothing, skip execution cycles

Behavior :: *b2*: *assist collaborators*
(*event : supply information*)
Perform the percept gathering
Perform the percept processing
Perform the supply information
(*event : action request*)
Perform the percept gathering
Perform the percept processing
Check feasibility(*requested action*)
Feasible $\rightarrow$ *perform action.*
$\neg$ *Feasible* $\rightarrow$ *message: action refused.*

Behavior :: *b3*: *serve average performance*
Calculate the error in θ
Generate attain goal

Behavior :: *b4*: *serve poor performance*
(*bl is high priority* $\wedge$ *in use*) $\rightarrow$ *wait*(*b1: remain ideal*)
(($\neg$*bl is high priority*)$\wedge$ *in use*) $\rightarrow$ *request training assistance* (*self state* $\leftarrow$ *training*))

Behavior :: *b5*: *serve bad performance*
Revert the last action on environment
Request fault assistance(*self state* $\leftarrow$ *fault assistance*)

Behavior :: *b6*: *serve attain goal*(*event:suggested* $\pm\Delta\theta$ *to bl*)
Optimised $\delta\theta \leftarrow f_{opt-\theta}(percepts, beliefs, \pm\Delta\theta)$
Optimised $[\Delta cv1, \Delta cv2, \Delta cv3] \leftarrow f_{opt-cv}(percepts, beliefs, skill, constraints, \delta\theta)$
Apply$[\Delta cv1, \Delta cv2, \Delta cv3]$ *to system*

Behavior :: *b7*: *serve maintain goal*
Calculate the error in θ
serve attain goal

Inhibition relation:: $b0, b2 \prec b5 \prec b4 \prec b6 \prec b3 \prec b7$

Table 5.10: Insertion device control agent design.

Percepts :: *BPI1, BPI2 , CV1, CV2,CV3,CV4 SS1*

Beliefs :: *performance, self state,* $\{bl1 \Leftrightarrow [good : P, \theta], [average : P, \theta], [poor : P, \theta]\}$, $\{cv1 \Leftrightarrow [good : a, \Delta a], [average : a, \Delta a]\}$, $\{cv2 \Leftrightarrow [good : a, \Delta a], [average : a, \Delta a]\}$, $\{cv3 \Leftrightarrow [good : a, \Delta a], [average : a, \Delta a]\}$, $\{cv4 \Leftrightarrow [good : a, \Delta a], [average : a, \Delta a]\}$

Skill :: $cv1 : cv2 : cv3 : \theta 1 : \theta 2$, $cv2 : cv3 : cv4 : \theta 1 : \theta 2$

Constraints :: $\{cv1 \Leftrightarrow [free\ running : max, min], [restricted\ mode : max, min]\}$, $\{cv2 \Leftrightarrow [free\ running : max, min], [restricted\ mode : max, min]\}$, $\{cv3 \Leftrightarrow [free\ running : max, min], [restricted\ mode : max, min]\}$, $\{cv4 \Leftrightarrow [free\ running : max, min], [restricted\ mode : max, min]\}$

Behavior :: $b0$: *work in synchronous way*
(*event : execute*)
Execute single cycle
Else
Remain ideal

Behavior :: $b1$: *remain ideal*
Do nothing, skip execution cycles

Behavior :: $b2$: *assist collaborators*
(*event : supply information*)
Perform the percept gathering
Perform the percept processing
Perform the supply information
(*event : action request*)
Perform the percept gathering
Perform the percept processing
Check feasibility(*requested action*)
Feasible $\rightarrow$ *perform action.*
$\neg$ *Feasible* $\rightarrow$ *message: action refused.*

Behavior :: $b3$: *serve average performance*
Calculate the error in P *and* θ
Generate attain goal

Behavior :: $b4$: *serve poor performance*
(*bl is high priority* $\wedge$ *in use*) $\rightarrow$ *wait*(*b1: remain ideal*)
(($\neg$*bl is high priority*)$\wedge$ *in use*) $\rightarrow$ *request training assistance* (*self state* $\leftarrow$ *training*))

Behavior :: $b5$: *serve bad performance*
Revert the last action on environment
Request fault assistance(*self state* $\leftarrow$ *fault assistance*)

Behavior :: $b6$: *serve attain goal*(*event:suggested* $\pm\Delta P$, $\pm\Delta\theta$ *to bl*)
Optimised $[\Delta cv1, \Delta cv2, \Delta cv3, \Delta cv4] \leftarrow f_{opt-cv}(percept, belief, skill, constraint, \Delta P, \Delta\theta)$
Apply$[\Delta cv1, \Delta cv2, \Delta cv3, \Delta cv3]$ *to system*

Behavior :: $b7$: *serve maintain goal*
Calculate the error in P *and* θ
serve attain goal

Inhibition relation:: $b0, b2 \prec b5 \prec b4 \prec b6 \prec b3 \prec b7$

Table 5.11: Trainer agent (for beamline control type student agent) design.

Percepts :: *sensors* $\leftarrow$ *assisting team members, actuators*$[CV1, CV2, CV3] \leftarrow$ *student agent*

Beliefs :: *performance* $\leftarrow$*student agent, self state, safe beam movement area, limit bump height*(L_{BH}), *bump step size* (B_{SS}), *coarse step size*(C_{SS}), *fine step size*(f_{SS}), *bpi noise band* (Δ_{noise}),$[BPM]_{Bump-sense}$,$[BPM]_{quality-sense}$

Skill :: $[cv1 : cv2 : cv3 : \theta 1 : \theta 2] \leftarrow$ student agent

Constraints :: $[bpi] \leftarrow$ *golden beam orbit,*$\{$*beam orbit: relaxed band*(Δ_R),*start band*(Δ_{start}),*stop band*(Δ_{stop}),*constraint band*$(\Delta_{const})\}$, *maximum iterations*(N_{max}),*useful performance limit* $(L_{Use-Per})$,*useful bump height limit* (L_{Use-BH}),*limit max corrector current change* (ΔI_{cv1})

Initialisation ::

Inform all agents about the training activity
Formulate coalition for the participating agents and distribute their role
wait for different agents to finish their work and come to desired state
Training starting point$\leftarrow f_{starting-point}$(*database, skill, system state*)

Behavior :: *b*0: *serve stop training request*
(*event : stop training*)
$Per_{value} \leftarrow f_{cal-per-value}$(*percepts, beliefs*)
if $((Per_{value} < L_{Use-Per}) \wedge (\mid [BPM]_{Bump-sense} \mid_{max} > L_{Use-BH}))$
  Release agents coordinating in training
  Release student agent (*performance*$\leftarrow$ *average*)
else
  Release agents coordinating in training
  Release student agent (*performance*$\leftarrow$ *poor*)

Behavior :: *b*1: *adjust bump height*
Increment CV_1 *by* C_{SS}

Behavior :: *b*2: *adjust step size*
((*increment*))
  Increment C_{SS} *by* f_{SS}
((*decrement*))
  Decrement C_{SS} *by* f_{SS}

Behavior :: *b*3: *goal achieved*
Release agents coordinating in training
Release student agent (*performance*$\leftarrow$ *good*)
Record the activity in database for future use

Behavior :: *b*4: *execute single training cycle*
[*Suggested moves*] $\rightarrow f_{suggested-moves}$(*percepts, beliefs, skill, constraint*)
For each suggested move
  Coordinate activity
    Student agent$\leftarrow$ *event: action request(suggested move)*
    Assisting team members$\leftarrow$ *event: supply information*
  $Per_{value} \leftarrow f_{cal-per-value}$(*percepts, beliefs*)
  Stop coordinate activity
Skill$\leftarrow f_{new-skill-value}$([*Suggested moves*],$[Per_{value}]$, *skill*)
Student agent$\leftarrow$ *update*(*skill*)

Inhibition relation:: $b0 \prec b1 \prec b2 \prec b4 \prec b3$

Table 5.12: Fault assistance agent design.

Percepts ::	*sensors ← assisting team members, actuators← faulty agent*
Beliefs ::	*performance ←faulty agent, self state, safe beam movement area, coarse step size*(C_{SS})*, fine step size*(f_{SS})*, bpi noise band* (Δ_{noise}), $[BPM]_{Bump-sense}$, $[BPM]_{quality-sense}$
Skill ::	$[cv1 : cv2 : cv3 : \theta1 : \theta2]$ ← faulty agent
Constraints ::	*[bpi] ← golden beam orbit,{beam orbit: relaxed band*(Δ_R)*,start band*(Δ_{start})*,stop band*(Δ_{stop})*,constraint band*(Δ_{const})*}, maximum iterations*(N_{max})*, limit max corrector current change* (ΔI_{cv1})
Initialisation ::	*Inform all agents about the fault assistance activity* *Formulate coalition for the participating agents and distribute their role* *wait for different agents to finish their work and come to desired state* *[Suspected device list]* ← $f_{suspecteddevicelist}$*(database, skill, system state)* *[Fault finding plan list]* ← $f_{planlist}$*(percepts, beliefs, skill, constraints, [suspected device list])*

Behavior :: *b*0:	*serve stop fault assistance request*
	(event : stop fault assistance)
	    Release agents coordinating in fault assistance
	    Release faulty agent (performance ← fault cannot be identified)

Behavior :: *b*1:	*goal achieved* *Release agents coordinating in fault assistance* *Release faulty agent (beliefs ← faulty (device ID))* *Record the activity in database for future use*

Behavior :: *b*2:	*execute single fault assistance cycle*
	    Coordinate activity
	        plan ← next plan (fault finding plan list)
	        faulty agent ← event: action request (plan (action))
	        Assisting team members ← event: supply information
	    $VAR_{Bool} \leftarrow f_{cal-per-value}$*(percepts, beliefs)*
	Stop coordinate activity
	    if (Fault equation (VAR_{Bool})
	        self state ← fault found (goal achieved)
	    if (Fault equation (VAR_{Bool})
	    else
	        if (all plans in the list executed)
	        serve stop fault assistance request

Inhibition relation::	$b0 \prec b1 \prec b2$

5.4.4 Agent design and implementation

As discussed in earlier sections the different types of agents in this multi-agent based orbit control system are proposed with different type of agent architectures ranging from simple subsumption architecture to high complexity model based goal based modular architectures. This section discusses the abstract implementation for each distinct type of agent that is needed for implementing the proposed multi-agent based orbit control scheme. Table 5.9 gives the percepts, beliefs, constraints and behaviours for implementation of beam line control agent. Table 5.10 gives the percepts, beliefs, constraints and behavior for implementation of *insertion device control agent*, where "percepts"are the signals, that are interfaced with the agent. The agent in the beginning of each execution cycle reads data from beam position indicators ($BPI1$ and $BPI2$), correctors ($CV1, CV2$ and $CV3$) and safety shutter ($SS1$ and $SS2$). It then checks for the firing condition of each behaviour and lists the behaviours triggered as per the current percepts. It then according to the inhibition relation selects the highest priority behaviour and executes the associated plan. The "beliefs"stores the system information needed by the actions in the plans for calculation of parameters such as the optimized angle deviation ($\delta\theta$) to be applied to the electron orbit and the calculated optimized corrector ($[\Delta cv1, \Delta cv2, \Delta cv3]$) values (using Eq. (5.68) and (5.69)) to be applied to the correctors to achieve the desired orbit deviation. The "constraints"mainly store the parameter limits needed for assisting in action selection or needed for some parameter calculations like the minimum and maximum limiting values for correctors under different operating modes (free running mode, restricted mode). The "Skill"stores the information about the effect on local environment that agents action produces. This information is used by the agent in calculating the suitable corrector current values needed for correcting the local beam orbit. The quality of agent skill is calculated using the Eq. (5.73). Where H_{Bump} and $\breve{p}_{value}$ given by Eq. (5.71) and Eq. (5.72) represent the generated hump height in the local region and the performance value calculated from percepts of the assisting team member agents. Skill level is an important parameter for correct operation of this scheme. It represents the agents ability towards correcting the electron orbit in region local to it while at the same time isolating the effect of this applied local correction from rest of the orbit. The degradation of the skill level causes the leakage of corrective action applied to the local orbit (orbit region which is in the direct influence of an agent) to the entire orbit. This will cause the interference between the simultaneously occurring actions of different agent on the electron beam orbit.

$$H_{Bump} = f_{Bump}(Percepts, beliefs) \tag{5.71}$$

$$\breve{p}_{value} = f_{cal\ per\ value}(Percepts, beliefs) \tag{5.72}$$

$$Skill\ level = \frac{\left(\frac{H_{Bump}}{\breve{p}_{value}}\right)}{\frac{L_{BH}}{\Delta_{stop}}} \tag{5.73}$$

The behaviours $b0$ to $b7$ broadly provides the agent ability to correct the beam angle ($\pm\Delta\theta$) at the beam line in beam line control agent and beam angle and position ($\pm\Delta\theta$,$\pm\Delta P$) in the *insertion device control agent* case, maintain the beam position and angle at BL and ID, help other agents in training and fault finding activities and handling the cases when agents performance degrades due to degraded agent skill level under system dynamics.

Table 5.11 gives the design of trainer agent (for *beam line control agent* type student agent). At the time of trainer agent creation, the *orbit control agent* associates the variables in percepts, beliefs, skill and constraints according to the student agent designated to it for imparting the training. At the time of starting, the trainer agent informs all the lower level agents about its intention to conduct the training for its student agent. It then distributes the role to each participating agents that it has to play as team member in training. It then waits till all the participating agents come to a predefined state before proceeding for the start of training activity. The behaviour $b0$ to $b4$ implements the constraint gradient based reinforcement skill learning activity discussed in detail in next subsection by systematically coordinating the activities of distributed lower layer agents through sequential instruction issuing to individual agents such that the actions of all the participating agents become synchronized towards training. Under such team formed condition the complete team can be viewed as a single unit performing the learning activity. After the completion of learning (when either the skill to sufficient level is learned or the activity is stopped due to high priority goals) the trainer agent updates the learned *skill* along and *performance* value to student agent and releases all the participating agents to work independently.

Table 5.12 gives the design of fault assistant agent. Similar to the trainer agent case this type of agents are also created by *orbit control agent* and variables in percepts, beliefs, skill and constraints are initialized according to the suspected faulty agent designated to it for fault finding. Similar to the trainer agent case at start time fault assistance agent informs all the participating agents about the intended fault finding activity and prepares them for assistance by designating their role in the activity. Based on the historical data, system believes, and systems state it then prepares the suspected device list. Using this suspected device list it than prepares the fault finding plan list for all the suspected devices. Behaviour $b0$ and $b1$ deals with the cases of stopping of fault assistance activity either due to the stopping request from a high priority goal or if the faulty device has been found or if the entire selected fault finding plans have been tried and agent could not associate the fault with any of the suspected devices. Behaviour $b2$ performs the function of coordinating the activities of participating lower level agents so that the fault finding team of agents can be thought of as a single unit. This behaviour selects one fault finding plan from the list. It than directs the plan *actions* to actuating agent to perform the perturbation to environment through actuator and collects the changed environment information supplied from assisting team member agents. It then generates the Boolean variables from the measured parameters. The Boolean variables are then put in the predefined logical equations to conclude that the device is faulty or ok.

Table 5.13: Monitoring agent design.

Percepts ::	*sensors* $\leftarrow$ *lower level agents, monitored agent* $\leftarrow$ *current agent*
Beliefs ::	*performance* $\leftarrow$ *current agent, self state*
Constraints ::	[*bpi*] $\leftarrow$ *golden beam orbit,*{*beam orbit: good performance band*($\Delta_g ood$)*,average performance band*($\Delta_{average}$)*, poor performance band*(Δ_{poor})*, bad performance band*(Δ_{bad})*, constraint band*(Δ_{const})}

Behavior :: *b*0:	*serve agent performance evaluation request*
	Inform all agents about the performance evaluation activity
	Formulate coalition for the participating agents and distribute their role
	wait for different agents to finish their work and come to desired state
	Coordinate activity
	    Monitored agent $\leftarrow$ *event: action request (suggested move)*
	    Lowest layer agents $\leftarrow$ *event: supply information*
	    $Per_{value} \leftarrow f_{cal-per-value}(percepts,\ beliefs)$
	Stop coordinate activity
	$(Performance \leftarrow f_{performance}(Per_{value}, constraints)\ )$
	Update (*current agent* $\leftarrow$ *Performance*)

Behavior :: *b*1:	*serve reactive agent performance evaluation* *event: evaluate performance (agent)* *current agent* $\leftarrow$ *agent* *serve agent performance evaluation*
Behavior :: *b*2:	*serve proactive agent performance evaluation* *current agent* $\leftarrow f_{suspected-agent}$(*database, percepts, policy*) *serve agent performance evaluation*
Inhibition relation::	$b0 \prec b1 \prec b2$

Table 5.13 gives the design for monitoring agent. The *monitoring agent* through its behaviour $b0$ performs the job of evaluating the performance of the lower level agents. It does so by coordinating the team of agents where one agent performs the demo action on environment and all other distributed team members supply the information about the effect of this demo action on the environment. This is used to evaluate the performance of the agent. The performance of the agent is evaluated in four fuzzy levels; *good, average, poor* and *bad*. The behaviour $b1$ and $b2$ provides the mechanism of engaging into performance monitoring activity through reactive mechanism (i.e. when *orbit control agent* requests for performance evaluation of some lower level agent) or through proactive mechanism (i.e. the policy based performance evaluation, both routinely evaluation as

well as history based evaluation can be used).

Table 5.14: COD agent design.

Percepts ::	*sensors* $\leftarrow$ *lower level agents*
Events::	*get new orbit, get optimised corrector settings*
Beliefs ::	{*system model:horizontal response matrix* (RM_x), *vertical response matrix* (RM_y) }, { *system database: machine database* (*static, dynamic*)}
Constraints ::	[*bpi*] $\leftarrow$ *golden beam orbit*,{*beam orbit: good performance band*($\Delta_g ood$),*average performance band*($\Delta_{average}$), *poor performance band*(Δ_{poor}), *bad performance band*(Δ_{bad}), *constraint band*(Δ_{const})}, { $cv_i \Leftrightarrow$ [*free runing: max, min*], [*ristricted mode: max, min*]}
Capability :: $C0$:	*get new golden orbit* *sector orbit into restricted and relaxed regions* [*restricted, relaxed*] $\leftarrow f_{sector-orbit}$(*golden orbit, percepts, beliefs*) *new golden beam orbit*$\leftarrow f_{optimised-orbit}$([*restricted, relaxed*], *constraints*) *Dispatch* $\leftarrow$*new golden orbit*
Capability :: $C1$:	*get optimised corrector settings* *sector orbit into restricted and relaxed regions* [*restricted, relaxed*] $\leftarrow f_{sector-orbit}$(*golden orbit, percepts, beliefs*) *Optimised corrector settings*$\leftarrow f_{optimised-corr}$([*restricted, relaxed*], *constraints, policy*) *Dispatch* $\leftarrow$*optimised corrector settings*

Table 5.14 gives the *COD agent* design. This agent uses the system model in its beliefs and exhibits the complex capabilities $C0$ and $C1$ that provides this agent the functionalities to generate the optimized golden orbit and the optimized corrector settings using the global orbit correction schemes like SVD, sliding bump or combined depending upon the policy selected. These values are used by the *orbit control agent* to initialize the lower level agents at the time of their creation.

Table 5.15 gives the design for *orbit control agent*. This agent is a higher level agent and mainly controls the overall local beam orbit control job. It provides the man-machine interface for operators to interact with machine in normal operations when operators directly interacts with the system to start/stop the system, increase/decrease beam angle and position at particular beam lines, declares the particular devices such as BPIs or correctors as faulty devices (i.e. the devices that are not to be used in normal operation), declares the sensitivity of experiment being performed at the beam lines ([angle::

high, medium, low], [position:: *high, medium, low*]), declares the upcoming beam lines, declares the present orbit to be used as the golden orbit for the system. This agent contains the complete system model in its beliefs. The knowledge base is divided into two parts- machine specific knowledge base and accelerator physic specific knowledge base. The machine specific knowledge base is further divided into static knowledge base and dynamic knowledge base. The static knowledge base contains the machine specific information which does not change according to the systems state of operation like the sequence of components, location of components, coefficients of different components etc. The dynamic data base contains the information that is dependent on the systems state of operation like power supply on/off status, BPI enabled/disabled state etc. The capabilities $C0$ to $C5$ implements the functionalities for generating the *beam line control agent, insertion device control agent, trainer agent, fault assistance agent,* and handling of proactive events, user generated events and machine generated events. Figure 5.62 shows the detailed architecture for this agent.

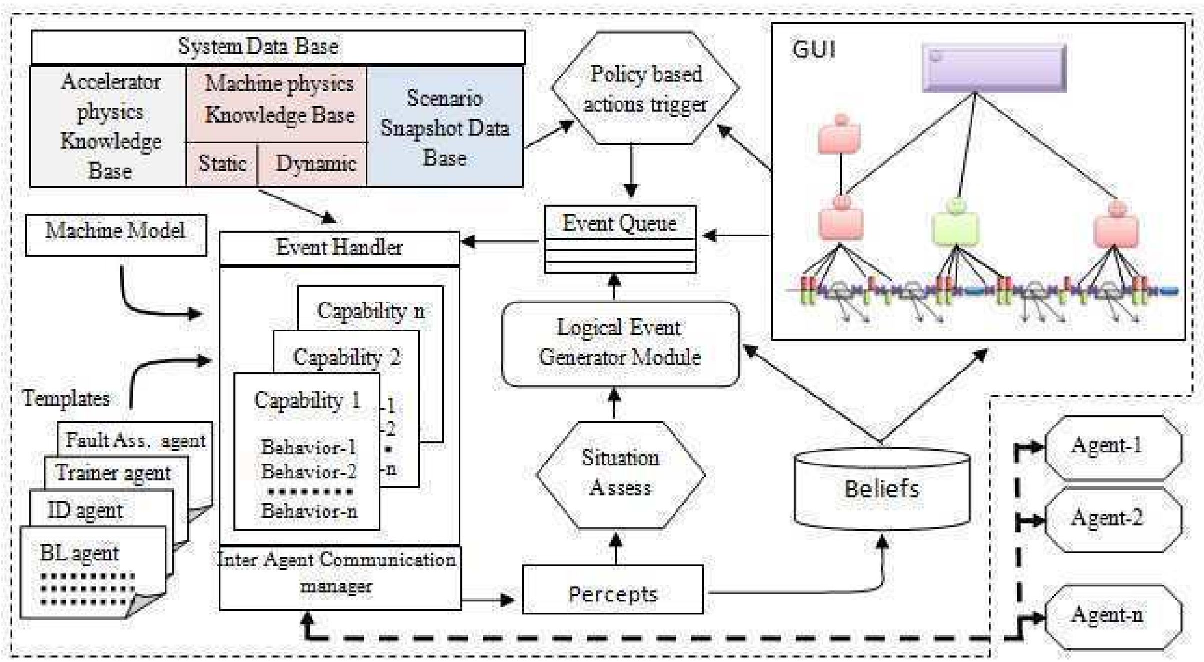


Figure 5.62: Architecture of Orbit control agent.

Table 5.15: Orbit control agent design.

Percepts ::	*sensors* $\leftarrow$ *lower level agents*
Events::	*user generated events*
	e_1*:Start operation*
	e_2*:Stop operation*
	e_3*:Increase/decrease* θ *at* BL_i *or* ID_i
	e_4*:Increase/decrease* P *at* ID_i
	e_5*:Donot use* BPI_i
	e_6*:Donot use* CV_i/CH_i
	e_7*:Experiment at* BL_i *is sensitive*
	e_8*:*BL_i*is upcoming BL*
	e_9*:Force, present orbit as golden orbit*
	system generated events
	e_{10}*:Beliefs update*
	e_{11}*:Agent died*
	e_{12}*:Agent not responding*
	e_{13}*:Fault connot be identified*
	e_{14}*:Seek operator suggestion request*
	e_{15}*:Agent status updated*
	e_{16}*:Updated system mimic*
Suggestions seek from operator ::	
	evaluate overall performance of the machine (good/average/poor)
	evaluate present orbit (good/average/poor)
	evaluate position at BL_i *(large hi/ slight hi/good/ slight low/ large low)*
	evaluate BPI_i *performance (good/average/poor/faulty)*
	evaluate CV_i/CH_i *performance (good/average/poor/faulty)*
	clarify doubt, is the BPI_i *faulty (normal/faulty)*
	clarify doubt, is the CV_i/CH_i *faulty (normal/faulty)*
Beliefs ::	{*system model:horizontal response matrix* (RM_x), *vertical response matrix* (RM_y) }, { *system database: machine database (static, dynamic)*},{ *accelerator physics database, scenario snapshot database*}
Capability ::	$C0$: *generate beam line control agent*
Capability ::	$C1$: *generate insertion device control agent*
Capability ::	$C2$: *generate trainer agent*
Capability ::	$C3$: *generate fault assistance agent*
Capability ::	$C4$: *generate proactive events*
Capability ::	$C5$: *handle events* (e_i)

5.4.5 Constraint gradient based reinforcement learning

The reinforcement learning (RL) is the common learning scheme used by the agent based systems. In this scheme an agent selects and engages in behaviour with respect to sensory inputs obtained from sensors. As a result the learning is performed by repeating a cycle in which a reward from the environment and sensory input for the next state is given. The Q-Learning and Temporal-Difference (TD) Learning are two common RL methods; the former learns the utility of performing actions in states, while the latter usually learns the utility of being in the states themselves. But apart from learning the system states and the action relationships in some situations like the local orbit control in accelerators, there is also a need to identify the local environment's behavioural representation that is independent of the action selection by the agent. This can be thought of as the local environment representation (local environment model) in agents beliefs that does not affect the agents action selection decision but directly affects the outcome of the agents actions on the environment. For example if one of the *"beam line control agent"* decides to correct the beam angle at the respective beam line and engages in behaviour "serve attain goal (event: suggested $\pm\Delta\theta$ to bl)"to attain the suggested $\pm\Delta\theta$ value. Apart from action sequences and all other relevant things the outcome of this behaviour also depends on the correctness of the calculated corrector strengths and this calculated corrector strengths depends on how well the local environment is modelled by the agents skill. For beam line control agents, the skill is represented by the set [*Skill::CV1:CV2:CV3:BPM1:BPM2*] where the *CV1,CV2,CV3* are the corrector magnet set values and BPM1, BPM2 are the beam position monitor values for the stated corrector values. Now for the case of agent with a good skill level the skill variables (*CV1, CV2 and CV3*) will be having a definite ratio that satisfies the Eq.5.67 to generate the non leaky closed bump. Thus for changed accelerator environment this skill will not be able to generate the close bump and the amount of bump leakage will directly depend on the amount by which the system dynamics has changed the environment. Thus in such cases the relearning of the skill is required by the agent. As in this case the agent learning is not concerned with the action selection but is linked to the local system identification, we are referring this as the agents skill learning and as in the highly distributed environment the agent itself does not contain the sufficient resources to carry out this learning on its own. The coordinated training of the student agent is carried out by the trainer agents who suggests the moves (actions) to the student agent, evaluates the outcome of this move and depending upon this outcome decides the next move and new skill level. The next move is decided, based on the gradient of the cost value (reward value). For simplicity we will view the distributed multiple agents performing the coordinated learning activity as a single unit.

Some systems like the accelerator orbit control in SRS, demand that during the training the agents action performed towards learning activity should not perturb the electron beam at nearby or other high priority beam lines to a level such that the experiment data gets affected or the complete beam gets killed. To overcome this problem the constraint based skill learning algorithm is proposed here.

Complete skill learning by the agent in the presence of noise requires that the system should be perturbed such that, at least a minimum level of bump height at the desired position is generated while keeping the disturbance in the beam at all other locations confined within the limits. Contrary to this in actual system such desired bump can only be generation while fulfilling the constraint requirements only if the correct model of nearby environment is available (i.e. the fully trained skill set is available). So in order to solve this paradox the proposed algorithm in the beginning (when skill set is initialized to all zero or some pre-known skill set) starts the system identification with very small perturbation done to the environment through first corrector (CV1) such that the beam positions at constraint locations ($\breve{p}_{value}$) remains within the constraints limit (Δ_{const}) i.e. the Eq. (5.74) is satisfied. It than keep on increasing the CV1 values in steps till the $\breve{p}_{value}$ just crosses the start limit band (Δ_{start}) (i.e. the Eq. (5.75) is satisfied). It then tries various combinations for CV2 and CV3 and selects the one based on the steepest gradient method that reduces the Per_{value} the most.

$$\breve{p}_{value} < \Delta_{const} \tag{5.74}$$

$$\breve{p}_{value} > \Delta_{start} \tag{5.75}$$

$$H_{Bump} \geq L_{BH} \tag{5.76}$$

$$\breve{p}_{value} \leq \Delta_{stop} \tag{5.77}$$

It then updates the skill level according to the best action performed in the iteration cycle. This process of refining the skill continues till the $\breve{p}_{value}$ reduces below start limit (Δ_{start}). It then use this partially learned skill and calculates new values for CV1, CV2, CV3 such that it satisfies the Eq. (5.75). It then again starts the skill refining process using gradient based method. This multi step alternate bump height increasing and partial skill learning process continues till the desired bump height (H_{Bump}) is achieved. Finally after the desired bump height is reached the skill refining process ends when the disturbance in beam orbit in the constraint region ($\breve{p}_{value}$) reduces below stop limit (Δ_{stop}). This overall process for the proposed algorithm is shown by flowchart in figure 5.63.

5.4.6 Early implementation simulation results with INDUS-2 model and discussion

The agents are developed according to the proposed scheme and the simulation trials are done for training of different beam line control agents and insertion device control agents so that they learn the sufficient skills for controlling the beam lines. Figure 5.64 shows the different beam positions at different BPMs throughout the INDUS-2 ring during one of the lower level agent ("DP3 agent") training using the proposed constrained gradient based reinforcement method. Figure 5.65 shows the different beam positions at different BPMs throughout the INDUS-2 ring for training of same agent using the general gradient

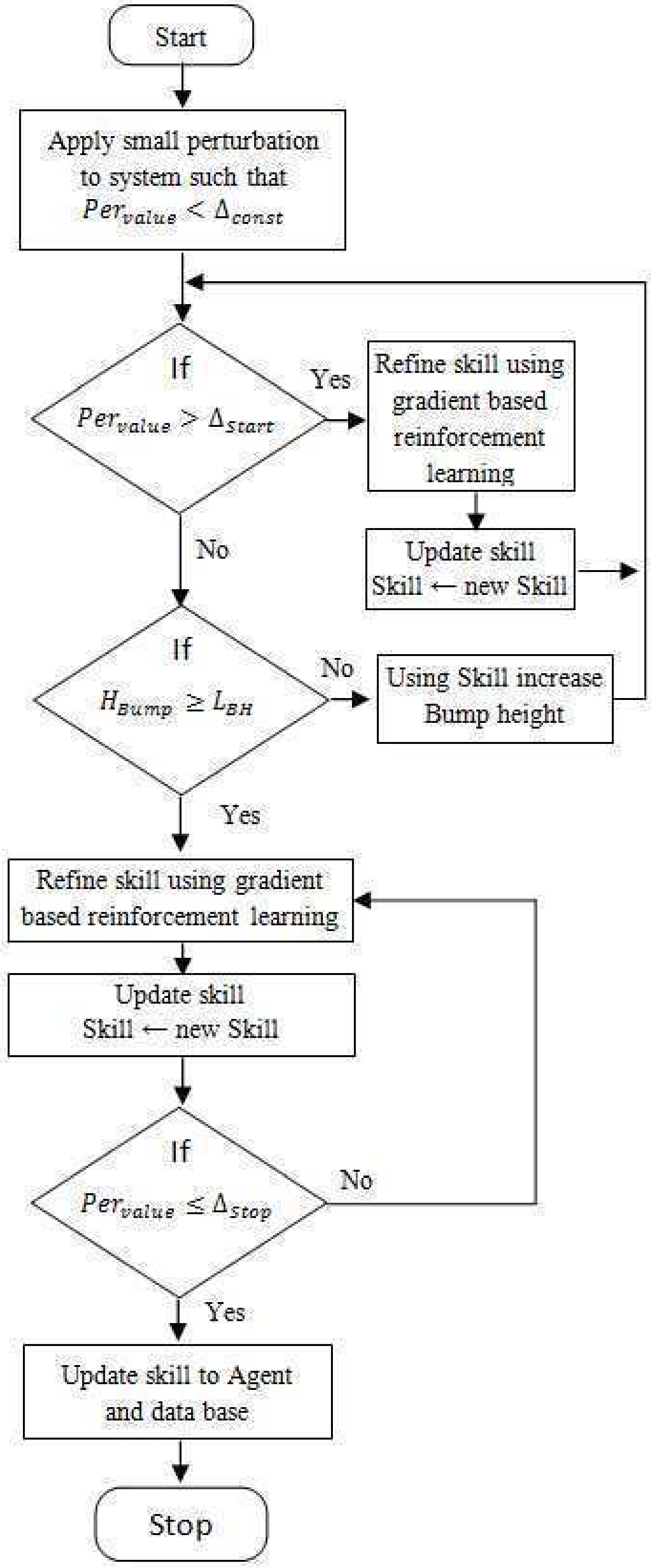


Figure 5.63: Flow chart for the constrained gradient based reinforcement skill learning.

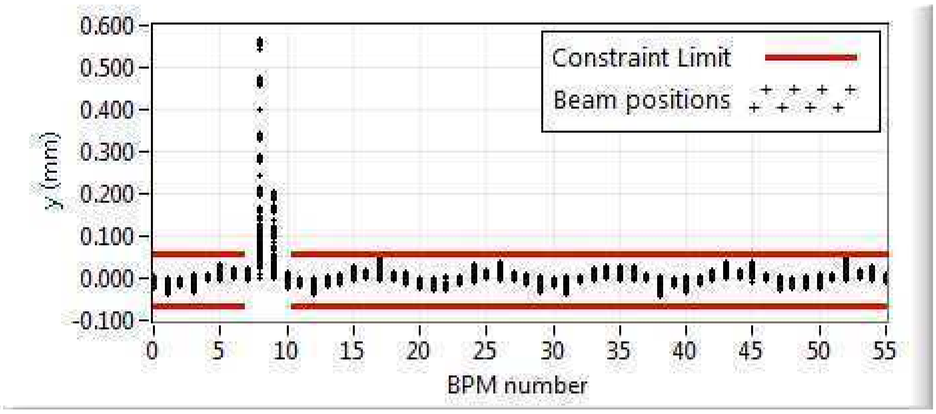


Figure 5.64: Beam positions on different BPMs during the skill learning using the constraint gradient based reinforcement learning.

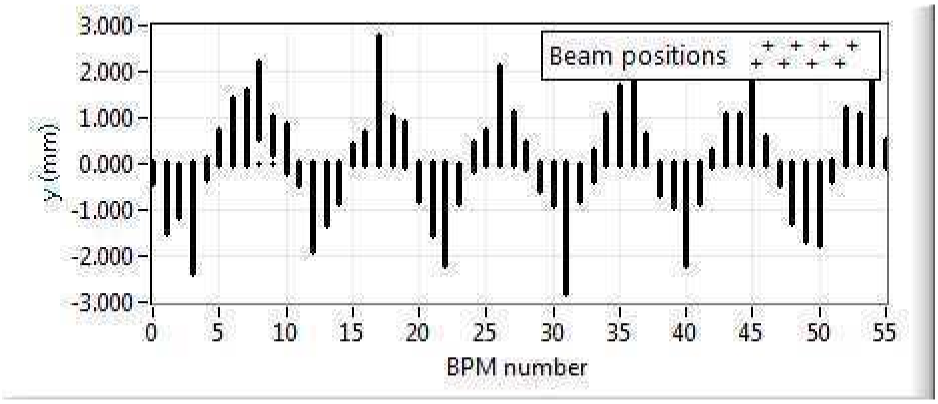


Figure 5.65: Beam positions on different BPMs during the skill learning using the general gradient based reinforcement learning.

based reinforcement method. Figure 5.66 shows the different skill states acquired by the agent during the training cycles for which the beam positions are shown in figure 5.64 and 5.65. Figure 5.67 Shows the skill states acquired by the agent ("DP3 agent") for the case of agent imparted with different initial skill levels. Figure 5.68 shows the skill learning by the agent ("DP3 agent") for the case of machine operating point changes due to different error introduced in the defocusing quadrupole (QD) family. From Figure 5.64 one can clearly see that for the agent skill learning process using the proposed constraint gradient based reinforcement learning method, is well capable of imparting the same skill level to the agent as general gradient based learning method while traversing almost the same

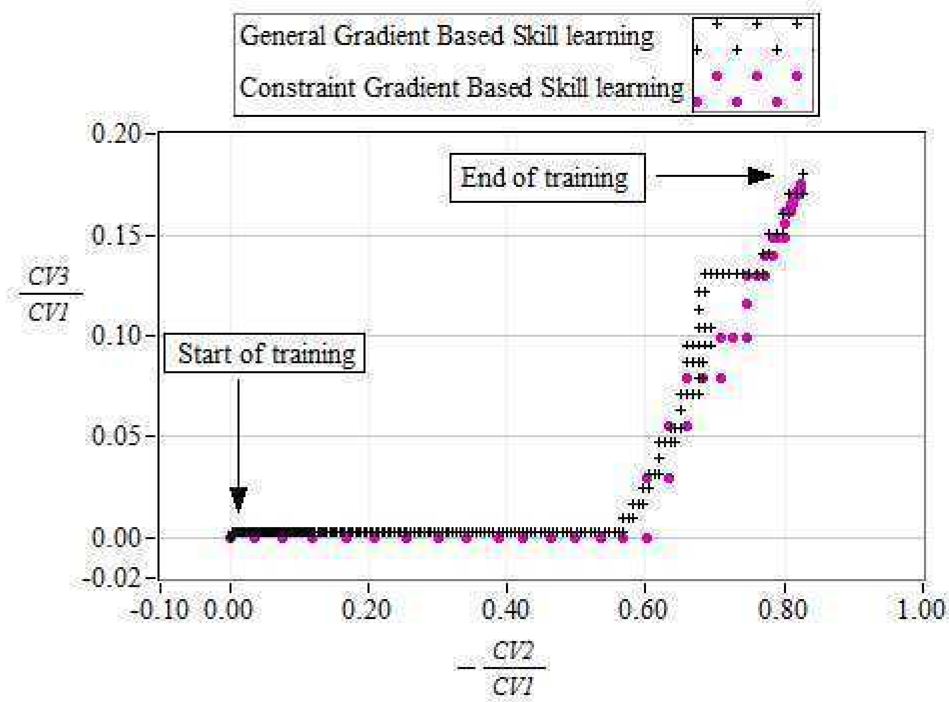


Figure 5.66: Skill states acquired by the agent during the training cycles for which the beam positions are shown in figure 5.61 and 5.62 For the case of starting skill level ([cv1:cv2:cv3:bpm1:bpm2] = [0.000:0.000:0.000:0.00:0.00]). The final learned skill state by general gradient based skill learning is [1:-0.828:0.180:0.247:0.088] and the final learned skill state by proposed method is [1:-0.824:0.175:0.248:0.088].

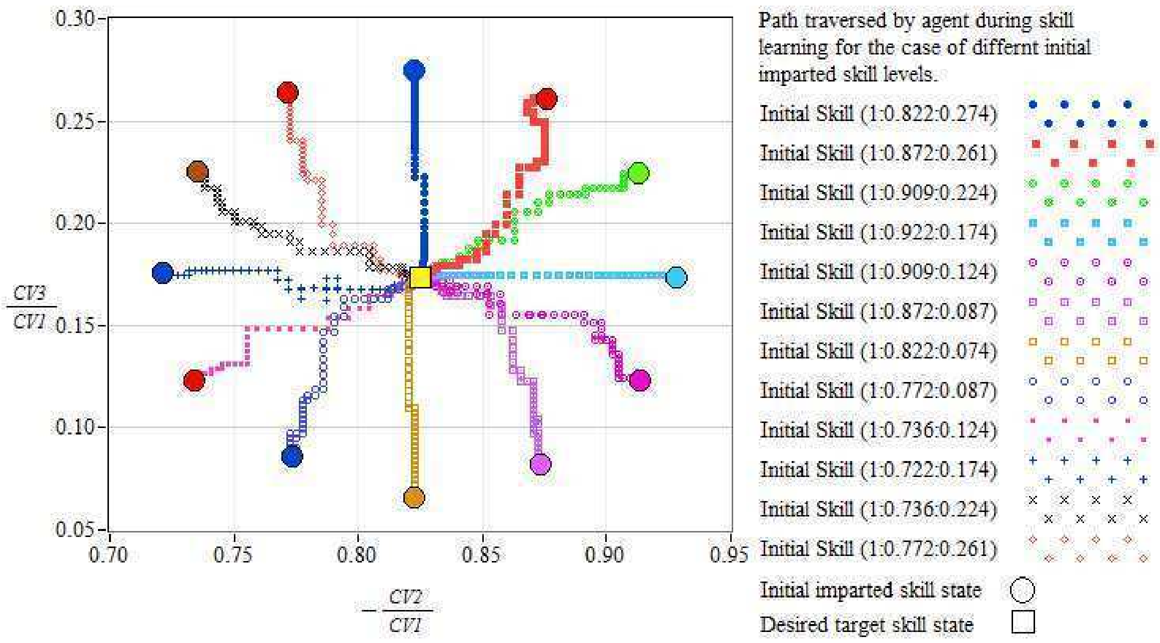


Figure 5.67: Skill states acquired by the agent ("DP3 agent") for the case of agent imparted with different initial skill levels.

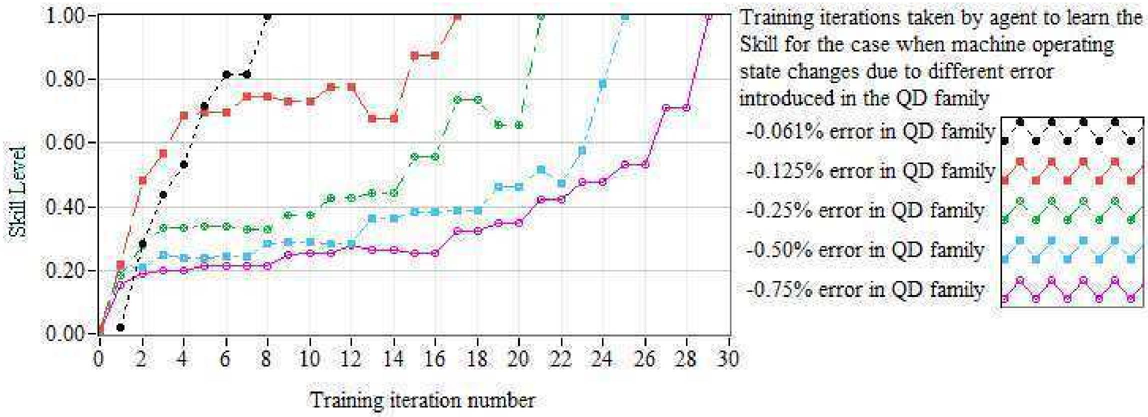


Figure 5.68: Skill learning by the agent ("DP3 agent") for the case of machine operating point changes due to different error introduced in the defocusing quadrupole (QD) family.

path through skill set. Further it can be seen from figure 5.64 that the proposed skill learning method is successful in limiting the beam perturbation in the constraint region (BPM0 to BPM6 and BPM10 to BPM55) within the constraint limits ($\pm 70\mu m$). Whereas the general gradient based method perturbs the beam to very large values ($\approx \pm 2.5mm$) for similar skill set learning thereby making the beam lines unusable in this region at the time of training. From figure 5.67 one can see that for the case when agent has acquired some skill (i.e. the agent is having some initial skill learned according to some previously occurring state) and then in run time if the system changes to some other operating state then the agents skill learning using the proposed method is well capable of converging to desired new skill from different initial skill sets. Further from figure 5.68 one can see that the training iteration cycles required by the agent to acquire this new skill set (from the previously trained state) is limited to less than 30 cycles for the worst case considered of the parameter variations up to 0.75% . Thus for the practical case if the agent decides to go for training to improve the performance, only the related beam line will be affected for the period less than a minute.

5.4.7 Summary and conclusion

Electron beam position and angle are controlled in SR sources for obtaining and maintaining the maximum photon flux at desired sample location in experimental SR beam lines. Towards this a novel multi-agent based operator support and beam orbit control scheme is formulated for improving the overall reliability and beam availability to the SR users. The design of different types of constituting individual agents is presented. This scheme is also helpful in improving the available beam quality to users through prioritized optimization for the cases when multiple beam lines demand the beam correction simultaneously. A novel gradient based cooperative multi-agent based reinforcement skill learning scheme is also presented for imparting training to lower level agents in case the skill level towards orbit control of lower layer agents degrades. The effectiveness of the proposed scheme is shown through simulation results obtained with the early implementation of the scheme with Indus-2 model. Assuming that under dynamic environment the agents can successfully maintain their up to date skill levels with the presented skill learning method, then this distributed skill information can be further utilized for extracting the systems model which can be used to improve the system performance.

Chapter 6

Summary and future direction

The study was set out to explore the concept of distributed intelligent agent based control system for real time control of accelerator and has identified the accelerator control system architectures, specific subsystem control requirements, agent architectures, agent based cooperative control scenarios, agent representations and multi-agent distribution over different accelerator control system layers for effective resource utilisations. The study has also sought for modelling of different accelerator subsystem components from controls perspective and overall accelerator control system simulation to know whether the agent based control can result in effective accelerator tuning particularly for the case of disturbances observed at the beam source, sensor failures and orbit correction strategies in synchrotron radiation sources. The general theoretical literature on this subject and specifically in the context of synchrotron radiation sources control systems is not sufficient to answer some of the vital questions within this scope. The study sought to answer three of these questions:

- Which agent architectures are more suited for transforming the existing accelerator subsystems into intelligent agents using the existing control system infrastructure?
- How can we represent such intelligent agents suitable for integration with the accel-

erator control system?

- How to distribute different agents over the multi-layer accelerator control system framework for exercising the multi-agent control approach?

The answer to these questions are important for exercising the AI based operator support and machine control system that are required for the reliable, maintainable and easy operability of future accelerator projects in the present trend of increasing operation parameters and growing size of facilities.

6.1 Summary

To withstand the current demand for increase in the energy of charged particles the future accelerator machines needs artificial intelligence based advanced control method deployment at different system/subsystem levels. Moreover with the current technological developments in the control system support hardware the existing accelerator facilities will also explore the advantage of using agent based control for improving the system operability and reliability. From implementation point of view the first research question address the issues related with the optimised dispersion of abstract agent representation at subsystem levels based on the existing control system infrastructure.

1. Which agent architectures are more suited for transforming the existing accelerator subsystems into intelligent agents using the existing control system infrastructure?

a. Modular architectures: The present accelerator control and operation methods group the system parameters as per the accelerator subsystems moreover many times these share different development platforms thus the modular architectures are the best suited at the subsystem level agent development. For the existing accelerator facilities this will further minimise the switch-over time during intermittent deployment and testing of such schemes through interfacing modules that will selectively isolate the existing man machine interface commands from that of agents. Also from software programmers point of view the subsystem level agent goals with rational action sequence can be easily collected from the machine operators and system experts with clear and concise information thus supporting the speedy development. Although in software theoretically all the agent architectures discussed in chapter 2 can in one or the other form can decompose the tasks in modules with sufficient granularity but from the perspective of this study the agent architectures such as subsumption architecture and modular architecture are the one that directly comes in this category.

b. Model based decision making: The accelerator environment being complex natured (derived-parameter handling and multiple modes of operation) imposes the restriction on the decision making process of agents to encompass the physics model of subsystems for generating the intermediate derived parameters thus the agent architectures supporting the direct inclusion of system model such as the modular architecture is the suggested one for the subsystem level of agents. Moreover from controls point of view(Repeatability and operability) it is not always possible to derive the useful system model from the available physics model of the subsystem to an extent that the correct decision can be taken towards machine tuning. For such cases the on-line identification of the subsystem over the control parameters in networked control system configuration can be a solution.

c. Inter-agent coupling and integration with accelerator physics tools: The cooperative optimisation tasks in multi-agent organisations for accelerator control needs the information exchange at subsystem levels. Further the information needed by the coaliating agents may not be directly available through sensors and many times is needed to be derived through subsystem model thus the agent development framework must integrate to some of the available accelerator modelling tools like MAD and AT.

d. Multi level goal and information distribution: The multi-agent based cooperative optimisation and control needs the information and goal distribution between different agents and on a broader perspective these agents of varying complexity may be found distributed on different accelerator control system layers thus the agent goals and meta data is needed to be travelled trough different control system layers. Thus the agent framework supporting the direct coupling between inter and intra agent modules is desirable.

After formulation of the abstract requirements for such agents the next question is.

2. How can we represent such intelligent agents suitable for integration with the accelerator control system?

a. Use independent modules as building block: The simplest building block used for building agents in this study are formulated by combining the message based communication interface called as blocks postman with the blocks body that comprises of a continuous loop performing the event serving one at a time taken from message queue. This design is selected because of ease of implementation in multi platform system. As the message parser and socket based postman can be implemented in any language supporting the case construct and socket programming API. The event serving loop provides the behaviour invoking mechanism whereas the behaviour can be implemented using any platform.

b. Postman post-office based centralised inter and intra agent communication: In the distributed accelerator control environment the post-office based communication infrastructure is proposed with the custom agent communication language framework for event and data transmission across different agent. Being open to expansion it relieve the agents from the burden of managing the message routing as in this case each agent is required to only know the address of post-office. further more this scheme also automatically handles the agent movement across different control system organisational layers and at the same time serves as the valuable tool for debugging and rational action sequence linking & verification.

c. Separate concurrent and sequential execution loops for module execution Although each agent will have some specific implementation details according to the domain and agent type in accelerator environment but broadly the agent structure is comprised of three parts a) the agents postman: the one who handles the agents communication in accelerator environment, b) concurrent module execution loop: the execution loop which handles the independent behaviours and c) The sequential module execution loop: the execution loop that handles the dependent behaviours.

with the development of agent framework the next question is about their distribute and goal assignment for the multi-agent based control scenario.i.e.

3. How to distribute different agents over the multi-layer accelerator control system framework for exercising the multi-agent control approach?

a. Place small response time demanding agents at lower layers and large response time demanding agents at higher layers: With every increase of layer in the control system of accelerators the distance of actors(system controlling agents) from machine increases. thus increasing the overall system response time. Therefore the policy is to place the small response time demanding agents towards lower layers.

b. Place simple agents at lower layers and complex agents at higher layers: For agents both speed and intelligence can not be increased simultaneously, the limitations arise due to the limitation on available resources. In the multi-layer architectures of accelerator control systems the computational resources increases with every increase of system layers, This advocates the policy of placing large computation demanding agents towards higher layers.

With the developed agent framework the thesis further investigate the possibility of performance enhancement in the accelerator control through agent based control simulation. As a case study the basic model of the typical synchrotron radiation source facility is developed with the interface similar to the actual machine interface. Particularly for the synchrotron radiation sources application wise the agent based realtime control is studied towards automated tuning of transport line to maintain the injection current above a certain threshold for the case of beam movement at the source. The results show that with the inclusion of model assessed genetic algorithm based plans(MAGAP) agent can successfully derive the beam parameters at source in real time using measurement data. In a feedback manner from these derived beam parameters the agent was successful in calculating the control settings again using different MAGAP. The results supports the Schirmer et al.[103] findings for GA based methods suitability in restoring the beam position at BPI's. further to this the study shows with this method injection current can be maintained above 90% value for the practical case of beam movement bounds observed during cathode life cycle, moreover this is found true for the different booster settings thus the proposed method when implemented will assist operators in fast machine tuning on day to day basis. The study shows this improvement in injection current can not be exploited fully by the machines that uses interceptive type of BPI's in transport lines. This occurs because of non availability of beam during the measurements using such BPI's. For such cases the multi-agent based approach with model based tracking plans are investigated. Study shows this approach can also handle the cases of limited sensor (BPI) and Actuator (Correctors) failures thus improving the overall system reliability and robustness along with supporting operators during machine tuning.

Four input five output nonlinear model is formulated for Microtron accelerator through system identification method. Using this model Microtron agent is developed with system dynamics learning plans. The study carried out for the multi-agent based cooperative control for the problem of beam disturbance at source (at Microtron output) shows that provided there exists a method for online model identification and an adaptive controller for Microtron, the agents can successfully maintain the injection current at booster above threshold value while reducing the number of changes required in transport line settings. This scenario further improves with the cooperative optimisation by both agents using the system dynamics learning and future moves prediction plans. Where both the agents Microtron and TL-1 cooperatively work towards common goal of maintained the injection current at booster while fulfilling their individual goals without operator interventions.

For controlling the electron beam orbit in synchrotron radiation sources the conceptual multi-agent based operator support and beam orbit control scheme is formulated. The orbit control job is distributed among multiple low complexity reactive agents at lower layer to control the local orbit for individual beam lines and insertion devices in an optimized manner. The high complexity monitoring agents at higher layer evaluates the performance of the beam line control agents and insertion device control agents and facilitates the generation of medium complexity trainer and fault assistance agents at the middle layer. In this multi-agent based environment study shows that with gradient based constrained trainer agents the lower layer agents can be trained online in the dynamic accelerator environment. Further with this constrained training the beam availability to beamlines can be improved as during the agent training, the local orbit bump leakage is constrained to within beam usable limits thus rest of the beam lines (beam lines other than the one whose control agent is being trained) can be used for routine experiments. Moreover with this scheme the available beam quality to users can be improved through prioritized optimization for the cases when multiple beam lines demand the beam correction simultaneously. Assuming that under dynamic environment the agents can successfully maintain their up to date skill levels with the presented skill learning method, the further study can be done on using this distributed skill information for extracting the systems model which can be used to improve the system performance.

c. Stability in multi-agent systems in context to safety critical systems.

For ensuring the safe operation of the machine in multi-agent based control of the potentially hazardous and safety critical systems like accelerators, the safety measures are needed to be taken at each level of system design. On the positive side the agent based control provides opportunity for maintaining system in stable states through automatic alarm handling. The agents with goals to handle undesirable system events (events that indicates the likelihood of system approaching towards potentially unstable system states) can be made so that the correction can be done much before the system actually lands to some unstable state. Realisation of these goals requires that the agents must posses all the safety related parameter values and action list, at all instants of time. One of the possible method for this is the pre-built action recipe and the limit bounds set as the default by the designer at the time of system design. For avoiding the potentially hazardous system states resulting as the outcome of the agent's action in response to some changing environment condition, the online formal method of logic verification through model-checking algorithms can be adopted. This approach mainly requires a clear formulation of structural (static) and behavioural (dynamic) system specifications. The loss of communication and loss of functionality (resulting out of failure of some of the coaliating agents in the multi-agent environment) is another important parameter that needs attention. This can be avoided to some extent using the method of online system integrity checking through token passing within the coaliating agents. Using the state qualification methods and maintaining action and state history, the agents can easily recover the system from unstable state through action reversal for recoverable systems. Although the cases discussed in this thesis are all of recoverable type, for non-recoverable systems the agent plans can be built that allows the agent to control the system for limited states. Also for easy testing and validation of the inter agent actions and their sequence the inter agent interactions are advised to be through centralised data concentrating agents. Also it is advised to distribute the agents in the multi-agent environment at different layers considering their data grouping requirement as per the critical safety capability served by the individual agents (in other words the agents are to be built in such a way that the safety related capability served by the individual agent is dependent only on the system parameters that are in direct perception of that agent). This can avoid the loss of safety related services of the agent, in the event of malfunction/failure of the neighbouring agents. Also the thorough testing of the complete system in simulation mode along with the formal method of verification is a must to be done before allowing such systems to actually control the safety critical parts of the machine such as the automatic operations of safety shutters and vacuum gate valves in case of accelerators.

6.2 Future work

The thesis clearly brings out the benefit of intelligent agent based control implementations for the accelerator control system scenarios particularly for the synchrotron radiation source facilities like Indus-1 and Indus-2 in the form of enhanced machine operation and machine availability. Since at present there is no active control working at the Indus accelerator complex based on the developed system models and the distributed intelligent object based accelerator control system framework. The developed intelligent object can be deployed for automatic controlling of two subsystems, pre-injector Microtron and Transport line-1 of Indus-1. This requires modifications to the existing control system as well as some of the existing beam diagnostics in both of the subsystems (Like development and testing of the low level hardware interface for the new Microtron control system, installation of online beam position indicators in the TL-1 and development of their interfaces). In future we plan to put the accelerator control system up-gradation proposal based on intelligent object concept in-front of Indus accelerator committee for seeking relevant approvals and machine time allotment. Although the implementation and commissioning depends on the final availability of required hardware and software components, with the present level of developed system interfaces, agent framework and system models we expect that if this work is taken on priority basis (including the related material procurements) than this work can be completed within one and half year time.

Also there are three agent based active control implementation related works (not

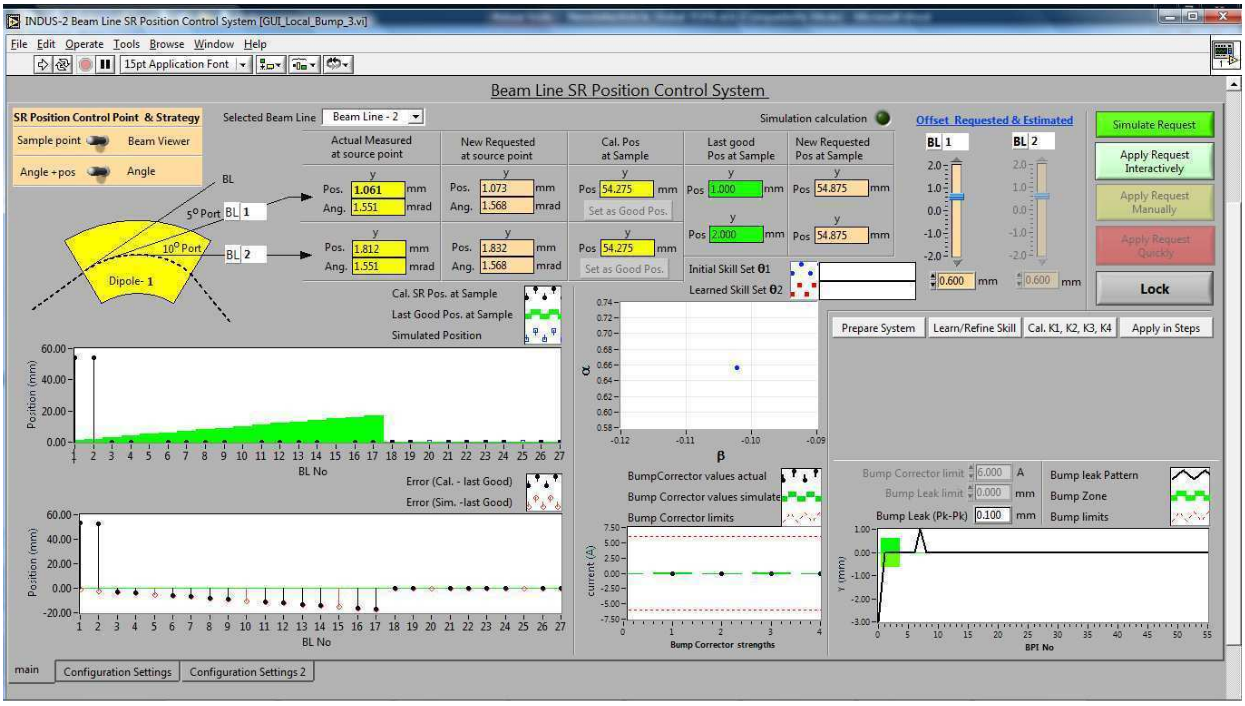


Figure 6.1: GUI of Beam line SR position control system.

discussed in this thesis) taken up at Indus complex that are inspired by this thesis and use the agent framework and system models developed as part of this thesis work. First one is the agent based controller for fast magnet power supply of fast orbit feedback control system of Indus-2. This system uses the developed subsumption reactive agent framework. In its first phase the agent based maximum rate digital controller is synthesized over National Instruments PXI platform using NI PXI-7841R, R Series Multifunction RIO with Virtex-5 LX30 FPGA for hardware in loop simulation and tested ok with simulated power supply transfer function over PXI-7841R card. Then in its second phase the hardware prototype for the system is being developed with the help of power supply section. Second is the model predictive controller for slow orbit feedback control system of Indus-2. This system uses the developed Indus-2 model in its implementation. Third is the agent based beam line synchrotron radiation position control system for Indus-2. This system is basically the cut-set version of the distributed intelligent agent based beam orbit control of synchrotron radiation sources presented in section 5.4 of this thesis. The main difference lies in the control system layer where the different agents are distributed and also that in this implementation only the minimal set of agents are considered for simplicity in its first version. This uses the developed agent framework with *beam line control agent* and *insertion device control agents*. In its first version, activation of the agent plans are being performed only by operator and the agents autonomy is restricted to only suggesting the applicable plan to operator. Figure 6.1 shows the GUI for this system. We expect these to be commissioned in the coming year.

There are many promising directions for future research and applications like it was shown that the multi-agent based beam orbit control scheme can successfully perform the orbit control job and at the same time can impart training to individual local orbit controlling agents to learn the skills of maintaining the closed orbit bump leakage within bounds. Future work can be carried out on the schemes for combining this distributed skill information towards extracting the overall systems response matrix. This can greatly improve the global orbit feedback control system performance for the dynamic environment cases.

The study shown that with the agent based control using model based tracking concepts the transport line performance in accelerators can be enhanced for the case of sensor and actuator failures. Further study can be done on similar grounds towards system robustness enhancement for the global orbit feedback control systems in synchrotron radiation sources by online identify the faulty BPM's and substituting their measurement values with the model based tracked data values for the purpose of calculating the correction settings. Moreover the continuous model based tracked data at BPI locations can be used with the statistical techniques to qualify the data from BPI showing the problem of time dependent offset error. This can improve the overall electron beam orbit stability in the synchrotron radiation sources.

Another promising area of the agent based control in synchrotron radiation sources can be the fast orbit feedback control system. In the fast orbit feedback control the orbit is sensed using BPI's at the rate of few KHz (5 to 10KHz) and correction is applied to fast correctors. One of the important limiting factor in this type of systems in the overall system delay which restrict the overall control system bandwidth. Future study can be carried on the agent based methods using the systems dynamics and the noise learning capabilities to improve the overall systems noise suppression ability thus this can result in the over all electron beam stability at rate up to hundreds of hertz. Further with the inclusion of online system identification methods for network control systems based plans for the agents this enhancement can be carried further by adaptive controller design.

Appendix A

Least square fitting methods

A.1 Linear least square fitting

For estimation of linear model from the experimentally collected data set, linear least square method is used. For the given data set of the form { (x_1, y_1),(x_2, y_3),...,(x_n, y_n) } where x is the system input(independent variable) and y is the system output (dependent variable) and the linear fitting function given by

$$y = ax + b \tag{A.1}$$

the square error between the fitted model and the measured data will be given by

$$E(a, b) = \sum_{i=1}^{n} (y_i - (ax_i + b))^2 \tag{A.2}$$

Now to find a and b that minimises E, requires one to find the value of (a, b) such that

$$\frac{\partial E}{\partial a} = \frac{\partial E}{\partial b} = 0 \tag{A.3}$$

thus differentiating $E(a, b)$ gives

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{n} 2(y_i - (ax_i + b)).(-x_i) \tag{A.4}$$

$$\frac{\partial E}{\partial b} = \sum_{i=1}^{n} 2(y_i - (ax_i + b)).1 \tag{A.5}$$

now setting $\frac{\partial E}{\partial a} = 0$ and $\frac{\partial E}{\partial a} = 0$ and dividing by 2 gives

$$\sum_{i=1}^{n}(y_i - (ax_i + b)).x_i = 0 \tag{A.6}$$

$$\sum_{i=1}^{n}(y_i - (ax_i + b)) = 0 \tag{A.7}$$

rearranging them gives

$$\left(\sum_{i=1}^{n} x_i^2\right) a + \left(\sum_{i=1}^{n} x_i\right) b = \sum_{i=1}^{n} x_i^2 x_i y_i \tag{A.8}$$

$$\left(\sum_{i=1}^{n} x_i\right) a + \left(\sum_{i=1}^{n} 1\right) b = \sum_{i=1}^{n} x_i^2 y_i \tag{A.9}$$

writing it in matrix form gives

$$\begin{pmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i \end{pmatrix} \tag{A.10}$$

and the solution for (a, b) are calculated as

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i \end{pmatrix} \tag{A.11}$$

A.2 Polynomial least square fitting

similar to the previous section case the least square method for the polynomial function given by

$$y = a_m x^m + a_{m-1} x^{m-1} + ... + a_0 \tag{A.12}$$

is used to estimate the values of $a_m, a_{m-1}, ..., a_0$ by solving the equation

$$\begin{pmatrix} a_m \\ a_{m-1} \\ \cdots \\ \cdots \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} x_i^{2m} & \sum_{i=1}^{n} x_i^{2m-1} & \cdots & \sum_{i=1}^{n} x_i^{m+1} & \sum_{i=1}^{n} x_i^{m} \\ \sum_{i=1}^{n} x_i^{2m-1} & \sum_{i=1}^{n} x_i^{2m-2} & \cdots & \sum_{i=1}^{n} x_i^{m} & \sum_{i=1}^{n} x_i^{m-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \sum_{i=1}^{n} x_i^{m+1} & \sum_{i=1}^{n} x_i^{m} & \cdots & \sum_{i=1}^{n} x_i^{2} & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^{m} & \sum_{i=1}^{n} x_i^{m-1} & \cdots & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{n} x_i^{m} y_i \\ \sum_{i=1}^{n} x_i^{m-1} y_i \\ \cdots \\ \cdots \\ \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i \end{pmatrix} \tag{A.13}$$

A.3 Non-linear least square fitting

For estimating the regression parameters for non-linear fitting the trust-region method is used. For the purpose of least square minimisation, the υ is formulated as a function of γ, in particular

$$\upsilon(\gamma) = \sum_{i=1}^{n} (y_i - f(x_i, \gamma))^2 \tag{A.14}$$

where the $f(x_i, \gamma)$ is the fitting function.

Trust-Region-Reflective Algorithm

The Trust-Region-Reflective (TRR) algorithm is among one of the optimisation algorithms generally used for fitting. The principle of TRR is as follows: for minimisation of any function $\upsilon(\gamma)$ (where γ can be a vector), to find the point γ_{i+1} with a smaller function value than the current point γ_i, the non-linear function υ is approximated by a quadratic (from its Taylor expansion around γ_i) in the vicinity N of γ_i. Where N is the trust region, and the improved point γ_{i+1} should fall in this region. The step $s_i = \gamma_{i+1} - \gamma_i$ is calculated by approximately solving the equation

$$\min_{s \in N} \psi_i(s), \tag{A.15}$$

where

$$\psi_i(s) = \mathbf{g}^T s + \frac{1}{2} s^T \mathbf{H} s \tag{A.16}$$

and $\mathbf{g}$ and $\mathbf{H}$ are the gradient and Hessian, respectively, of υ evaluated at γ_i. Generally $s \in N$ is equivalent to $\|D_i s\| < \Delta_i$ where D_i is a scaling matrix and Δ_i is the positive trust region size and $\|.\|$ is 2-norm.

Generally in order to solve for s_i more quickly and easily, one restrict the vicinity N to a 2 dimensional subspace $\mathbf{V}$ with vector υ_1 in the direction of gradient $\nabla\upsilon(\gamma_i)$(negative of the steepest descent direction) and υ_2, an approximate Gauss-Newton direction. This Gauss-Newton direction is the solution υ_2 to

$$\mathbf{H}\upsilon_2 = -\mathbf{g}. \tag{A.17}$$

The steps of TRR algorithm is given in algorithm 7

Algorithm 7: Trust-Region-Reflective algorithm.

Data: regression variables γ, trust region size Δ

Result: calculated value for regression variables γ such that $\upsilon(\gamma)$ is minimum

initialise $\mathbf{g} \leftarrow \nabla\upsilon(\gamma)^T$; $\mathbf{H} \leftarrow \nabla^2\upsilon(\gamma)$;

```
/*if gradient is grater than some minimum limiting value ϵ i.e.
  if minima has not reached then repeat following steps */
while ||g|| > ϵ do
    /*solve for trial step s that minimises the below function */
    s ← arg min_{||D_i s||<Δ_i} ψ(s) = g^T s + ½ s^T H s;
    Pred ← ψ(γ_i + s_i) − ψ(γ_i) ;        /*calculate predicted reduction*/
    Ared ← υ(γ_i + s_i) − υ(γ_i) ;        /*calculate actual reduction*/
    if (Ared/Pred) < η_1 then
        γ_{i+1} ← γ_i ;                   /* reject step */
        Δ ← Δτ_1 ;                        /* reduce Δ by a factor τ_1 < 1 */
    else
        γ_{i+1} ← γ_i + s ;               /*accept step and update H*/
        if (Ared/Pred) > η_2 then
            Δ ← max { Δ, τ_2||D_i s||};   /*enlarge Δ by a factor τ_2 > 1*/
        end
    end
end
```

This algorithm uses ratio $r = (Ared/Pred)$, the ratio of the actual reduction and the predicted reduction for decides the acceptance/rejection of calculated step s and the trust region adjustments. It is based on the following key points

- if the ratio $r = (Ared/Pred)$is between $0 < \eta_1 < r < \eta_2 < 1$, one have that the approximated model is quite appropriate; one accept the step and do not modify the trust region.
- if the ratio $r = (Ared/Pred)$is smaller $r \leq \eta_1$, one have that the approximated model is not appropriate; one do not accept the step and one must reduce the trust region by a factor $\tau_1 < 1$.
- if the ratio $r = (Ared/Pred)$is large $r \geq \eta_2$ one have that the approximated model is very appropriate; one accept the step and also enlarge the trust region by a factor $\tau_2 > 1$.
- The typical values for different parameters are $\eta_1 = 0.25, \eta_2 = 0.75, \tau_1 = 0.5, \tau_2 = 3$.

References

[1] Adacher L., Agnetis A., and Meloni C., *Autonomous agents architectures and algorithms in flexible manufacturing systems*, IIE Transactions, vol. 32, no. 10, pp. 941-951, (2000).

[2] Agrawal R. K., Chauhan A.,Singh G., Saifee K., Seema M., Fatnani P., Husain R., Sheth Y., *Implementation and experience of energy ramping for INDUS-2*, In proceedings of Asian Particle Accelerator Conference, Jan 29- Feb 2, Indore, India,(2007).

[3] Alvaro Carrera & Carlos A. Iglesias, *A Modular Agent Architecture for Optimising Hypothesis conrmation cost in Network Diagnosis*, In 9th European Workshop on Multi-agent Systems,(2011).

[4] ANL web page `http://www.aps.anl.gov/Accelerator_Systems_Division/Accelerator_Operations_Physics/software.shtml`(2011).

[5] Amundson J., Spentzouris P., Qiang J. and Ryne R., *Synergia: An Accelerator Modeling Tool with 3-D Space Charge*, available from `http://cd-amr.fnal.gov/aas/synergia_paper.pdf`,latest accessed on Jan (2013).

[6] Barnes M.J., Borburgh J., Goddard B., Hourican M., *Injection and extraction magnets: septa*, CERN, Geneva, Switzerland, online available from `http://arxiv.org/ftp/arxiv/papers/1103/1103.1062.pdf`, latest accessed on Jan (2013).

[7] Barnes M.J., *Beam Transfer Devices: Septa*, CERN TE/ABT , online available from `http://cas.web.cern.ch/cas/Belgium-2009/Lectures/PDFs/Barnes-1.pdf` latest accessed on Jan (2013).

[8] Baxter J. and Bartlett P. L., *Infinite-Horizon Policy-Gradient Estimation*, Journal of Artificial Intelligence Research, Vol.15, pp.319-350,(2001).

[9] Benson, S., and Nilsson, N., *Reacting, Planning and Learning in an Autonomous Agent*, in Machine Intelligence 14, K. Furukawa, D. Michie, and S. Muggleton, (eds.), Oxford: the Clarendon Press, (1995).

[10] Bonasso R. P. , Kortenkamp D., Miller D. P., and Slack M., *Experiences with an Architecture for Intelligent Reactive Agents*, in Intelligent Agents II eds Wooldridge M., Muller J. P., and Tambe M., pp 187-202, Lecture Notes in Artificial Intelligence 1037, Springer Verlag, (1996).

[11] Brooks R.A., 1986, *A Robust Layered Control System for a Mobile Robot.* In IEEE Journal of Robotics and Automation, Vol. 2(1),pp.14-23, (1986).

[12] Bratman M. E., *Intention, Plans, and Practical Reason*, Cambridge, MA, Harvard University Press, (1987).

[13] Bratman M.E., Israel D.J., and Pollack M.E., *Plans and resource-bounded practical reasoning*, Computational Intelligence, Vol 4, pp.349-355,(1988).

[14] Bradshaw J. M., *An introduction to software agents*, in Software Agents, Bradshaw J. M. Ed., AAAI Press/MIT Press, 1997, pp. 3-46, (1997).

[15] Bryant P.J., *A brief history and review of accelerators*, CAS-CERN Accelerator school:4th general accelerator physics course, Julich, Germany, September 17-28, pp.1-16.(1990).

[16] Chiddarwar S. Shital. , Babu N. Ramesh. *Multi-agent System for Off-line Coordinated Motion Planning of Multiple Industrial Robots* , International Journal of Advanced Robotic Systems, Vol. 8, No. 1, pp 102–112, (2011).

[17] Chakraborty D. and Sen S., *MB-AIM-FSI : A Model Based Framework for exploiting gradient ascent MultiAgent Learners in Strategic Interactions*, Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Padgham, Parkes, Mller and Parsons (eds.), Estoril, May 12-16, Portugal, pp.371-378, (2008).

[18] Chung Y., Kahana E., Kirchman J., Lumpkin A., Meyer J., Plouviez E., Scheidt K., Taurel E., Ando A., Sasaki S. and Taketani A., *Local Beam Position Feedback Experiments on the ESRF Storage Ring*, Particle Accelerator Conference, May 1-5; Dallas, Texas, pp.2699-2701,(1996).

[19] Choi J., Oh S., Horowitz R., *Cooperatively learning mobile agents for gradient climbing*, In the Proceedings of 46th IEEE Conference on Decision and control, December 12-14, New Orleans, Louisiana, USA, pp.3139-3144,(2007).

[20] Clocksin, William F., Mellish, Christopher S., *Programming in Prolog*, Berlin , New York, Springer-Verlag, (2003).

[21] Cockcroft J. D. and Walton E. T. S. , *Experiments with High Velocity Positive Ions*, Proc. R. Soc. Lond. Vol. A 129, pp.477-489, (1930).

[22] Crowley-Milling M.C., *The Design of the Control System For the SPS*, CERN Library , CERN 75-20, 22 December, (1975).

[23] Dang V.D., Dash R.K., Rogers A., Jennings N.R., *Overlapping coalition formation for efficient data fusion in multi-sensor networks*, in: Proceedings of The Twenty First National Conference on Artificial Intelligence (AAAI-06), pp.635640, (2006).

[24] Donald H. B., Pascal E. and Edgar W., *Review of third and next generation synchrotron light sources*, Journal of Physics B: Atomic Molecular and Optical Physics, Vol.38, pp.S773S797,(2005).

[25] Edwards D.A., Syphers M. J., *An Introduction to the Physics of High Energy Accelerators*,Book,Publisher: Wiley-VCH, 1^{st} Edition, 1993, pp.112-113,(1993).

[26] Englemore R. and Morgan T., *Blackboard Systems*, Addison-Wesley,(1988).

[27] ERMAN, L., HAYES-ROTH, F., LESSER, V. and REDDY, D., *The HEARSAY II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty*, ACM Computing Surveys. Vol. 12, (1980).

[28] Ferguson I. A., *Towards an architecture for adaptive, rational, mobile agents*, In Werner E.and Demazeau Y., editors, Decentralized AI 3 - Proceedings of the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91), pp. 249-262. Elsevier Science Publishers B. V.: Amsterdam, The Netherlands, (1992).

[29] Ferber J., *Multi-Agent Systems. Addison-Wesley*,(1999).

[30] Genesereth M. R. and Nilsson. N., *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann Publishers: San Mateo, CA, (1987).

[31] Georgeff, M.P. and Lansky A.L.,*Reactive reasoning and planning*, in Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87), Seattle, WA, p.677-682.(1987).

[32] Ghosh H. and Chaudhury S. , *Distributed and Reactive Query Planning in R-MAGIC: An Agent based Multimedia Retrieval System*, IEEE Transactions on KDD, Volume 16, Issue 9, Sept. 2004 Page(s):1082–1095, (2004).

[33] Glenn Decker, *Beam Stability In Synchrotron Light Sources*, Proceedings of DIPAC 2005, Lyon, France, (2005).

[34] Hayes C. C. , *Agents in a nutshell a very brief introduction*, IEEE Transactions on Knowledge and Data Engineering, vol. 11, no. 1, pp. 127-132, (1999).

[35] Head J.D., Gomes J.R., Williams C.S., Lee K.Y., *Implementation of a multi-agent system for optimized multiobjective power plant control*, North American Power Symposium (NAPS), September 26-28; Arlington, USA, pp.1-7, (2010).

[36] Herr W., *Algorithms and procedures used in the orbit correction package COCU*, Technical report CERN SL/95-07(AP),(1995).

[37] Hudson, D.L., Cohen, M.E.,*Use of Intelligent Agents in the Diagnosis of Cardiac Disorders*, Computers in Cardiology, Vol 29, pp.633-636, (2002).

[38] Iselin F.C., Jowett J.M., Pancin J., Adelmann A., *MAD Version 9* , EPAC2000, Vienna, Austria, June 26-30,(2000).

[39] James F. R. , *An Investigation into Reactive Planning in Complex Domains*, AAA1-87, pp202-206, (1987).

[40] Jennings N. R. and Wittig T., *ARCHON: Theory and Practice, in Distributed Artificial Intelligence: Theory and Praxis* edited by Avouris N. M. and Gasser L., Kluwer Academic Press, pp.179-195,(1992).

[41] Jennings, N. R., Varga L. Z., Aarnts R., Fuchs J. and Skarek P. , *Transforming Standalone Expert Systems into a Community of Cooperating Agents*, Int. Journal of Engineering Applications of Artificial Intelligence 6(4) pp.317-331, (1993).

[42] Jennings, N. R., *Commitments and Conventions: The Foundations of Coordination in Multi-Agent Systems*, In: The Knowledge Engineering Review, 2(3): pp.223-250,(1993).

[43] Jennings, N. R., and Pople, J. A., *Design and Implementation of ARCHONs Coordination Module*, Proc. Workshop on Cooperating Knowledge Based Systems, Keele, UK, pp.61-82,(1993).

[44] Jennings N. R. , Corera J. M. , Laresgoiti I. , Mamdani E. H., Perriollat F., Skarek P. , Varga L. Z., *Using, ARCHON to develop real-world DAI applications for electricity transportation management and particle accelerator control*, IEEE Expert, Vol.11, pp.64-70,(1996).

[45] Jennings N. R. and Wooldridge M., *Agent Technology: Foundations, Applications and Markets*. Springer. (1998).

[46] Jingyi Li, Gongfa Liu, Weimin Li, Baogen Sun, Caozheng Diao and Zuping Liu, *Development of orbit correction system of HLS storage ring*, Proceedings of the second Asian Particle Accelerator Conference, September 17-21; Beijing, China, pp.681-683,(2001).

[47] Kapitza S.P. and Melikhin V.N., *The Microtron* [in Russian] (Izd.Naka,Moscow, 1969). *The Microtron* [in English], trans. by Sviatoslavsky I.N., Harwood, London,(1978).

[48] Karen S. White, *Status and Future Developments in Large Accelerator Con-trol Systems*, 9th International Computational Accelerator Physics Conference (ICAP 2006),Oct 2-6, Camonix Mont-Blanc, France,(2006).

[49] Karl L. Brown, Carey D. C. , Iselin Ch., and Rothacker F., *TRANSPORT—A Computer Program for Designing Charged Particle Beam Transport Systems*, CERN 73-16, revised as CERN 80-4, CERN, (1980).

[50] Katz E. P.,*A Simplifying Diagrammatic Representation Of Crisp And Fuzzy Teleo-Reactive Semantic Circuitry For Application In Robotic Agent Task Control*, 1998 IEEE International Conference on Systems, Man, and Cybernetics (SMC '98), San Diego, California, (1998).

[51] Kao-Shing Hwang, Chia-Yue Lo, Wen-Long Liu, *A Modular Agent Architecture for an Autonomous Robot*, Instrumentation and Measurement, IEEE Transactions on , vol.58(8), pp.2797-2806, August(2009).

[52] Kay, (1984). Computer Software. Scientific American 251(3): pp.53-59,(1984).

[53] Khosla R. and Dillon T., *Engineering Intelligent Hybrid Multi-Agent Systems*, Kluwer Academic Publishers.(1997).

[54] Kissler K.H., Rausch R., *New Control Architecture for the SPS Accelerator at CERN* , CERN Library , CM-P00061066. report. latest accessed on (2013).

[55] Klusch M., Shehory O., *A polynomial kernel-oriented coalition formation algorithm for rational information agents*, in: Proceedings of International Conference on Multi-Agent Systems (ICMAS-96),pp. 157164, (1996).

[56] Klein W. , Stern C., Luger G. and Olsson E. , *Designing a portable architecture for intelligent particle accelerator control*, Proceedings of Particle Accelerator Conference. 12-16 May 1997, Vancouver, BC, Canada , Volume 2, pp.2422-2424, (1997).

[57] Klein W.B., Stern C.R., Luger G.F., and Olsson E.T., *An Intelligent Control Architecture for Accelerator Beamline Tuning*, in Proceedings of AAAI / IAAI, pp.1019-1025, (1997).

[58] Klein W.B., Westervelt R.T., and Luger G.F., *A general purpose intelligent control system for particle accelerators.* In Journal of Intelligent and Fuzzy Systems, New York: John Wiley, (1999).

[59] Klein W.B., Stern C.R., Luger G.F., and Pless D., *Teleo-reactive Control for Accelerator Beamline Tuning*, in Proceedings of Artificial Intelligence and Soft Computing, July 24-26, Banff, Alberta, Canada. pp.372-379,(2000).

[60] Klein W.B., Westervelt R.T., and Luger G.F., *An Architecture for intelligent control of particle accelerators available* from `http://www-bd.fnal.gov/icalepcs/abstracts/PDF/wpo63.pdf`,(2011).

[61] Kragic D., Christensen H. I.,*Confluence of parameters in model-based tracking*, In Proceedings of IEEE International Conference on Robotics and Automation, Taipei, Taiwan, vol. 3, pp.3485–3490,(2003).

[62] Konolige K., *A Deduction Model of Belief*, Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA, (1986).

[63] Komma V.R., Jain P.K. and Mehta N.K, *Agent-based simulation of a shop floor controller using hybrid communication protocols*, International Journal of Simulation modelling Vol.6 , No. 4, pp.206–217, (2007).

[64] Kyrki V., Kragic D., *Integration of Model-based and Model-free Cues for Visual Object Tracking in 3D*, In IEEE International Conference on Robotics and Automation, Barcelona, Spain, pp.1554–1560,(2005).

[65] LabVIEW Introduction, `http://www.ni.com/labview/`(2011).

[66] Lars Braubach, Alexander Pokahr and Winfried Lamersdorf, *Jadex: A BDI Agent System Combining Middleware and Reasoning*,in book"Software Agent-Based Applications, Platforms and Development Kits",2005,Edited by Unland R., Calisti M., Klusch M., pp.143-168, publisher Birkhuser-Verlag.(2005).

[67] Lesser, V. R., and Corkill, D. D, *Distributed Problem Solving*, in Encyclopedia of AI (ed S. C. Shapiro), pp.245-251, John Wiley and Sons,(1987).

[68] Lee S.Y., *Accelerator Physics*, World Scientific, Singapore, (1999).

[69] Lesperance Y., etc. *Foundations of a logical approach to agent programming*, In Wooldridge M., Muller J. P., and Tambe M., editors, Intelligent Agents II (LNAI volume 1037), pp. 331-346, Springer-Verlag: Berlin, Germany, (1996).

[70] MAD web page `http://mad.web.cern.ch/mad/`(2011).

[71] Mark S. Fox, Mihai B. , and Rune T., *Agent-Oriented Supply-Chain Management*, The International Journal of Flexible Manufacturing Systems, vol. 12, pp.165188 (2000).

[72] Musliner J., Durfee E. H., Shin K. G., *CIRCA: A cooperative intelligent realtime control architecture*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23(6), pp.561-1574, (1993).

[73] Muller J. P., Pischel M., *Modelling Interacting Agents in Dynamic Environments*, in Proc. of Eleventh European Conf. On Artificial Intelligence, ECAI-94, pp.709-713, (1994).

[74] Muller J. P., Pischel M. , Thiel M., *Modelling Reactive Behavior in Vertically Layered Agent Architectures*, in Wooldridge M. and Jennings N. R., editors, Intelligent Agents: Theories, Architectures and Languages, LNAI, vol. 890, pp. 261-276, Springer Verlag, (1995).

[75] Murugesan, S., *Intelligent software agents on the Internet and Web*, TENCON '97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications., Proceedings of IEEE , vol.1, pp.220, 4-4 Dec. (1997).

[76] Morgan Gwyn , *The application of multi-agent systems to the design of an intelligent geometry compressor*, PhD thesis , Brunel University , London, (2002).

[77] Morgan, G., Rzevski, G., Wiese, P.,*Multi-Agent Control of Variable Geometry Axial Turbo Compressors*, IMechE Journal I, vol.126, (2004).

[78] Morge, Maxime and McGinnis, Jarred and Bromuri, Stefano and Toni, Francesca and Mancarella, Paolo and Stathis, Kostas, *Toward a modular architecture of argumentative agents to compose services*, Fifth European Workshop on Multi-Agent Systems, (2007).

[79] Maturana F., Staron R., Hall K., Tichy P., Vrba P., Vladimir M., *Agent virtual machine for automation controllers*, Robotics and Computer-Integrated Manufacturing, vol. 24(5), pp.616-624, (2008).

[80] Metzger, M.; Polakow, G.; , *A Survey on Applications of Agent Technology in Industrial Process Control*, Industrial Informatics, IEEE Transactions on , vol.7, no.4, pp.570-581, (2011).

[81] Nilsson, N. J.,*Teleo-Reactive Programs for Agent Control*, Journal of Artificial Intelligence Research, Vol.1, pp.139-158, January (1994).

[82] Nilsson, N. J., *Teleo-Reactive Programs and the Triple-Tower Architecture*, Electronic Transactions on Artificial Intelligence, Vol.5, Section B, pp.99-110.(2001).

[83] Norman J., Preece A.D., Chalmers S., Jennings N.R., Luck M., Dang V.D., Nguyen T.D., Deora V., Shao J., Gray W.A., Fiddian N.J., *Agent-based formation of virtual organizations*, International Journal of Knowledge Based Systems Vol. 17(24), pp.103111,(2004).

[84] Nwana H. S., Lee L. and Jennings N. R., *Co-ordination in multi-agent systems*, Software Agents and Soft Computing, Nwana H. S. and Azarmi N., Eds. New York : Springer, pp.42-58 (1997).

[85] Parunak H. V. D. , *Agents in overalls: experiences and issues in the development and deployment of industrial agent-based systems*, An expanded version of an invited talk at PAAM '99, Available at http://www.erim.org/ vparunak /PAAM99Keynote.pdf, (1999).

[86] Patruu B., Tomozei C., *Agent Technology in Monitoring Systems*, Int. J. of Computers, Communications & Control,Vol. V(2010), No. 5, pp.852-861, (2010).

[87] PBO Lab web page `http://www.ghga.com/accelsoft/pbolab.html`,latest accessed on Jan (2013).

[88] Part-1- Accelerator Systems, Chapter6, *NSLS-II Preliminary Design Report*, Brookhaven national Laboratory, (2007).

[89] Pattie Maes, Robert H. Gutmann, Alexandros G. Moukas, *Agents that Buy and Sell: Transforming Commerce as we Know It*, Communications of the ACM, Vol. 42, No.3, pp.81-91, ACM Press, March (1999).

[90] Pugliese R., Bille F., Abrami A. and Svensson O., *Applying Intelligent System Concepts to Automatic Beamline Alignment*, International Conference on Accelerator and Large Experimental Physics Control Systems 1997, 3-7 Nov 1997, Beijing, China, (1997).

[91] Rao A. and Georgeff M. *BDI Agents: from theory to practice*. In V. Lesser, editor, Proceedings of the First International Conference on Multi-Agent Systems (ICMAS95), pp. 312319. The MIT Press: Cambridge, MA, USA, (1995).

[92] Rausch R., *The Evolution Of Bus System And Their Impact On The Process Control Environment* Journal of the European Physical Society, Vol. 22, 1991 . CERN document number CERN SL/90-120 (CO)(1991).

[93] Ray D. N., Mukhopadhyay S., and Majumder S. *A brief comparison between the Subsumption Architecture and Motor Schema Theory in light of Autonomous Exploration by Behavior Based Robots*, In 14^{th} National Conference on Machines and Mechanisms (NaCoMM09), Durgapur, India, December 17-18, (2009).

[94] Richard W. Helms and Georg H. Hoffstaetter, *Orbit and optics improvement by evaluating the nonlinear beam position monitor response in the Cornell Electron Storage Ring*,Phys. Rev. ST Accel. Beams vol. 8, pp.062802/1-062802/6, (2005).

[95] Rosenschein S. J. and Kaelbling L. P., *A situated view of represent-action and control*, In Agre P. E. and Rosenschein S. J., eds, Computational Theories of Interaction and Agency, pp 515-540. The MIT Press: Cambridge, MA, (1996).

[96] RRCAT web page `http://www.cat.gov.in/technology/accel/indus/actrep03/2_1.html`(2011).

[97] RZEVSKI, G. 1994. *Designing Intelligent Machines Volume 1, Perception, Cognition and Execution*. Butterworth-Heinemann,(1994).

[98] Sangulagi P. , Sutagundar A. V., Manvi S. S., Vidya S Bennur , *BDI Agents for Information Fusion in Wireless Sensor Networks* , International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 1, No. 7, pp. 144–150,(2012).

[99] Sidnal N. and Manvi S. S., *BDI Agents for Bidding in English Auctions in Mobile E-commerce*, In International Conference on Information and Computer Networks (ICICN 2012), IPCSIT vol. 27, IACSIT Press, Singapore, pp. 258–263(2012).

[100] Sandholm T.W., Lesser V.R., *Coalitions among computationally bounded agents*, Artificial Intelligence Vol. 94(1) pp.99137,(1997).

[101] Sandholm T.W., Larson K., Andersson M., Shehory O., Tohm F., *Coalition structure generation with worst case guarantees*, Artificial Intelligence Vol.111(12), pp.209238 (1999).

[102] Schirmer D. Hartmann P., Buning T. and Muller D., *Electron Transport Line Optimization using Neuronal Networks and Genetic Algorithms*, Proceedings of 9th European Particle Accelerator Conference EPAC2006, Edinburgh, Scotland, June 26-30, (2006).

[103] Schirmer D., Hartmann P., Buning T.,Muller D., *Electron Transport Line Optimization using Neuronal Networks and Genetic Algorithms*, In Proceedings of 9th European Particle Accelerator Conference EPAC06, Edinburgh, Scotland, pp.1948–1950,(2006).

[104] Sessler and Edmund Wilson Andrew, *Engines of Discovery : A century of particle Accelerators* , Word scientific , (2007).

[105] Shen W., Norrie D. H. and Barthes J.P. 2001. *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*, Taylor and Francis,(2001).

[106] Sheng G., Jiang X. , and Zeng Y., *Optimal coordination for multi-agent based secondary voltage control in power system*, IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific, August 14-18; Dalian, China,(2005).

[107] Sidnal, N. S., Manvi, S. S.,*Intelligent Agent Based Model for Auction Service Discovery in Mobile E-Commerce*, International Journal of E-Business Research (IJEBR),Vol. 8, No. 1, pp.76–97, (2012).

[108] Sinha G., Kumar A., Mishra A.K., Singh G., *Effect of magnetic field coupling on Indus-2 quadrupole magnets*, Asian Particle Accelerator Conference - 2007 (APAC-07), Jan.29-Feb.2, Indore, India, pp. 378-380,(2007).

[109] SIMION web page `http://simion.com/info/manual.html`,latest accessed on Jan (2013).

[110] SLAC web page `http://www.slac.stanford.edu/grp/ssrl/spear/at/`(2011).

[111] Stassinopoulos G., and Lembesis E. , *Application of a Multi-Agent Cooperative Architecture to Process Control in the Cement Factory, ARCHON* ,Technical Report 43, Atlas Elektronik, Bremen, Germany,(1993).

[112] Storn R., Price K., *Differential Evolution, A Simple and Efficient Heuristic Strategy for Global Optimization over Continuous Spaces*, Journal of Global Optimization, Vol. 11, pp.341-359,(1997).

[113] Stern C. , Olsson E., Kroupa M., Westervelt R., Luger G. and Klein W., *A Control System for Accelerator Tuning Combining Adaptive Plan Execution with Online Learning*, International Conference on Accelerator and Large Experimental Physics Control Systems 1997, Nov 3-7, Beijing, China, (1997).

[114] TRACE web page `http://laacg.lanl.gov/laacg/services/traceman.pdf`, latest accessed on Jan (2013).

[115] Toru OHATA, *Controls developments at SPring-8*, presentation available from `http://www.aps.anl.gov/News/Conferences/2003/3way/.../3way_ohata.pdf`, latest accessed on (2013).

[116] Tuijnman, F., and Afsarmanesh, A., *Distributed Objects in a Federation of Autonomous Cooperating Agents*, Proc. Int. Conf. on Intelligent and Cooperative Information Systems, Rotterdam, The Netherlands, pp.256-265,(1993).

[117] Weiss G., *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*, The MIT Press,(1999).

[118] Wiedemann H., *Particle Accelerator Physics*, Springer, 3rd edition, (2007).

[119] Wittig T., *ARCHON an architecture for multi-agent systems*, Ellis Horwood Ltd, (1992).

[120] Wittig T., Jennings N. R., and Mamdani E. H., *ARCHON - A Framework for Intelligent Cooperation*, IEE-BCS Journal of Intelligent Systems Engineering Vol.3(3) pp.168-179,(1994).

[121] Worrall A. D., Marslin R. F., Sullivan G. D., Baker K. D., *Model-based Tracking*, In Proceedings of The British Machine Vision Conference, Glasgow, UK, pp.310-318,(1991).

[122] Wrulich A., *RACETRACK: a computer code for the simulation of nonlinear particle motion in accelerators*, DESY Report 84/026, Deutsches Elektronen-Synchrotron, Hamburg, Germany (March 1984).

[123] Yamada T. and Yamaguchi S., *Reinforcement Learning Using a Stochastic Gradient Method with Memory-Based Learning*, Electrical Engineering in Japan, Vol. 173(1),(2010).

[124] Yan J., Guan X. P., Tan F. X., *Target tracking and obstacle avoidance for multi-agent systems*, International Journal of Automation and Computing, vol.7(4), pp.550-556,(2010).

[125] Yan J., Guan X. P., Luo X. Y., Tan F. X., *Target tracking and obstacle avoidance for multi-agent Networks with Input Constraints*, International Journal of Automation and Computing, vol.8(1), pp.46-53,(2011).

[126] Yufeng Mo. and Yisong Xu., *A Navigation System Based on Subsumption Architecture*, In Proceedings of the 2010 International Conference on Measuring Technology and Mechatronics Automation - Vol. 01 (ICMTMA '10), Vol. 1. IEEE Computer Society, Washington, DC, USA, pp.161-164. (2010).

[127] Yuri M. Tsipenyuk, edited by Sergey P. Kapitza ,*The Microtron: Development and Applications*, Taylor & Francis, London,(2002).