Universitat de Girona

MODELLING, CONTROL AND SUPERVISION FOR A CLASS OF HYBRID SYSTEMS

Santiago ESTEVA PAYET

ISBN: 84-688-6957-0
Dipòsit legal: GI-445-2004

Universitat de Girona

Departament d'Electrònica Informàtica i Automàtica

THESIS

MODELLING , CONTROL AND SUPERVISION FOR A CLASS OF HYBRID SYSTEMS

Santi Esteva i Payet

Thesis directors:

Josep Lluís de la Rosa i Esteva
Naly Rakoto-Ravalontsalama

Reports:

Josep Aguilar-Martin
Jean Claude Hennet
Vicenç Puig
Gilles Roux
Joan Colomer
Emilio García
Joaquim Meléndez

Date : 14-11-2002

Contents

CHAPTER 1 HYBRID SYSTEM DESCRIPTION

CHAPTER 2 ALPHA MODEL

CHAPTER 3 CONTROL

CHAPTER 4 SUPERVISION

CHAPTER 5 APPLICATION

CHAPTER 6 CONCLUSION

APPENDIX A

THESIS STRUCTURE

The aim of this thesis is to narrow the gap between two different control techniques: the continuous control and the discrete event control techniques DES. This gap can be reduced by the study of Hybrid systems, and by interpreting as Hybrid systems the majority of large scale systems. In particular, when looking deeply into a process, it is often possible to identify interaction between discrete and continuous signals.

This thesis has the following structure:

Chapter 1 provides a definition of Hybrid systems and describes the models currently used in industry. Then it provides a description of research groups, models and tools used to analyse different applications for the mathematical theories.

Chapter 2 explains the proposed model, and its use for control and supervision of Hybrid systems. A comparison between different modelling methods is demonstrated, and the class of particular systems to be modelled by the proposed models is explained.

Chapter 3 presents the application of this model to control. Optimal control and safe control are analysed and, an application of mathematical study is presented.

In chapter 4, this model is applied to supervision. An analysis sequence to find the faults is developed and an example is presented to clarify this methodology.

Chapter 5 presents an application to control of mobile robots. Dynamic scenario with time constraints is treated as example for the theory developed in this thesis.

Chapter 6 presents the conclusion of this work.

Chapter 1

1.1 HYBRID SYSTEMS DESCRIPTION

1.1.1 INTRODUCTION

Hybrid systems is in general the mixture of two kinds of signals or techniques. In this work Hybrid systems are defined as systems which deal with both continuous and discrete signals. In the Automatic Control domain, these signals are those which represent the evolution of different parts of process.

This area has emerged in the last years due to the increase in the computational capacities of computers to model and analysis the possible evolution of systems . Great benefit resulted from computer scientists who have developed several tools to create safe protocols for the analysis and design of computer networks. Automatic Control scientists can benefit from their tools, to analysis and verify control procedures.

Several areas have emerged application Hybrid systems theories in Automatic control area: From low level control to supervision, passing over different topics as stability, controllability, hierarchy, optimality, reachability. This provides powerful techniques to design safe systems and controllers, and give as a result news applications which are now under development, such as automatic highways, automatic airplanes etc.

1.1.2 DEFINITION

Hybrid systems are systems that have both continuous, and discrete signals. Continuous signals are generally supposed continuous and differentiable in time, since discrete signals are neither continuous nor differentiable in time due to their abrupt changes in time. Continuous signals often represent the measure of natural physical magnitudes such as temperature, pressure etc. The discrete signals are normally artificial signals, operated by human artefacts as current, voltage, light etc.

The first works came from the study of timed systems [Alur 90] [Henzinger 91] which analyses the interaction between discrete events with time evolution, this provides real time specifications. Following the study of Hybrid automata with continuous evolution as proportional evolution of time [Alur Dill 94][Henzinger 96] defining the theory of Timed and Hybrid automata.

This thesis is focused in a particular types of systems with discrete and continuous signals in interaction. That can be modelled hard non-linealities, such as hysteresis, jumps in the state, limit cycles, etc. and their possible future behaviour are well expressed and interpretable in the model description.

1.1.3 INTERACTION

Classical continuous control can analysis systems without interactions between discrete and continuous signals. This work focuses on systems where the interaction between these two types of signals are expressed by:

- Changes in the discrete signals, which can cause dynamic changes to any of continuous signals (figure 1.1a).
- Several changes on the continuous signals, which can cause changes to any of discrete variables (figure 1.1b).

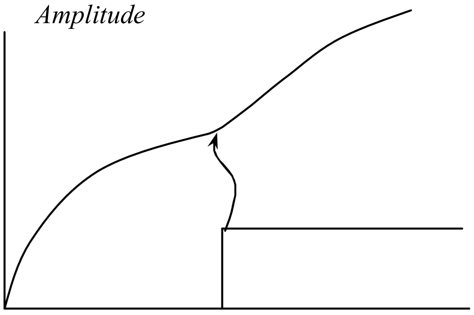


Figure 1.1a
Continuous behaviour as a response
to a discrete change

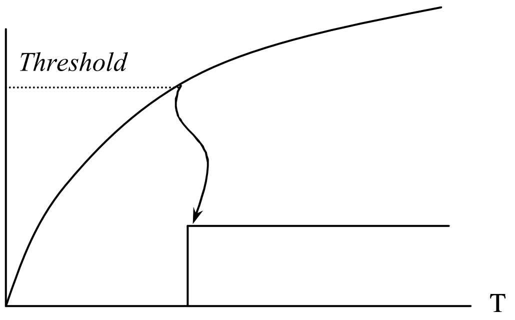


Figure 1.1b
Discrete behaviour as a response
to a continuous change of magnitude.

Typical cases are the interactions between natural and artificial signals, temperature changes for several heat artefacts, or continuous signals measured by discrete sensors as level measure for float relay. In general the majority of these processes has actuators of a discrete type and continuously measure of physical variables.

1.1.4 MOTIVATION

It is possible to find Hybrid systems in the nature as a result of actions of artificial devices, in the industry, in the laboratory, and most common systems as transport, cooking, and the technological developments as aircraft, generating power stations etc.
Typical processes modelled as Hybrid systems are production systems, chemical process, or continuos production when time and continuous measures interacts with the transport, and stock inventory system.

This large group of systems -hybrid in this sense- is the principal motivation for the study of mathematics and the control possibilities. It is needed to have a broad range of knowledge in control domain to join it together with the classical continuous control and the discrete automata, for applications in the complex processes that become commonly present in modern manufacturing systems.

Complex systems as manufacturing lines are in hybrid a global sense. They can be decomposed into several subsystems, and their links are natural continuous, discrete or hybrid subsystems. Hierarchy and interactions of subsystems is an important area under investigation.

Another motivation for the study of Hybrid systems is the tools developed by other research domains. These tools benefit from the use of temporal logic for the analysis of several properties of Hybrid systems model, and use it to design systems and controllers, which satisfies physical or imposed restrictions.

1.1.5 PARTICULARITIES

To classify the Hybrid systems it is necessary to specify the characteristics of systems. The behaviour of continuous system can be represented by a differential equation in its states. For the discrete variables, the states are defined for the events or changes. A new state is defined for each different group of differential equations and different values of discrete signals. This means that the states are unique, there are not two states not differentiable for their signals.
This specification is a particularity of Hybrid systems that can be represented by this approach.

This approach is not interesting to use with:

Systems with only one state.
Systems with only one group of differential equation.

Simple systems with one state can be analysed by the classical continuous control, with initial and end conditions, temperature control systems, control the level, etc. this are individual processes without interactions.
Systems that can be represented by several states, with only one group of continuous equations, can be analysed by continuous control, aggregating continuous equations and treated by discrete inputs.
Sequential systems are a simplification of class of Hybrid systems treated in this thesis. The application of all techniques presented does not allow Batch processes which involve one sequential phases and control actions.

This current approach can be adequately when is applied over systems with several states, where a network can be created which expresses the relations between the state transitions, and the multiple ways to drive the system from the Starting point to the final one are present (figure 1.2).

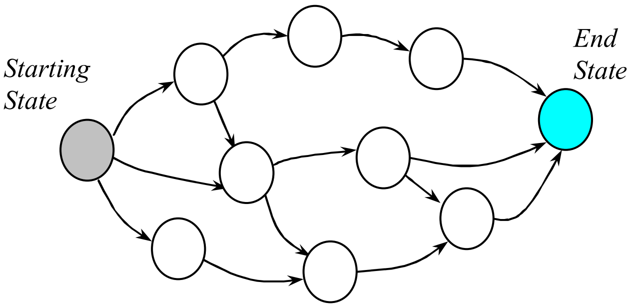


Figure 1.2 Possible system evolution represented by a 2D graph.

1.2 SURVEY OF RELATED MODELS FROM THE LITERATURE

1.2.1 CLASSIFICATION

The large class of signals and the analysis of their characteristics, can be used for the classification of Hybrid systems. From economics to the engineering, it is possible to find different signals to be analysed. Continuous and discrete signals discovered from the process need to be separated and analysed their specific characteristics.

Further classification can be applied for the characteristics of continuous and discrete signals to be analysed. Signal aspects such as determinism, linearity, time invariance and structure relations are used to classify the research groups and the physical systems [Deshpande 94].

- Characteristics of Discrete signals:
    - *Observable or Unobservable events.*

      Those events that can be recorded by sensors.
    - *Controllable or Uncontrollable events.*

      Events that can be activated or deactivated by the supervisor or controller.
    - *Deterministic or non-deterministic.*

      Depending if multiple transitions can be activated at same time and the controller can choose the order to be treated.

- Characteristics of Continuous signals:
    - *Linear models or non-linear.*

      Depending on the superposition of signal dynamics versus the inputs.
    - *Differentiable or non differentiable.*

      Depending of continuity of variables.
    - *Deterministic or non-deterministic.*

      Depending on the possible perturbation of system, not measurable.
    - *Time varying invarying or Stochastic.*

      The variability in time of the model.

- Control action :

  Discrete

  In the supervisor all the signals are discrete.

  Continuous

  In the supervisor or controller treats continuous signals in domain and range.

  Discretized

  In the supervisor interprets the continuous signals in intervals. The domain could be continuous.

  Combined

  The control actions are combined by continuous and discrete components.

The combination of these characterised signals spans the diversity of applications. Different mathematical algorithms are used to analysis and resolve the control issues. This classification can be used to classify the large literature on Hybrid systems.

1.2.2 RESEARCH GROUPS

Different authors in the 1990's investigated and developed Hybrid systems models. The origin of their work comes from the commuted systems in the sliding mode [Utkin 78] and the works developed since the analysis of computer programs interacting over physical systems.
Using the augmented description of finite state automaton, the different groups defines their particularities with own descriptors.

The finite state automaton is defined by (Q, Σ, Q_o, Ω) where Q is a finite set of states, Σ is the set of events Q_o is the initial state and the Ω is the transition function: $\Omega : Q \times \Sigma \to Q$

Following is a description of models for several of main research groups

- Alur-Courcoubertis-Henzinger-Ho 93 AT&T Bell Labs USA
  Define a hybrid automata as a discrete system under continuous environment, time, space, and variables with linear progression.

The model is the tuple: $HA=(Vd,Q,u1,u2,u3)$ where Vd are continuous variable, Q discrete states ,$u1$ activities, $u2$ invariant, $u3$ transition relations.

- Deshpande-Varaiya 95 U. Berkeley USA

  Define a non-deterministic automata with edges , transition function adding the reset transition and information structure to fix the control strategy. Uses the rectangular automata with the linear progression confined in intervals.

  The model: (Q,R,Σ,E,Φ) where Q are discrete states ,R continuous components ,Σ set of events, E edges, Φ control vectors.

- Tittus, Edgard 94 Chalmers Technological U., SWEDEN

  Adjoin guards to control the transitions.

  The model: $(Q,\Sigma,\delta,X,W,f,g,\gamma,\alpha)$ where Q are discrete states ,Σ set of events, δ transition function, X continuous variables, W input vector, f state vector, g transition function, γ events generator ,α actuator.

- Branicky 94 Massachusetts Institute of Technology LIDS USA

  It is a continuous system interacting with a computer intervening a A/D and D/A converter mapping the continuous system in states.

  The model : (Q,Σ,σ,C,F) where Q are *states,* $\Sigma=(x \quad \tau\,\phi)$ *is a tuple of state vector, transition function and continuous dynamics,* σ *transition map,* C *set of continuous jumps,* F *destination map.*

- Antsaklis Lemmon 93 U. Notre Dame USA

  Continuous systems interfaced with discrete controller give a discrete expression represented as a graph. Over the state space is represented as a overlapped sliding surfaces.

  The model: (S,X,R,δ,ϕ) where S are set of states ,X plant symbols, R control symbols, δ transition function, ϕ output dynamic function.

- Sifakis, Maler 93 Verimag Grenoble F

  It is a model running in a two steps, the temporal evolution which evolves the continuous behaviour, and the discrete evolution resetting the time.

  The model: $(V,X,\text{->},tcp,\varphi)$ where V are set of positions, X state variables ,-> set of edges, tcp time can progress, φ *state* transition function.

A classification of dynamics covered by different authors:

Model	Cont. Dynamics.	Discrete D.	Control action
Alur-Henzinger	LN DT	DT	DC
Antsaklis-Lemmon	LN DT	DT	CTD
Branicky	NL DT	DT	CT DC
Deshpande-Varaiya	NL NDT	NDT	CT DC
Tittus-Egard	LN DT	DT	DC
Sifakis-Maler	NL DT	NDT	DC

Where:
LN Continuous dynamics expressed in linear models.
NL Continuous dynamics expressed in non linear models
DT Deterministic behaviour. NDT Non deterministic behaviour.
DC Control action over discrete, variables CT over continuous.
CTD Control action over discretized continuous variables.

All the models can model MIMO and SISO systems. The most interesting aspect is to use a model which covers non-linear time varying systems. The price is a complex mathematical expression, and control algorithms. It is preferable to use the adequate model which covers the specific system.

1.2.3 APPLICATION

The bulk of models is used to solve different applications of Hybrid systems, from system design to control design. These models provide Mathematical expression used to design accurate models under wide set of restrictions.

Analysis

Determine the behavioural characteristics (input-output, state trajectories) of a given dynamic system.

The concrete analyses over the system are:

- Limit cycles , deadlock free.
  Analysis of limit cycles to avoid the blocking or cycling states.
- Verification.
  Determine whether a given system/program satisfies given properties/specifications. [Alur Henzinger 92][Peleties Decarlo 93][Deshpande Varaiya 94][Clarke 95]

Synthesis

Define a complete and unambiguous syntax and semantics for describing the desired and current behaviour system/program.

- Design
  It is possible to create a structure for the system which able the control and supervision. [Larsson 97, Mosterman 97]

Control

Given a model of dynamic system and specification of desired controlled (closed loop) behaviour, synthesise a controller to achieve the specification.

The models can be used to analyse:

- Controlability
  It is the capacity to drive the system from one state to another. [Nerode Kohn 93][Stiver Antsaklis 94][Tittus Edgard 94]
- Reachability
  It is the capacity to reach the end state from the Starting point. [Asarin 95]
- Stability
  Lyapunov criteria applied over the commuted systems to analysis the global stability and robustness. [Branicky 94, Antsaklis 95, Passino 94, Pettersson Lennartsson 96]
- Discrete decision
  Analysis of discrete actions to be done to jump at specific state. [Ramadge 90]
- Continuous control

Expansion of continuous control to the switched systems. [Guckenheimer 95, Tittus Egardt 98]

- Optimality

  Analysis of optimal control to minimise a specific cost function.[Petterson Lennartson 95] [Kohn 95] [Branicky Mitter 95]
- Hierarchy

  Analysis of hierarchical control.[Caines 95][Nerode Kohn 95 Pappas Sastry 98]
- Piecewise linear control.

  Use the approximation in linear piecewise regions. [Delchamps 90] [Pettit Wellstead 96].
- Discrete event systems

  Convert the Hybrid systems in discrete systems. [Antsaklis Lemmon 93][Ramadge Wonham 94]
- Supervisory control

  Study of supervised control as high layer of control [Cury 98, Lemmon Antsaklis 98]
- Agentified control

  Design of Hybrid controllers for a large systems. [Nerode Kohn 93, Lygeros 96, Tomlin Sastry 98]

1.2.4 TEMPORAL LOGIC

Sequence of events in time need to be represented by means of a temporal logic that allows for representing and analysing well-formed formulas WFF.

Clark defines the Computational Temporal Logic CTL [Clark 81] by means the Existential (E) Global (G) Always (A) Future (F) Next (X) and Until (U) operators, which represent by formulas the time behaviour of sequence of events.

Examples:

EF(p)	p can be activated at several future step
EX(p)	p can be activated in the next step
AF(p)	p will be activated at several future step
AG(p)	p will be activated at all future step
AU(p,q)	q will be activated at several future step, until p will be.

In other side Pnueli defined a Prepositional Linear Temporal Logic PLTL [Pnueli 97] which express the time or continuous variables in linear relation versus time in quantitative form marking time between events.

Examples:

AG (T<100)	Temperature is always less than 100º
¬ EF (L>50)	Level should never be above 50 cm.

Dual relations can transforms one expression to another:

Always *p* activated is equal to not exist not *p* active.

Exist *p* active is equal to not always no *p* active.

This logic is used in several tools to automatise the analysis process of behaviour for Hybrid systems.

1.2.5 MODEL CHECKING

Model checking is a mechanism to analysis the behaviour of the model. This method determines if the proposed formula is contained in the possible behaviour of system. A well formed formula WFF in temporal logic is searched in the binary graph diagram BDD of complete system behaviour.

Representation of the graph for all possible behaviour of system and finding in the graph the desired controlled behaviour is a manual procedure to check the system. Otherwise the automatic procedure uses the abstraction and reduction techniques over the complete system to reduce the behaviour expression and enable the computation analysis.

Automatic analysis of Hybrid systems uses the model of the timed automata [Pnueli 97], and the variation of rectangular automata [Henzinger 91], which represents all possible situation of the evolved system in time from one starting state. Varaiya [Deshpande 95] augments this model to the reset state and different initial progression. Lafferriere Pappas Sastry study the decidability problem for several classes of LTI systems. The extension for all possible LTI systems are the actual research of different groups [Kurzhanski Varaiya 99, Lafferriere et al 98]

1.2.6 VERIFICATION

Verification is whether a system satisfy certain properties. Is the solution of reachability problem. It is a property which denotes the capability to achieve the goal, the End state or the End point, from the Starting point in a finite time. Forbidden states can be avoided verifying the capability of system to reach these states as a safety property.

It is possible to solve the reachability problem through solving the dynamic equations of system, this is the algorithmic solution. For the reachability problem it is necessary to have a complete knowledge of system to understand and express the question to be solved. An automated processing can be obtained by computer through a specific program that verifies the question raised for the system. The analysis is done reasoning using temporal logic and exploring over all possible behaviours of system expressed in a BDD. On a Sun computer workstation, the verification of three tank system use a megabyte of memory and takes 2 seconds of time using HyTech.

The objective in the reachability analysis, is to find the language desired to reach our objective from the language generated by the Hybrid systems from the Starting point. This is analysed by this expression:

$$L_H \cap \overline{L}_D = 0$$

which L_H is the behaviour of Hybrid systems, and L_D is the desired behaviour of system.

Forward verification determines the states that can be reached from the Starting point. This property is useful to analysis to avoid several forbidden states and assure safe conditions.

Backward verification is useful to determine if there is a path to arrive at specific End state from the Starting point.

Forward and Backward verification give as a result the simple possibility to find the solution

1.2.7 TOOLS

The different research groups to facilitate modelling and model checking in Hybrid systems have developed these different tools:

- SHIFT

  Is a tool developed in the PATH group, at Berkeley, to model and analysis Hybrid systems. At this time it is under development the verification capacity of tool. Have their own language of programming. It is a powerful simulator, can work with thousands states and hundreds continuous non-linear variables. [Deshpande Gollu Varaiya 99]

- HYTECH

  Tool created at Berkeley in the group of Henzinger to verify the Hybrid systems. The rate of evolution of continuous variables must be constant, this force to approximate LTI systems in a group of states. Can use the intervals to confine the progression of continuous signals. A graphical interface is adopted to program the models. [Henzinger Ho 95]

- COSPAN

  Is a commercial tool developed in the AT&T Bell Laboratories oriented to the analysis and verification of Hybrid systems. It is presented in a graphical representation of graph and all details inside the states. [Alur Kurshan 96]

- UPPAAL

  A tool developed in the UPSALA Sweden University at Alborg, to model and checking reachability and liveness properties, and is based on the theory of timed automata. Time is discretized to be analysed as a different states [Bengtsson Larsen Petterson Yi 95]

- KRONOS

  Is a tool developed in the IMAG Grenoble Lab. to verify real time systems. This tool can analyze Timed systems as protocols or asynchronous circuits. [Sifakis Yovine 96]

1.3 THE CLASS OF ANALYSED HYBRID SYSTEMS

The Hybrid systems treated in this work are systems with several discrete states, always less than thirty states (it can arrive to NP hard problem), and continuous dynamics evolving with expression: $\dot{X} = K_1 - K_2 * X$ with $K_i \in \mathrm{R}^n$ constant vectors or matrices for X components vector. In several states the continuous evolution can be several of them $K_i = 0$.

In this formulation, the mathematics can express Time invariant linear system. By the use of this expression for a local part, the combination of several local linear models is possible to represent non-linear systems. And with the interaction with discrete events of the system the model can compose non-linear Hybrid systems.

The class of systems treated is a typical system in the physical domain. In this domain, differential equations represent the evolution in time of the continuous signals, subject to the applied control action over, and the interactions between the states, and the possible ways to conduce the system from the origin to the reach goal are the focus interest in this processes . There are several areas of Hybrid systems that are applicable for the proposed techniques for modelling, control and supervision, from chemical reactions, with phase interaction as thermal ,pressure and humidity actuation, and transport systems, packaging and inventory systems.

Time Invariance and systems under perturbation are treated in this thesis. A interaction model-system is used to drive the system to the goal. Deviations are analysed in order to obtain the action to be applied over the system to arrange the reaction of the system at active perturbation.

Especially multistage processes with high continuous dynamics are well represented by the proposed methodology. Sate vectors with more than two components, as third order models or higher are well approximated by the proposed approximation. Flexible belt transmission, chemical reactions with initial start-up and mobile robots with important friction are several physical systems, which profits from the benefits of proposed methodology (accuracy).

1.3.1 PROBLEM STATEMENT

The motivation of this thesis is to obtain a solution that can control and drive the Hybrid systems from the origin or Starting point to the goal. How to obtain this solution, and which is the best solution in terms of one cost function subject to the physical restrictions and control actions is analysed. Hybrid systems that have several possible states, different ways to drive the system to the goal and different continuous control signals are problems that motivate this research.

How to control these systems by means of continuous and discrete variables makes it profitable in the automatic control area. Systems in all the areas, from chemical, industrial and electronics, to aeronautics, transport and traffic, are domains to be modelled by Hybrid systems.

Specific problems can be determined by the use of this kind of hybrid models are:

- The unity of order.

  Switching states with different orders create a problem in order to solve the initial conditions for the new Starting point in the new state. Aggregation can solve this problem by reducing the order of states, but this also means to reduce information on the model and its approximation.

- Control the system along a reachable path.

  Apply a sequence of control signals in the right direction to assure the controllability and reachability of desired solution or End point, is a crucial goal for Hybrid systems control.

- Control the system in a safe path.

  Forbidden states must be avoided on the way to drive the system through the End point. Critical states may create dangerous situations that need to be isolated in order to guarantee a safe control of system.

- Optimise the cost function.

Obtain the value of control signals in order to obtain the minimal cost for the control objective, subject to the previous premises, or safe and reachable solution is the global objective for the controlled system.

- Modularity of control

  Complex systems must be analysed in a modularity sense to solve the great problem to control a large scale system. Independent subsystems or relations between subsystems need to be known to treat this class of system models.

These problems are treated in the following chapters of thesis.

1.3.2 CONTRIBUTION

The contribution of this thesis into domain of automatic control is the presented model for a class of Hybrid systems, a system that can be modelled by piecewise linear system. It can handle a highly dynamical system with strong interactions between discrete and continuous signals.

This model is used to control and supervise the system: a previous analysis gives the control tables to guide the system from the Starting point to the End point, following the minimum cost analysis for desired trajectory. A supervisor is used to detect abnormal situations and analyses the cause to realign the system to the desired trajectory.

Contribution of this work is a complete framework to work with the majority Hybrid systems, the procedures to model, control and supervise are defined and explained and its use is demonstrated. Also explained is the procedure to model the systems to be analysed for automatic verification.

Great improvements were obtained by using this methodology in comparison to using other piecewise linear approximations. It is demonstrated in particular cases this methodology can provide best approximation.

Chapter 2

2 .1 ALPHA MODEL

2.1.1 INTRODUCTION

The necessity of the one new approach to model non-linear Hybrid systems, came from the class of the systems we are working with, a class of the Hybrid systems in the domain of Automatic Control scientists. A system with a few states and several continuous variables modelled by second order dynamics are the typical systems treated for our group, robots in dynamic scenarios, filling tanks and mixing them, are systems with three, four or fifth states and some continuous variables. The existing in the literature models is more appropriate to model computer networks or chemical reaction, systems those gives thousand states and a few continuous variables.

The requirements of the system on which we work is, a model that can represent the behaviour of the non-linear system, and to possibilities the prediction the possible future behaviour of the model in order to apply a supervisor which decides the optimal and secure action to drive the system toward the goal.

2.1.2 STATE SPACE PARTITION

State space partition is a methodology to model non-linear system intervening local linear models. It is the principle of divide and conquer is a huge methodology used in automatic control. Different authors apply this methodology in a different sense, modelling, control, etc. The great interest of this is the interpretability in a 2D plane for all the sub-models, which shows the knowledge of the complete system.

Some authors use this methodology to model and control non-linear systems, differences are the order, limits, and transitions.
A brief description of the authors:
[Sontag 81] Obtain the optimal linear local controller for non-linear systems.
[Asarin Maler Pnueli 95] Defines Piecewise Constant Derivative PCD as convex polyedrals associates a constant rate equation.

[Antsaklis 93] Partition the state space according the plant discrete symbols.
[Flaus Halla 97] Make regular partition in the state space, rectangular regions defines a valid model.
[Henzinger Wong-Toi 95] Solves the problem to model non-linear systems with linear phase portrait approximations.
[Pettit Wellstead 95] Redesign some parts of the vehicle controller in a PL system.
[Johansen Foss 95] Blends local models by fuzzy fusion to construct non-linear state space using operate regime decomposition.
[Johansson 99] Creates a toolbox for the computational analysis of piecewise systems.

The advantages to use state space partition are the compactness of the representation of the global model, the information of the current model and their evolution, tendencies, stability or next possible models. The problem is to model the system with several local models, the use of the unification for the order able to represent into the graph the global model. Also the problem is the identification procedure to cover all the possible operating regions for the system.

Operating regime decomposition uses the knowledge of the system, functional states, qualitative and quantitative information to identify the structure and the model. In a process where the operation regimes are well defined including overlapping regions this technique is very useful to identify the models. Johansen and Foss uses this methodology with local model validity function to obtain the local model in a global function.

Difference of the authors are the partition of the models and limits, in our case these limits are not fixed in the state space representation, these are variable according to the evolution of the system, different evolutions imply different limits in the state space diagram. The proposed Piecewise Linear approximation gives accurate modelling for the high dynamic systems.

The use of the second order models is for the advantage of the memory model which can represent little delays, tendencies, oscillations, etc. and the simplicity of the second order which uses two internal states. The vast majority of the systems can be modelled by one second order model, and the rest can be modelled by some second order local models.

Composition of local models using a measure which indicates the validity of the local model that can be represented by the structure in figure 2.1 .

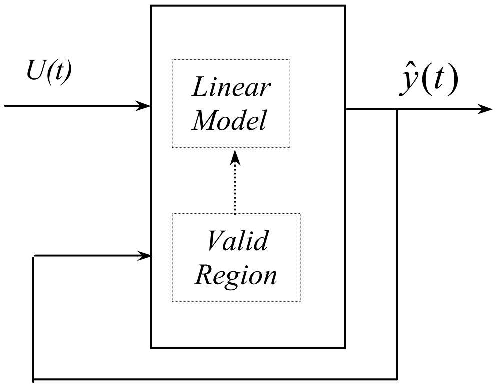


Figure 2.1 Model structure for local model composition

Figure 2.1 use the output prediction to measure the validity region for the local models. This is represented by a rectangular grid in the state space representation SS, the use of different signals to measure the valid region for the local model, makes complex partitions in SS.

The model can interact with the process measuring the noise and deviations, in order to readjust the model and detect possible faults into the process. A complete validated model is necessary when is required to detect faults for the control and the supervisor system. Erroneous model indicates false faults due the inexact model at certain situation.

2.2 LINEAR HYBRID MODEL

2.2.1 MODEL

The proposed model solves the specified problems in the switching models problem, the initial condition calculus and the unity of the order models.

Continuous and discrete phenomena are represented in Linear hybrid models, defined with eighth-tuple parameters to model different types of hybrid phenomena [Branicky 98]:

- Q_h Joint of set of α-plans, also is the discrete representation DS of the state space.
- $\sum S$ Joint set of state space regions for each α-plan.
- $\mathrm{X} \in R^2$ Bidimensional state vector corresponding to ordinary differential equation ODE for each SS region.
- Virtual discrete events $\{e_i\}$, generated to crossing the boundaries on SS
- $\Omega : S \times e \to S$ Transition function into α-plan.
- Join set of discrete events $\{V_i\}$, either generated internally or by the supervisor among α-plans.
- $\Psi : DS \times V \to DS$ Transition function in different α-plans.
- Continuous control signal $U(t)$ for SS

Where $Q_h = \sum \{DS_i\}$ is a part of the complete system model representation . $Q_h \subset Q_s$ is the part that the Hybrid system models. Q_h , is a joint set of the discrete states *DS* created using the composition of the output measure of the system and the alpha parameter, each one of this *DS* is a joint set of *SS* state space regions modelling the local model in state space form

$\mathrm{X} \in R^2$ are the estimated states for the system in the local region, two-dimensional vectors represents the part of the system, that can be modelled by the second order LTI model.

Limits between these local models mark transitions e fired when these limits are crossed. Virtual events e_i applied over a transition function Ω defines the next local model to be applied.

Discrete events of the system V_i transits to the next discrete state *DS.* Transition function Ψ determines the next discrete state and local model as the response to this discrete event.

Summaring , the α-plans are a continuous state space representation formed by a set of piecewise regions, so that $DS_i = \sum S_j$, where S_j is a limited region by bounds $[b_i, b_j]$, with a Linear Time Invariant system (LTI) with dynamics $\dot{x}(t) = Ax(t) + Bu(t)$, where $\dot{x} = \{x_1, x_2\}$ second order local model.

2.2.2 TRANSITION FUNCTIONS

Two types of transition functions are used, depending on who produces the event. The transition function is responsible to indicate the new model used to model the system in following period, and the parameters to initiate the modellisation.

$\Omega : S \times e \rightarrow S$ Determines the next model to be used as a response to the crossing limit of the local validity model This function calculates the initial conditions for the next model, by scaling the final values of the previous one, in order to avoid discontinuities in the output signal.

$\Psi : DS \times V \rightarrow DS$ Determines the discrete model *DS* to use or α-plan, assigning in the new α-plan the Starting point, as a response to the discrete transition triggered. The subtraction of the parameter needed to start the next model will be given by Ω.

In some systems the continuous variables not measurable can be estimated from the system via observers. Virtual events e_i are triggered by the computer, and discrete events V not directly observable from the system can be deduced [Lemmon Antsaklis 93] to fire the transition function.

The following table resumes the relation to be applied over this class of the Hybrid system. These relations indicate the next model and initial conditions as a response to the events. This is a state event table, similar that used in the discrete systems.

CURRENT STATE	OUTPUT CONDITIONS	EVENT PRESENTED	NEXT STATE	INITIAL CONDITIONS
V1ON-M1	$X1_f$, $X2_f$	δ	V1OFF-M1	$X1_i$, $X2_i$
V1OFF-M1	$X1_f$, $X2_f$	σ	V1OFF-M2	$X1_i$, $X2_i$
. . .				

Table 2.1 Transition table

The inputs of the transition table are:

The current state. The event presented. And the output conditions for the continuous variables at the instant of the fired event.
The outputs of the transition table are: The next state with the information of the continuous dynamics model and the actions to be applied. Initial conditions of this next state for continuous variables to be calculated once the output conditions are obtained. Ω and Ψ transition functions are intrinsically expressed in this table.

This table can represent a system with memory, as a non-monotonous system. Through the current state it is possible to differentiate states in front of the same event using different names for this state. This is result of the non complete information in the state, only measured variables of changing values are represented in this state model. The memory information possibilities to represent the state within minimum quantity of information.

Details of how to fill the table is explained afterwards when initial conditions and Ω transition function are treated.

2.2.3 INITIAL CONDITIONS

In the switching models is necessary to express the initial conditions for the next model. The continuity of the continuous variables, their tendency, and the conservative energy are the objectives. To solve this problem recursive backward algorithm is applied to obtain an acceptable initial condition.

In the state transitions the initial conditions need to be determined by the following method: Some restrictions are imposed, continuity of the output, and continuity of the dynamics of the continuous variables, these are the conservative energy principle.

The point to start to calculate these initial conditions, is the end value of the continuous variables when the system cross the limits for the validity model.

Two different cases in the transitions are present, transitions into DS, between two local models, and transitions between DS , among local models in different α-plans. The first one is the model approximation to the continuous system, the second one is the varying system at the discrete event. In both cases is used the same algorithm.

Different models structure with different coefficients make it impossible to use the end values of the state vector as the initial conditions of the next model. In special cases, in canonical forms a relation between end values of the state space and initial conditions for the next model can be determined with this form :

$$X_2' = X_2 * C_1 / C_1' \quad \text{and} \quad X_1' = X_1 * K \qquad (2.1)$$

Where X_2 is the scaled output, and the X_1 express the dynamics. X' are the components for the next model , C_i is the coefficient of the LTI models and K is the empirical relation between X_1 components.

When the models have different structure, this means p.e. the previous is ARX the next is BJ etc. Is not possible to find a relation between the internal states from one model to the next. One possibility is to simulate the new model according the current evolution of the process.

The proposed solution to overcome this problem is the following procedure:
Back to the system on the past, in the previous model before the transition, and acquire one value of the past state vector. And use this point to start the simulation with new model, once the state vector reach the limits of the previous model bounds, the new starting state vector is obtained. Only rest to validate this solution, is used another Starting point from the past to analysis if the Starting point for the new model is the same (Figure 2.1)..

As example to exemplify this solution think in a tank with two parts of different diameter. The bigger one is up and the thinner is bottom, the tank is filling from empty. The process is to simulate the bigger is as tall as the combined one, and to simulate the evolution from the same Starting point. Once the simulation arrives at the level of the change in the other model, the values of the state vector are obtained, the level and the velocity of the level measure are well defined for the new model.

Lemma 1. Initial condition of state vector assures available error model.

Proof : Generated error depends on the control signal starting at the model. This error is reduced due to the form of the control signal, in sense to drive the system to this new adequate model for the continuous signals.

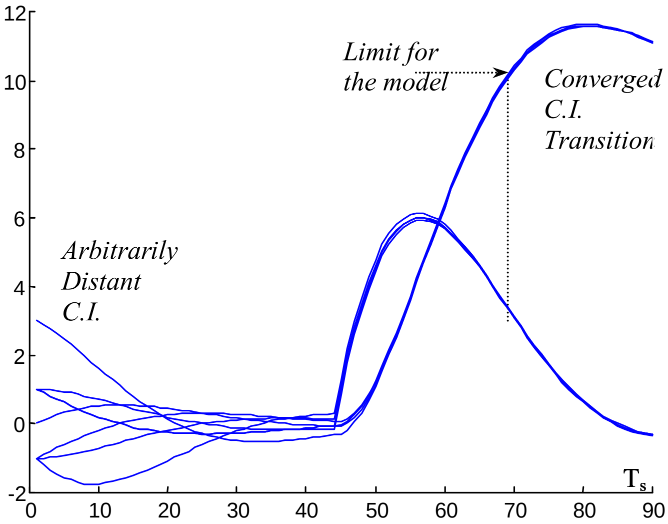


Figure 2.1 : Calculus of the transition initials conditions.

In figure 2.1 the convergence for the arbitrary initial conditions are indicated. In the abrupt change of the signal of control step in the 45 T samples even though it presents differences on the internal states, in the point of the switching the systems 70 T, the value of the initial internal states are converged and only one value can be taken to simulate the next model.

Proceeding is as follows: Use two different arbitrarily values at same distant point for the calculus of the initial conditions, and applying the past values of the control signal before crossing the limit for the next model, if these converged does not need to enlarge the distant point in time.

2.2.4 REPRESENTATION

The HS we are working with are systems with continuous and discrete variables interrelated, by the fact that discrete transitions will cause behaviour changes on the continuous variables. These systems present a great problem switching between local models if they have a different order. The methodology shown in this paper solves this difficulty by making all the models to have the same order, particularly, second order models.

$$\dot{X} = AX \quad + BU \qquad (2.2)$$

Applying a transformation over the state vector $g(x_1,x_2) : R^2 \rightarrow R$ for (2.2) we obtain from a two-dimensional SS a single parameter, which still maintains the dynamical information. Combining this parameter with the system output, a complete description of the system is obtained in a form of a graph in polar representation.

Definition 1. The partition of the SS into regions is defined according to the angle limits composed by the increments of the two components of the state vector.

$$g_1: \quad \alpha = atan\frac{\Delta X_1}{\Delta X_2} \qquad (2.3)$$

Lemma 2. This partition generates a representation of the valid regions in the state space, projecting and moving the same region into different zones over the trajectory.

Proof:

Proportional increments have the same angle

$\Delta X1 * k / \Delta X2 * k$ (2.4)

Different amplitudes can assume the same angle $\frac{\Delta(X1+k)}{\Delta X2} \rightarrow \alpha_1$

The representation in polar form, the radius ρ : the y output amplitude of the model and the angle: α the angle of variation of state space vector, gives as representation a set of superposed sectors, limited on angles α_{imax} for the limit of the local model validity, over groups of concentric sectors representing the outputs non-linearity's in regions $y_{1min} < y < y_{1max}$ (figure 2.2).

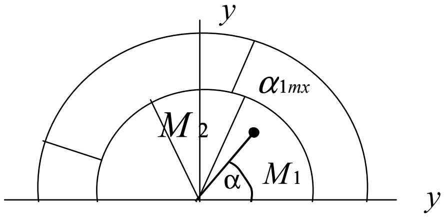


Figure 2.2: α-graph representation

The set of possible evolved sequences are, depending on the control *U(t)* : sequences of models with consecutive indexes, this means consecutively models $k+1$ or k-1, or with jumps $k \rightarrow 1$ according the abrupt changes over *U(t).*

Example:

A flexible belt transmission system $y(t) = -5\frac{dy}{dt} - 2\frac{d^2y}{dt^2} - \frac{d^3y^{\wedge}2}{dt^3} + U$ represented in three two-order models identified with the least square estimation LSE algorithm, with noise to help the identification. DS representation (figure 2.3) at two consecutive different steps, 1 and 1.5 amplitudes. The sequence of the models is truncated at response to the control signal.

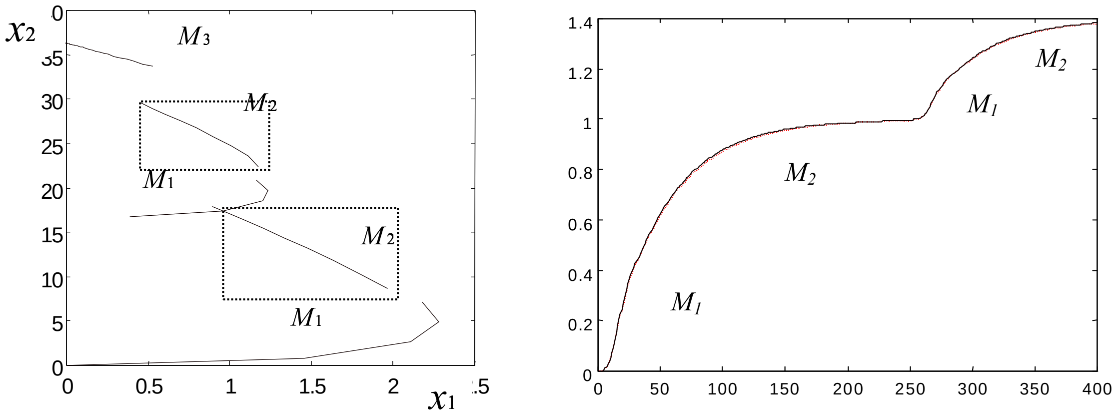
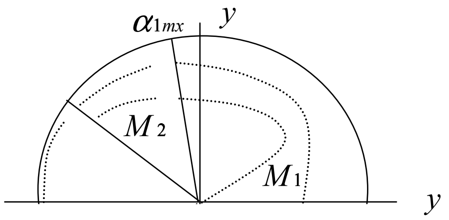


Figure 2.3: Example local models in SS, temporal response and alpha graph evolutions.



2.2.5 MODEL STRUCTURE

The model for modelled system has an interaction with the real controlled system. A structure of the model with relations over the processes is expressed with the following diagram:

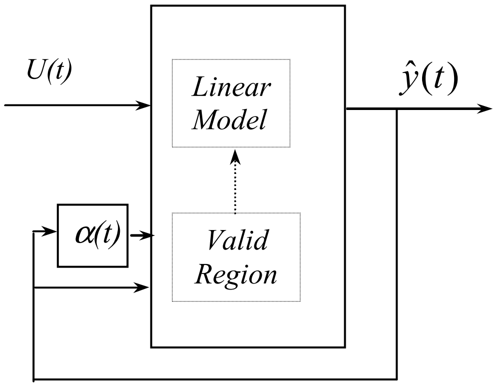


Figure 2.4 Alpha-Model structure

The inputs for the model are the control signal applied at the same time to the real controlled system, and the measure of the angle is estimated form the model in order to avoid the process noise. The outputs of the model are the estimation output for the controlled system, and the state vector used to estimate the angle. This angle is the result of the calculated angle from the both components of the state vector for the current linear local model used. The model selects the according this signals the local linear model used to model the process.

For MIMO systems the structure of the model is the following:

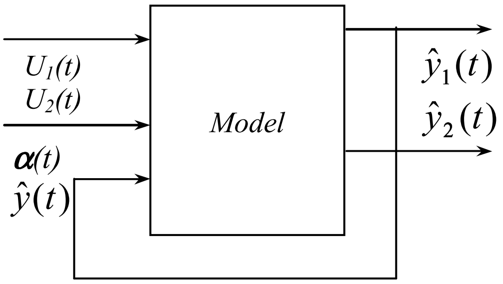


Figure 2.5 MIMO structure

In this case the measure of both angles are obtained from the two components of the state vector for each output. This produces two Alpha-graph, one for each decomposed MISO system $y_1(t) = f(U_1(t), U_2(t))$, and $y_2(t) = f(U_1(t), U_2(t))$. The approximate function is a LTI

MISO model, and the Alpha-graph is the same as SISO system, which makes partitions of the Alpha space by the output and the Alpha measure.

Interaction in the real process is the control signal *U(t)* and the output process measure, to recognise the noise, deviations of the model and instant of the event presented. The discrete variables are also measured in order to proceed to evolve the model over the graph diagram.

2.2.6 MODEL IDENTIFICATION

To reproduce systems by means of a mathematical expressions is largely treated and produces so many different expressions and methods to identify the system. ARMAX models for linear systems or NARMAX [Billings 87] for non-linear are able to represent observable models. The complex structure of the ARMAX comprises a great group of the systems. In the Hybrid systems identification the procedure is large and tedious.

Previously is to suppose one structure for the system, this is necessary to identify the model in each state. And choosing the correct data, this means the most representative, to adjusts the linear or non-linear model in each region, or in case a combination of linear model for the non-linear system. And the validation process, to assume all the previous process, or go back to change the structure or the model, all this is an extensive work over the process, that can be separated in a sequence of subtasks.

Taking into account that some processes need a previous sequence of actions to drive the system at the particular phase to be identified, this process can happen to be impossible for some systems. In this case all the knowledge obtained by the expert is necessary to predict the structure. Some time qualitative models are used for some untreatable process.

The methodology to model HS by using second order models is the following:

Identifying temporal behaviours by means a LTI second order model that satisfy $|y - \hat{y}| \le \varepsilon$ criterion. Where ε is the supposed maximum acceptable error prediction.

To model non linear stables system in a piecewise linear approximation, LSE algorithm or linear regression algorithm applied over different operating regions with a most representative data with same tendency, identifies this part of the system represented by the ordinary differential equations:

$$\begin{aligned} \dot{x}(t) &= A \; x(t) + B \; u(t) \\ y(t) &= C \; x(t) \end{aligned} \tag{2.5}$$

It is a second order model. A dynamic input U(t), for example different step amplitudes, can help to identify the system in several linear models, each region bounded at specified approximation error:

$$|y - \hat{y}| \le \varepsilon . \tag{2.6}$$

The simulation of the models in the SS will determine the regions' bounds within this acceptable error:

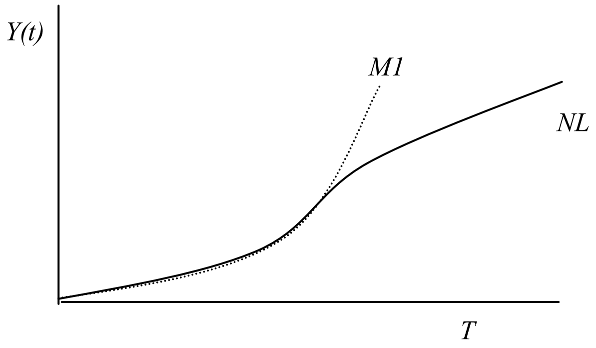


Figure 2.6 Validity of the model

The transformation $g_1 : \; \alpha = \operatorname{atan} \frac{\Delta x_1}{\Delta x_2}$ over $x_1, x_2 \in R^2$ the two components of the state vector, give the Alpha limits for the local model.

The Alpha limits can be superposed for collateral models, the sectors are overlapped in their extremes, this means both models are valid to model this part of the system. Using this system modelling the model only changes when it arrives at the valid extreme of the current local model.

This makes it possible to represent a high dynamic model in a combination of second order local models M_i. Some experiments at different input signals determine local models for a non-linear system. The conjunction of the models approximates the system in all possible trajectories.

2.2.7 CREATING α-graph

Continuous differentiable signals are identified like a combination of parametric two order models, sequenced these models depending on the evolved trajectory. Linear regression methods are applied in a different region for inputs-outputs data, in order to approximate each part to a second order linear dynamic model. This model approximates this region as large as the error model is confined into the limit. The parametric model is a virtual representation of this part of the system, linking the adjacent models and their evolution for the initial values of the state vector.

Once the first model is obtained a transformation of the state vector gives the angle of the system: α and also indicates the validity interval of angle for this model $[0-\alpha_{1max}]$.
The joint of the limits construct a complete circle with the combination of different models obtained recursively identifying linear models over input-output data. These intervals are not disjoint, the upper limit is not exact at the lower limit of the next model, depending of the used data in the identification method.

Different initial values gives different limits and models in the alpha graph, but also all the possibilities are valid because all local models obtained assures the admissible model error. The objective is to complete the circle with local models that represents the complete system in the operating region.

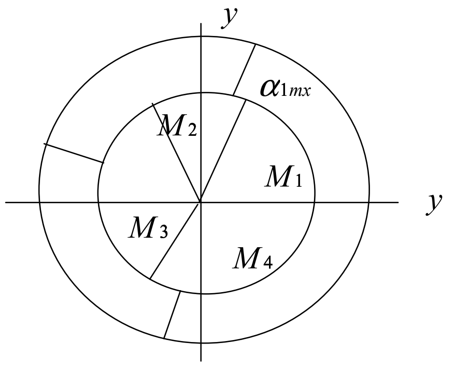


Figure 2.7 Alpha-Model representation

Static non-linearity's of the system is represented as superposed circles, approximating the output gain, dynamic ones as hysteresis , dead zones, also are represented by commuting regions in the graph, superposed circles and limit bounds collect this specification.

A complete representation is done when is determined all the models inside the circle Y_{mx} (figure 2.7) , where Y_{mx} is the maximal output possible for the system at response of different *U(t)* inputs signals, and α for all the angles from 0 to 360° , figure 2.7 represents a possible Alpha-graph for the non linear dynamical system.

This Alpha-graph represents one discrete state DS for the Hybrid system, each discrete state has one Alpha-graph similar to this. Discrete transitions interacts between Alpha-graphs and provoke jumps between these graphs (figure 2.8). A ψ and Ω transition function determines from departing region, the new used region in the new used alpha graph, and the new Starting point from the transformation of the final state vector X.

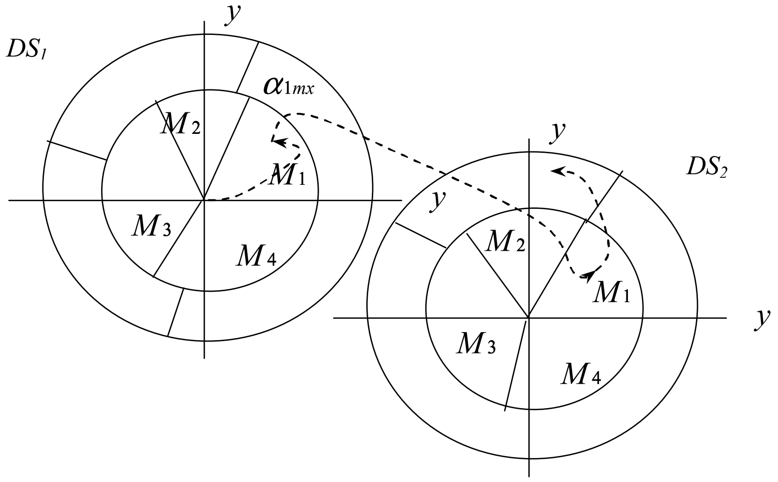


Figure 2.8 Interaction between Alpha-graphs

2.2.8 NOISE PROTECTION

Model is satifiable if the model error is confined in time and in successive models. Stable identified models assure convergence of the system reducing the initial error in time, this makes independence of the initial conditions error.

Possible range of initial conditions propagated along the model represents an interval of the end conditions for possible initial conditions to the next model. This interval must be lower than the model error in the next acceptable local model, this difference gives a possible range of $\dot{x}$, with limits $\leq \varepsilon$.

Where ε is the maximum acceptable error in the modellisation, and coming from the identification procedure. This margin is propagated into the model evolution, smaller limits propagation indicate the convergence error into the next model, the new extremes are inside the maximal error interval.

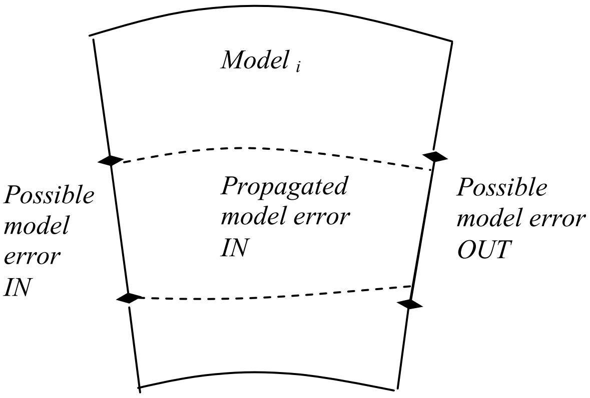


Figure 2.9 Model error propagation

When this condition is not fulfilled to accomplish, the control signal must be adapted to confine the error, in general a filter over the control signal is enough to smooth the signal and reduce the error margin for this local model.

Otherwise the effect of the noisy data is represented by rotation the model over a circle in the Alpha-graph. This can be detected and eliminated when the variation of the measure of the output amplitude for the system is inferior than the amplitude of the estimated noise and this avoids the evolution of the model. In this sense the model is protected from the noise, when the measured noise is previously determined.

2.4 LOCAL MODEL FUSION WITH TAGAKI-SUGENO MODEL

2.4.1 TAGAKI-SUGENO MODEL

Tagaki-Sugeno type III [Tagaki Sugeno 85]is a fuzzy model which include linear time invariant LTI models for each local model. A fuzzyfication of different LTI local model gives as a result a non-linear time invariant model.

$$\dot{x} = \sum_{i=1}^{p} \mu_i^*(x,u)(A_i x + B_i u + d_i) \quad \text{where} \quad \mu_i^* = \mu_i(x,u) / \sum_{l=1}^{p} \mu_l(x,u) \qquad (2.8)$$

A membership function μ_i determines the pertaining degree of the LTI local model i. Fusion of the weighted models μ^* weigh the evolution of the state vector.

In our case the membership function is governed by the output and the alpha measure, this converts the formulae to this:

$$\mu_i^* = \mu_i(y,i) / \sum_{l=1}^{p} \mu_l(y,i)$$

where p is the number of the models used in this approximation. And i is the index indicating the current model, this index came from assigning one index to every alpha-region. This index is used to preserve the limits of the local regions, and between them can be overlapping, jumps, hysteresis, and the index preserve them and avoids quickly commutations. This index needs to be represented by a state flow graph to indicate from the angle measure the model to use.

2.4.2 MODEL WITH SINGLE DS

Applied over the Alpha-graph for each local models this reproduces the non-linear model controlled by the angle of the state vector. SS regions are expressed as linear functions weighted by the membership of the angle and the output signals.

The parameters for a SISO local second order model in Matlab Simulink are:

$$[\ k\ \ -\xi\omega_o\ \ -\omega_o^2\ \ 0\ 0\ ] \qquad (2.9)$$

It is a fuzzy model with six inputs (2.9), the first one the weight input signal *U(t),* the second one the weight for x_1 ,internal continuous state, the third one the weight for x_2 ,internal continuous state, the forth the weight for the index measure of the model and the last is the biases by the output of the system.

The output signal of the model in a canonical form, is proportional to the x_2, and is also the integration of x_1.

The index measure of the model used for the pertaining degree of the local model are fixed by the measure of the Alpha and the output signal *Y(t).*

Control signal *U(t)* is an input to the model and the output as X_1 and their integration as X_2 ,are feedback the inputs to the model to give the dynamics for the system model.

Non linear in gain are adjusted by the membership function of the X_2 which is the measure of *Y(t)*, for the estimated output of the model, but it is preferable when it is possible to interacts with the system into the model by the real filtered *Y(t)*

The Alpha measure are taken from the $\dot{X}_1$ and $\dot{X}_2$ the continuous sates as a result of the formula:

$$\alpha = ta \qquad (\frac{\dot{X}_1}{\dot{X}_2}) \qquad (2.10)$$

Problem of this modelling is the difficulty to reproduce the hysteresis. To solve this restriction is needed to use a flow graph which transforms Alpha measure into an index to indicate the local model to be used. This represent the Ω transition function.

To soften the model transitions inside a DS, a filter over the index is applied to obtain intermediate values of the index for example: 1-1-1.3-1.6-1.9-2-2 which soft the transitions between the models by weighted models by the pertaining degree function.

The ψ-Table is represented as flow graph diagram, where the process events acts and evolves the model to the new situation. Is also responsible to enter the new initial conditions for the new active model.

2.4.3 ROBOT MODELLING

Non linear robot behaviour is well approximated by the Alpha approximation. Response to the step shows a deterministic disturbance of the robot, we assign this perturbation to the balance support to the earth plane.

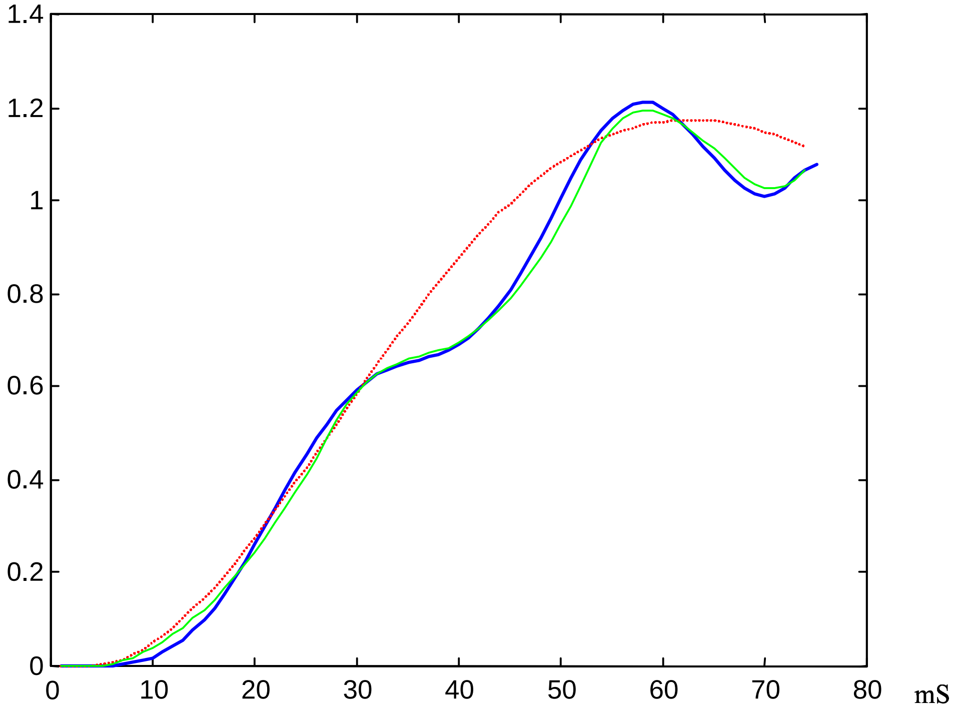


Figure 2.10 Modelling comparison

Velocity versus samples of time, at step response.

Blue, gross line: robot behaviour, filtered velocity,Red, dotted line: second order approximation, Green: Alpha approximation with three different models.

Models: M_1: `.6*U / (1 .78 1)Y,` M_2: `.85*U/ (1 1.17 1)Y ,`
M_3: `.92*U /(1 1.28 1)Y;`

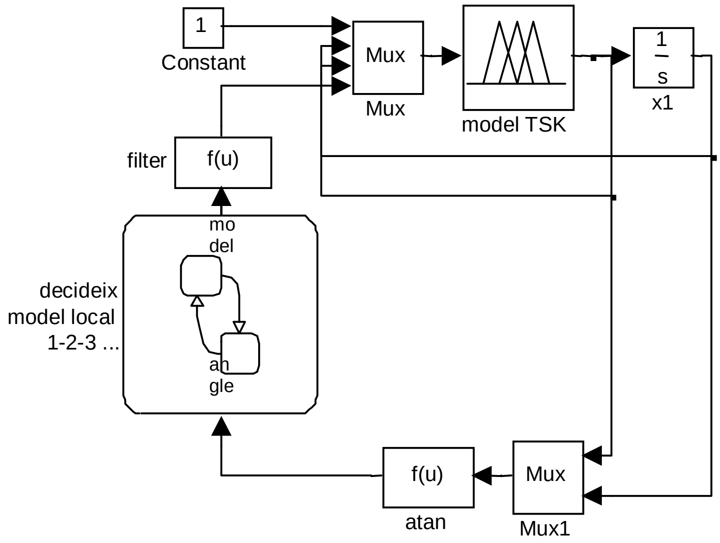


Figure 2.11 Simulink Alpha-Model

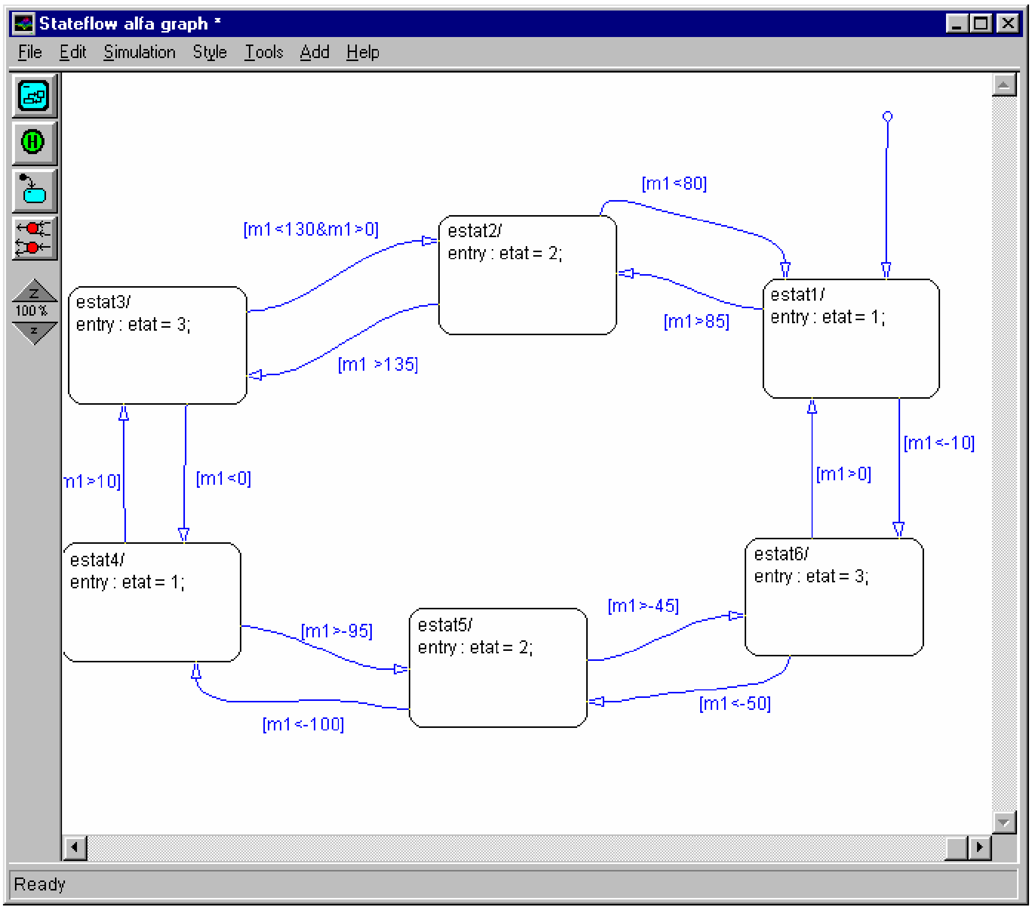


Figure 2.12 State flow diagram

2.4.4 MODEL WITH SEVERAL DS

Transitions between Alpha-plans DS are controlled by supervisor which determines into the ψ-Table the local model to be activated as a result of the fired transition. Supervisor interacts with a flow graph which is responsible to mark the local linear model to be used as the following schema:

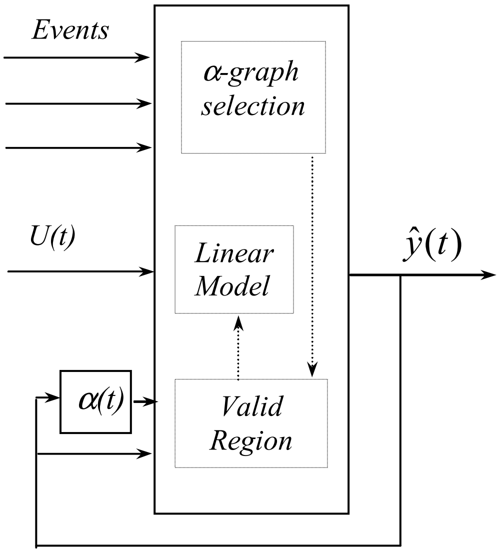
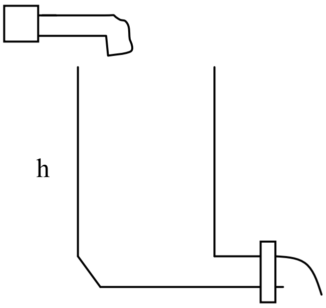


Figure 2.13 Alpha-Model structure

The fuzzy blocks are computing on line the corresponding output signal, a switch commutes the block as response to the discrete transition, and introduce them the initial conditions, the X_1, X_2 internal states to the new used block to assure the continuity of the output.



Example of the process with changes of the Alpha-graph:
The process treated is a filling tank with a leakage valve, at certain level the valve is opened and the model becomes non-linear. A modelling method approximates the linear and non-linear evolution of the signals.

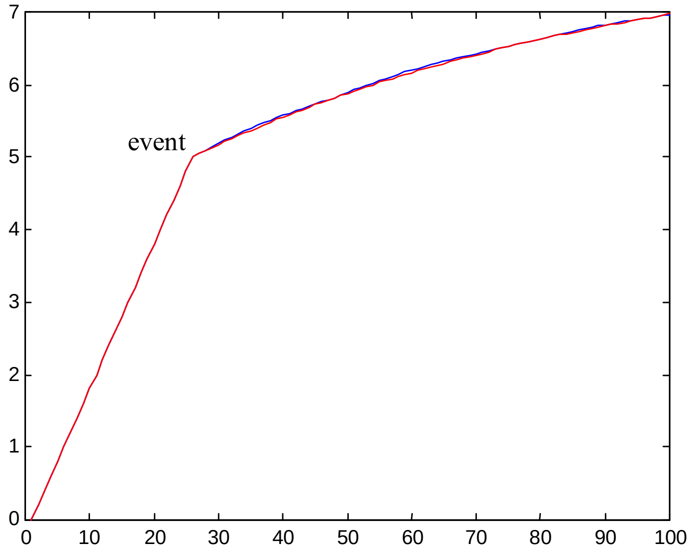


Figure 2.14 Approximation the h in red , and the level of the tank i blue, versus time, and change the DS at event presented

The model used:

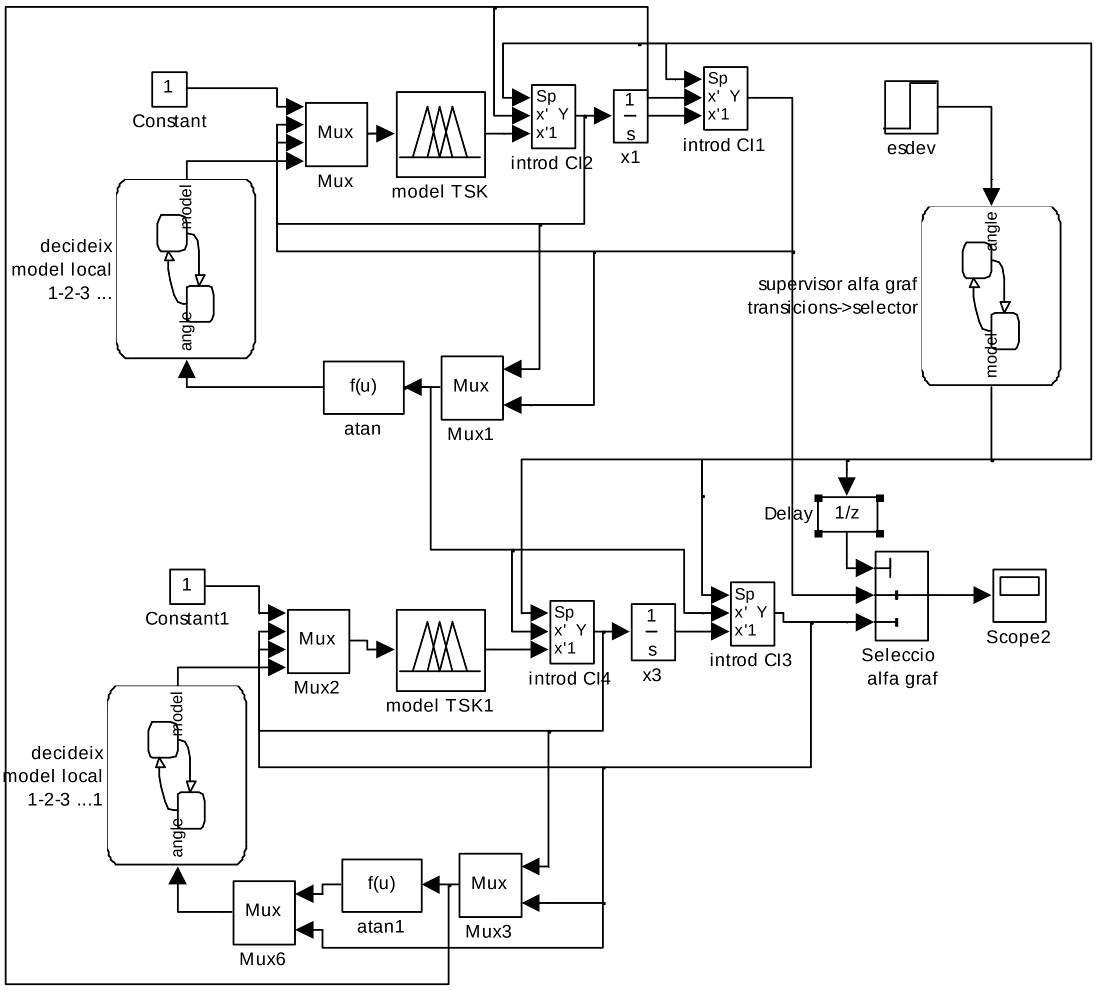


Figure 2.15 :Non-linear model using Alpha Tagaki Sugeno model approximation. Leakage valve at 5 seconds Ts=.2 seg.

2.5 EXAMPLE OF THE AUTOMATA TABLES

This is an example how to apply transition functions inside a computer, represented by a global transition table. Particularly this example uses only Ψ transition function to simplify the results, but is possible to include Ω transition function decomposing the non-linearity's into local linear models.

This example shows the effect over a system when the state is not completed represented by the information into the states, some variable are not measurable and it makes impossible differentiate them. Apparently the states are similar, but is possible to differentiate this states using information about the system past, this means memory states.

Also is a case when a non-deterministic situation can evolve the system towards a different evolution, and the past information help the control, to evolve the system into two different solutions, otherwise without the history the evolution will be the same, with less capability to control.

The situation is a tank with two inputs, and two outputs, a mixer, a heat resistance and PH, temperature and level sensors. The objective is to generate a mixed liquid with adequate PH under T_{max} temperature.

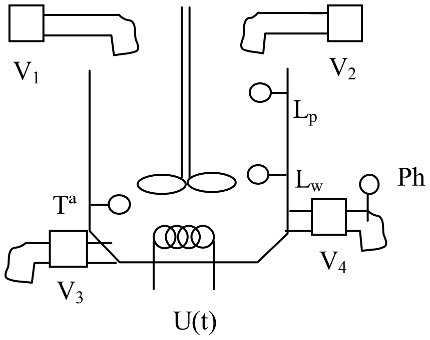


Figure 2.15 Process schema

The set of actions: V1,V2,V3,V4,Lw,Lp,M1, where Vi are the electric valves and Li level sensor and M the mixer motor.

The non-controllable events are Lw,Lp,Ph: level and PH sensors.

The states are: V1, V2, V3, V4, C, R and Empty, corresponding fill with valve V1, V2, empty with valve V3, V4, with continuous dynamics $\dot{h} = k_i$ or $\dot{h} = k_i \sqrt{h}$, and C to heat with continuous dynamics $\dot{t} = k_1 (T_{Max} - t) U(t)$, and State R mix and cool at $\dot{t} = -k_2 (t - T_{amb})$.

The history are added with the states V21,V22, to differentiate the evolution for the system. The continuous action is the applied energy to the heating system.

The possible sequences of the non-deterministic behaviour of the system controlled by the hybrid automata, are the sequences of events:

seq1': V1,V2,C,R,V4 Empty . Is the correct sequence to do the task. That means fill the tank with liquid 1 until Lw is reached and following the liquid 2 until Lp is reached, then heat until Tmax, and mix and remove until Tmin, and then open V4 to empty the tank during a large time.
seq2: V1,V2,V3,Empty , is the case when Lw is reached in a wrong situation (Leakage, source lack)
seq2': V1,V2,V3,Empty ,
seq3: V1 ,V2,C,V3,Empty , case when the Tmax is not reached (Heating system is damaged)
seq3': V1 ,V22,C,V3,Empty
seq4: V1,V21,C,R,V3,Empty, when Tmin is not reached
seq4': V1,V22,C,R,V3,Empty,
seq5: V1,V22,C,R,V4,V3,Empty , the Ph measure is incorrect.
seq6: V1,V21,C,R,V4,V21,C,R,V4,Empty. The Ph can be correctd by adding liquid 2.
seq7: V1,V21,C,R,V4,V21,C,R,V4V21,C,R,V4,Empty, adding aother dosis of liquid 2.

Difference between seq and seq' are the dosiffication of liquid 2, when the level adding the liquid 2 are optained are seq, and when the time for the liquid 2 is reached is in seq'. The history must to collect this information and V22 and V21 remembers this information.

Is generated a transition function Ψ as a state table with inputs: State, History, Events
And outputs: Next State, Control Actions: Continuous, Discrete, and new History

Historic	Current State	Event	Continuous Var.	Post State Actions	Control	New Hist
Empty	Empty	Start		V1		V1
V1	V1	Lp	Level	V2		V2
V2	V2	Tlim	Time	C	U(t)	V21
V2	V2	Lp	Level	C	U(t)	V22
V21	C	Tmax	Temp.	R	U(t)	previous
V22	C	Tmax	Temp.	R	U(t)	“
V21	C	Tlim	Time	V3		“
V22	C	Tlim	Time	V3		
V21	R	Tmin	Time	V4		
V22	R	Tmin	Time	V4		
V21	R	Tlim	Time	V3		
V22	R	Tlim	Time	V3		
V21	V4	Ph	Ph	V2		
V22	V4	Ph	Ph	V3		“
V21	V4	Tlim	Time	Empty		Empty
V22	V4	Tlim	Time	Empty		Empty
V21	V3	Tlim	Time	Empty		Empty
V22	V3	Tlim	Time	Empty		Empty

Table 2.1 Transition table for the Tank process

The control acts over the system looking into the table the current situation and the actions to make over the process, discrete and continuous actions. And up-to-date the new sate and history of the process.

Representing schematically the evolution of the process, that can be represented by the following schema, and is identified two possible evolutions with common parts, depending of the past behaviour.

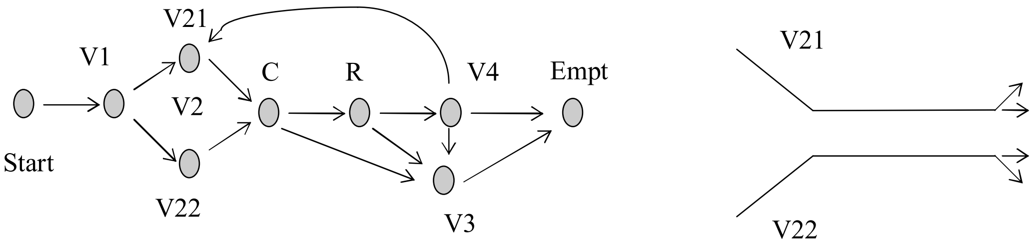


Figure 2.15 : Possible evolution of the process.

Particularity of this example is the evolution of the process that can be predicted in several situations. This can be collected using the information about the history of the process evolution. This represents the non-deterministic dosification system expressed by the non-operation dosis applied over the tank, but in succesively attempts the dosis can be well applied over the liquid and the problem resolved. Otherwise an excess of dosis is a problem non-resolvable and the solution is to empty the tank and start the process from zero.

Chapter 3

3.1 CONTROL

3.1.1 INTRODUCTION

Hybrid systems control is a huge task, the processes need to be guided from the Starting point to the desired End point, passing a through of different specific states and points in the trajectory. The system can be structured in different levels of abstraction and the control in three layers for the Hybrid systems from planning the process to produce the actions, these are the planning, the process and control layer.

The planning or scheduling layer is the top layer, in this layer is analysed the global process and is determined which methodology is applied to reach the objective. The order of the subtask, their restrictions and establish the cost function to optimise are the goals of this layer. Supervision task is present in this layer when this capability exists.

The process layer are determined from the restrictions and the objective cost function, the local objectives or set points for the subtask. The local restrictions and objectives for the local models fixes a controller and timings to complete this part of the global objective. This layer is responsible to synchronise and for sequencing the actions.

In the control layer are executed the proposed actions. The local control of the state models drives the system from one set point to the next. Precision and stable conditions are the premises for each local controller.

In Hybrid systems these three layers are presented, from Starting State to the End point in the End State. This requires decomposing the global objective to several local objectives, pointed by states. And inside these states, local linear models decompose the state and adjust the controller to acquire the goal . Local controller applied over these local linear models is obtained by one methodology that solves the global cost function criteria. In some cases Minimum Time solution, Linear Quadratic Regulator or simple Proportional Regulator can be controllers to apply in response at different objectives function.

3.1.2 CONTROLLABILITY

Some authors have treated the controllability problem [Nerode Kohn 93][Stiver Antsaklis 94][Tittus Edgard 94] for Hybrid systems, and definition of this problem is the capability to guide the system from one state to another. In this work the problem is divided into two subparts, the discrete and continuous controllability, combination of both premises mark the capability to drive the system from a specific initial state, and initial state vector, to the end state and end state vector.

Continuous controllability is known as the rank of the controllability matrix of the LTI systems, this marks the dependency of the outputs from the inputs, indicating the controllability of the system. For a second order models SISO system, these are controllable for stable systems. MISO systems are generally controllable in industrial systems. Analysis in a special cases can inform of this system non-condition.

Discrete controllability is the capability to drive the system from one state to another in a finite language, this means finite transitions. The discrete events system DES theorists analyses this capability in different methods, we use the Ramadge-Wonham theory [Wonham 97] and model description to analyse this capacity.

Interaction between discrete and continuous models can give erroneous results in the controllability criteria, strongly coupled systems need a specific analysis to conclude the controllability capacity.

3.1.3 HYBRID SYSTEMS CONTROL

Control of discrete systems is largely treated, from [Lafortune 92], Ramadge-Wonham over automata to Petri Nets. All are discrete controllers for discrete systems. Other authors treat the problem to control a continuous system by the discretized controller , a computer.[Antsaklis Lemmon 93]. And other group of authors treats the problem to control a Hybrid systems by means of a controller which gives continuous and discrete signals, [Lenartson94 and Deshpande-Varaiya95]. In this work the problem to control a Hybrid system with hybrid controller is treated.

The objectives of designing a controller over Hybrid system are, to find a viable and safe path to reach the goal. This term means: Safe path is the path which avoid forbidden states, forbidden are those which blocks the system, which one the system stops the evolution, this means, the system does not present any more events. Or forbidden states are those, which the interest for the controller is to avoid them. And viable path is the reachable path for the system, this means, there exists control signals to evolve the system from the Starting point to the End point in finite time.

The verification of the Hybrid system assures reachability for the solution, safety and liveness for the system. This is useful for synthesising safe controllers [Puri and Varaiya 94].
Verification of the solutions determines the right paths that control the system, reaching all the predicates in the transition states and the goal's end conditions.

Brajnik and Clancy explains a methodology [Brajnik Clancy 97] for synthesising, under uncertainty, a sequence of robust and discrete control actions to drive a continuous dynamical plant through admissible trajectories specified via temporal logic expressions.
The TeQsim qualitative simulation tool [Kuipers 94] uses temporal logic to model checking, validate or refine proposed control plan. TeQsim express behaviour trees representing the possible behaviours for the controller under a control law. Plan refinement infers bounds on the sequence of actions within a plan to guarantee that the specified goal constraints are satisfied.

Other point of view is to express a system with continuous and discrete parts as qualitative models and find the control reasoning over. [D.Brajnik Clancy 97][Lunze Nixdorf 97]

The control problem in this work is treated as follow: the control block generates a joint set of possible solutions to control the system from a Starting point to desired End point, which assures the controllability premise, a further analysis over this solution can obtain which one solves the restrictions over the system trajectory, the minimum cost, the safe path etc. At the end are obtained the crucial points as the set-point for the system to follow the optimal solution in order to minimise the cost function.

3.1.4 CONTROL STRUCTURE

The structure of the control into the process is indicated in figure 3.1. It is a hybrid controller, which uses the continuous and discrete information about the system, and acts over them with the same kind of signals.

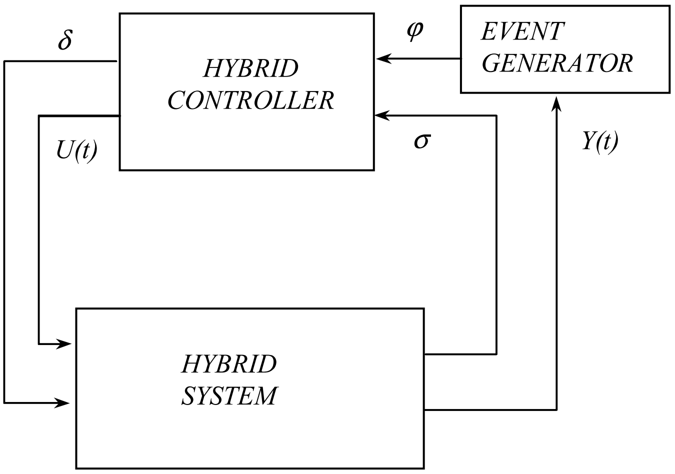


Figure 3.1 Block diagram for controlled system

The tasks which will be of interest for the controller are to:

- Know the state of the system
- Establish the way under a specified cost function
- Enclose the system into End point direction

These tasks are interesting for the controller man and it will be possible to implement into the controller automatism.

The means which have to realise the tasks are

- Discrete signals, as actions
- Continuous control
- Models and predictors.

In figure 3.1 the Hybrid system produce discrete events σ and continuous signals Y(t). Hybrid controller interacts to the system by continuous and discrete signals : U(t) and δ which makes the evolution of the system in a desired sense.

The hybrid controller is formed by the following parts:
Event generator : which converts the continuous signals in discrete levels and events φ in the region transitions. Includes the Alpha-graphs representation.
Hybrid controller : which drives the system by actions and control signals to the desired End situation. This block has two subparts:
Discrete model : which represents the knowledge of the system, represented by the state diagram.
Transition tables : created to drive the system to the destination point, under restrictions and goals.
These parts are defined to control and supervision tasks, a complete model of the system is suitable for fault detection and accommodation.

The model is designed with all the possible information of the system, continuous and discrete parts are represented in a several Alpha-graphs. The evolution into the Alpha-graph is represented as the adjacent states in the graph, and generates the φ events to make evolution of the transition tables. Transitions between Alpha-graphs are collected into the transition table, and marked by the inputs events, the discrete events generated by the HS.

The tables are created once the global goals are defined, cost function, restrictions, and desired End point. The table makes the evolution of the system by actions, discrete events which produces actions to the system and forces the evolution of the system to the next state. At the same time this evolution is reflected into the model.

It is possible to control the system without any model, but it is necessary for supervision task of the system. Any deviation of the desired path is informed to the supervisor which take the control task in order to realign the system in the right direction.

We present a methodology to solve the control problem finding the optimal controllable path respect to a performance measure. Different software tools are used to obtain a continuous abstraction, a controllable paths and the optimal control for a class of hybrid dynamic systems.

3.2 REACHABLE SET AND TRANSITIONS

3.2.1 REACHABLE SET

For linear 2-D SISO unconstrained system the reachable domain represented in state space is all the plane, this means it is completely controllable, there exists a signal of control U(t) which drives the state vector to any point of the phase plane. This is a mathematical expression of the unknown system. The real systems presents restrictions in the control signal and in the state vector.

This controllable system with restrictions can reach a certain domain in the state space:

Consider the following SISO linear system (3.1)

$$\begin{aligned} \dot{X} &= AX + BU \\ Y &= CX \end{aligned} \qquad (3.1)$$

With $X \in \mathrm{R}^2$ and A stable, and U within limit bounds $|U(t)| \leq U_{max}$ (3.2)

and X components confined into the maximal values, according to the physical limits $X_1 \leq X_{1max}$, and $X_2 \leq X_{2max}$ (3.3)

The system evolves in time, towards the stationary solution for stable systems when stabilisation time has been reached. The union of the maximum limits of different trajectories gives the reach set of the system when *U(t)* and *X* is in a interval range.

The reach set of the system with C=[0 k] can be :

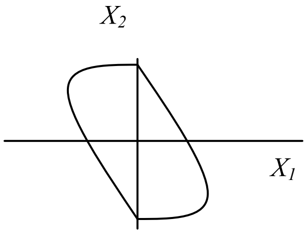


Figure 3.2 : Reachability set for a state

With X_1 and X_2 are the state vector and limits are the maximum response to the Bang-Bang [Kirk 70] control. Inside the region there are all of the reachable states in finite time for this system (3.1).

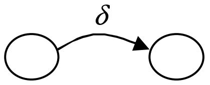
3.2.2 REACHABLE TRANSITION

Reachability analysis of the system must consider two possible situations for Hybrid systems, either reaching the bound limits in the transition conditions or reaching the End point.

In the first case, the intermediate state reachability analysis determines if the system crosses the bounds limits of the linear predicates (3.3), which leads the system to the next state. Linear predicates are compound by linear inequalities of the State vector with first order logic (*): combinations of ∧, ∨, ¬ logic expression. With $K_i \in$ R.

$$\delta : k_1 X_0 \geq k_1 * k_2 X_2 \geq k_3 \qquad (3.3)$$

$$\begin{aligned} X_i &= (k_4, k_5) \\ X_f &= (k_6, k_7) \end{aligned} \qquad (3.4)$$



Theorem 1: If part of the reachable set (figure 3.2) is superposed with the region of (3.3), then there exists a finite control signal $U(t) \in [U_{min}, U_{max}]$ which activates the condition transition (3.3).

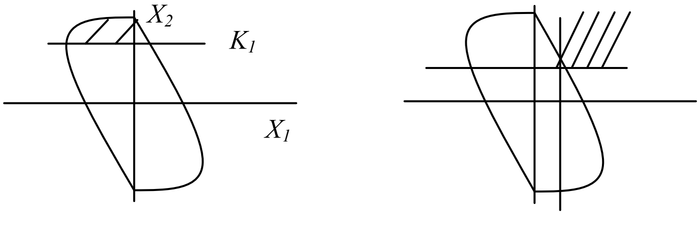


Figure 3.3: predicates δ : $X_2 > K_1$ and $X_1 > K_1 \wedge X_2 > K_2$

In the second case, the reachability points in the End State is to be defined as the ability to reach the goal X_f (3) on the Hybrid system.

Theorem 2: If the end condition is inside of the reachable set (figure3.4), then it exists a finite control signal $U(t) \in [U_{min}, U_{max}]$ which translates the system (3.1) from some initial condition in the state to the goal X_f(3) of the state space.

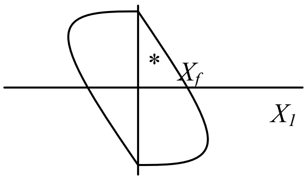


Figure 3.4 Reachability End condition

Proof: The continuous controllability condition, assures the control signal solution to reach any point into the reachable set from any other point.

Safety property is assured if there is no solution to any non-permitted state (Figure 3.5). For this analysis is used a verification tool, HYTECH [Henzinger Ho 95], studying the reachability of the transitions for this non-permitted state. Approximation for the second order models is done in order to use this util, limited to continuous dynamics with constant variation. In Appendix 1 an example shows the procedure and results.

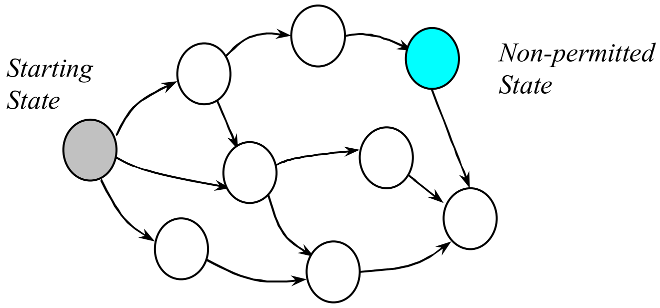


Figure 3.5: Diagram with reachable transitions for non-permitted State

3.3 CONTINUOUS ABSTRACTION

3.3.1 PROJECTION

In the Hybrid systems control, the goal is to find an admissible and controllable path between the initial point in the Starting state and the goal End conditions. A controllable system, is a system which enable to drive through one path towards the goal, with different kind of events in the way.

To apply the RW theory [Wonham 87] of discrete event systems on a Hybrid system, we must abstract the continuous signals and to project the events generated for these signals, to obtain new sets of observable and controllable events. Ramadge & Wonham's theory along with the TCT [Wong 96] software give a Controllable Sublanguage of the legal language generated for a Discrete Event System (DES).

Continuous abstraction transforms predicates over continuous variables into controllable or uncontrollable events, and modifies the set of uncontrollable, controllable observable and unobservable events. Continuous signals produce into the system virtual events, when this crosses the limit (3.3). If this event is deterministic, they can be projected. It is necessary to determine the controllability of this event, in order to assign this to the corresponding set, $\{\Sigma_c, \Sigma_{u\,c}, \Sigma_o, \Sigma_{u\,o}\}$, controllable, uncontrollable, observable and unobservable set of events.

Controllable events are those events that can be activated or deactivated by the controller. Actions can activate some events. The set of the controllable events by the controlled actions is Σ_c .

Uncontrollable set of events are those produced by the system as response to some precedent action. It is called non-controllable, because is not the direct response to the stimulus it depends on the process. And they can be altered by perturbations over the system.

Observable set of events are those that can be recorded by sensors, and the controller can react front these events. And unobservable set of events are those for that there have not information about their state or value.

The set of uncontrollable events has to be projected to another set of uncontrollable events, changing the indirectly uncontrollable events, which they are continuous reachable by the continuous control U (t), to the set of controllable events.

$$\Sigma_{u\,c} = P(\Sigma'_{u\,c}) \qquad (3.5)$$

These events are those that the region generated by the linear predicate (3.2) crosses the reachable set of continuous reachable domain.

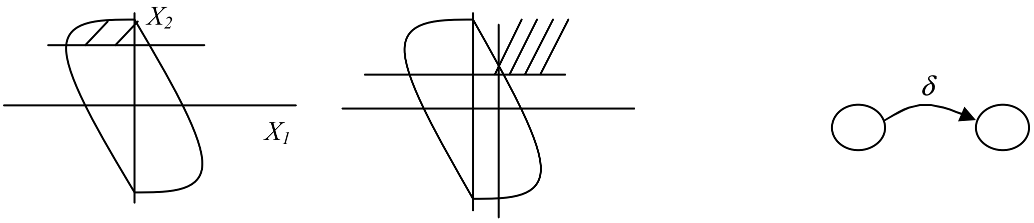


Figure 3.6: predicates δ : $X_2>K_1$ and $X_1>K_1 \wedge X_2>K_2$

This means, the uncontrollable event passes to the set of controllable event, because the event is to response to the continuous control U(t), which produces the event at controllable form.

States with more than one uncontrollable transition can change the behaviour of the graph, when the events can be converted into controllable events, after this transitions becomes deterministic.

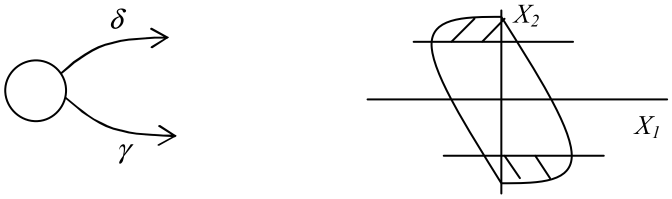


Figure 3.7: predicates $X_2>K_1$ and $X_2< K_2$

The projection over the unobservable events, generated by continuous signals, eliminate from the set of unobservable events those events that are identifiable by the continuous signals analysis [Lemmon and Antsaklis 93].

$$\Sigma_{u\,o} = P(\Sigma'_{u\,o}) \qquad (3.6)$$

Unobservable events also can be detected when the state is out of the reach set of the current model, indicating the use of the other reach set, this means other model, other state. It is not observable in continuous sense if is not possible to determine the state change instant by identification methods. Otherwise it is observable in discrete sense when the state crosses the reach bounds.

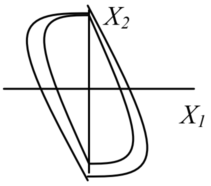


Figure 3.8: Observability of the state

At the end the transition is true unobservable, if the event is not presented, and it is unidentifiable by continuous identification, and does not go out of the reach set of the current model.

Conditions of the projection the sets are disjoint sets (6).

$$\Sigma_c \cap \Sigma_{u\,c} = 0 \;\; , \;\; \Sigma_o \cap \Sigma_{u\,o} = 0 \qquad (3.7)$$

3.3.2 CONTROLLABLE PATH

The RW theory [Wonham 87] defines a Controllable Sublanguage, with respect to a language composed of erased unobservable events which accomplish the following expression:

$$\overline{K}\Sigma_u \cap L \subseteq \overline{K} \qquad (3.8)$$

This means that uncontrollable events presented in some prefix of the Controllable Sublanguage $\overline{K}$ which is formed by the union of the prefixes of maximal K the way of control, and permitted for the language L of the system, are part of the Controllable Sublanguage, language generated by the system under the controller. A string u is a prefix of a string v if for some w , $v=uw$ [Ramadge Wonham89], and is closure of $\overline{K}$, if $\overline{K} = \{ u : uv \in \overline{K} \text{ for some } v \in \sum^* \}$, which $\sum^*$ is the set of all finite strings of elements of the set of events $\sum$

The Supremal Controllable Sublanguage of K is represented by $K^{\uparrow}$, and is the result of the operation:

$$\Omega(J) = K \cap \sup \left\{ T : T \subseteq \Sigma^*, T = \overline{T}, T\Sigma_u \cap L \subseteq \overline{J} \right\} \qquad (3.9)$$

Then $K^{\uparrow}$ is the maximal sublanguage of Ω.

All the valid trajectory's from Starting point to the End point which accomplish the Controllable Sublanguage are the possible ways to drive the system, and we called this a set of Reachable Way's RchW. A further analysis is applied over them to choose the adequate to make evolve the system.

3.3.3 MODULARITY

To reduce the complexity of the control problem it is possible to modularise the control structure. Independent subsystems or interrelated modules can be analysed apart if these parts accomplish some requirements.

In term of the controllability conditions as specified above, the intersection of the closed languages $\overline{K}_1, \overline{K}_2$, are controllable if there exists compatibility between language intersection and controllability. This compatibility is mathematically expressed as non-conflicting languages $\overline{K_1 \cap K_2} = \overline{K}_1 \cap \overline{K}_2$, this means this intersection language satisfy two constraints $\overline{K}_1, \overline{K}_2$, expressed as desired controlled language.

The Supremal Controllable Sublanguage of $K_1 \cap K_2$ can be found by computing the Supremal Controllable Sublanguages of $\overline{K}_1, \overline{K}_2$ and checking that these languages are non-conflicting , and if so, forming their intersection.

3.4 MINIMUM TIME TRAJECTORIES

Find optimal trajectories in order to minimise some cost function is the goal of the modelling procedure. Mathematical model for the system allows the user to apply mathematical techniques over this expression. These possibilities are, to minimise a specific cost function, to obtain optimal controllers and to approximate a specific trajectory.

Interest for us is to obtain controllers and obtain the trajectory that minimises a specific cost function. The cost function can be minimum time, minimum final error, or control effort.

Usually optimal controllers are obtained by the solution of the general cost function: $J = \int \left(X'QX + U'RU + X'NU \right) dt$ where the cost of the state vector and the cost of the control effort are expressed and weighted by matrices *Q* ,*R* and *N*. This formula applied over a LTI model gives as a solution the Ricatti equation and obtains a Linear Quadratic Regulator K which minimises the cost function proposed.

Different theories and researches communities attach optimal trajectory problem for non-linear systems by different methods : Strictly Non-linear Optimal Regulators: Poyntraguin Maximum principle [Pontryagin 93], is the mathematical principle of variations to optimise non-linear models. Hamilton Jacobi Bellman equation [Bellman 71], is the solution for Non-linear multi-stage systems. Our case diverges from these authors due the linearity of the model. The linealising of the non-linear models allows us the use of simpler techniques.

Local linear models creates a combination sequences of linear functions, this admits the use of Dynamic Programming Convex optimisation in order to minimise a cost function over a sequence of models. The sequence is not pre-determined, they will be done using the Bellman Optimality Principle. The Bellman Principle decomposes a great problem into sub-problems. The subparts are those that have a starting and end stage specified, and the intermediate stages are free. Solvable subparts helps to determine the global solution for the problem using the principle of any subpart, are part of the global optimal solution.

Minimum time problem are the most interesting problem in Hybrid systems, [Shedlund Rantzer 98]. The solution of the minimum time problem is the Bang-Bang control for controllable linear SISO systems.[Kirk 70] The interpretation of the Bang-Bang control is to apply maximum power to approach at a desired point, and next to this point apply minimum control action to reach the exact point.

The combination of this result and the Dynamic Programming with Bellman Principle of optimality, give us the procedure to solve the minimum time trajectory for Hybrid systems. The problem is greater when there exists interaction between adjacent states.

In Hybrid systems the problem is to determine the partial set points to be applied at the local models. Optimal controller can be implemented in each local model in order to assure the minimisation of the local costs. How to decompose the global objective and reach the goal with minimum cost are the problems treated in the following part.

The cost function generally we apply over the Hybrid system is the time optimal control, this is the minimum time to reach the goal. The solution of this problem need to give us the trajectory to follow the system. Trajectory marked by a set of set points to force the system to passing over them.

In minimum time problem the performance measure is:

$$J = t_f - t_0 = \int_{t_0}^{t_f} d\,t \qquad (3.10)$$

With t_f the first instant of time when $x(t)$ intersects the target set in the state.

Dead Beat controllers are the optimal controller to the minimum time problem, and are interesting for the LTI MIMO systems with several internal states, which gives the solution for the control signals at each discretized time interval to be applied over the LTI system.

We obtain time optimal trajectory by quadratic programming from the End point to the Starting point, with a set of the second order local models, determining global trajectory as union of local optimal trajectories market in a path.

The applied Bellman equation: $V^*(x) = \min_{p\in\Pi}\left\{g(x,p) + V^*(x'(x,p)\right\}$,

Where Π is the set of states and $g\ (x,p)$ is the cost of the travel local state, and $V^*\ (x')$ the optimal cost of the rest.

The cost function $g\ (x,p)$ is cost of the p model in discrete state space form: $\min k \mid \sum_{i=1}^{k} \Phi(-iT)H(T)u_i - x_g \le \varepsilon$ which is the minimisation of the final state from zero minus the goal state. The index *k,* which minimises the final error is the time cost of this part of the trajectory. How to obtain the solution under the limits $u_l \le u_i \le u_p$, represents a convex problem of linear multi-criterion optimization.

Once the mathematics principles are defined we apply this methodology over a complex Hybrid system. A controlled Hybrid system is represented by a complex net of states with a starting state and end state. The continuous dynamics in the states are linear, and some of them in closed loop. The applied controller can be a proportional regulator to requirement of the LQR problem.

Several ways are possible to drive the system from the Starting point X_i to the End point X_f. Different ways are interesting in: dynamic sense , minimum states, approximation at set points, etc. These ways need to be safe and viable and RchW. And only one of them must to be applied, normally the best, which minimises the cost function proposed. A Reachable Way, this means the controllable way and safe, will be evaluated in order to obtain which one minimises the cost function.

Models with complete controllable state assures a RchW solution, to control and drives the system over a specific local model, able to jump with a discrete virtual transition V_i ,and discrete transitions δ which jumps to the local model in a new DS.

In respect to the virtual transitions the optimal trajectory is obtained when this trajectory does not pursue the corner of the regions, in this case the group of the models used must change, including the contiguous model of the corner not used and excluding any used model. At the end the models for which the optimal trajectory flows are obtained.

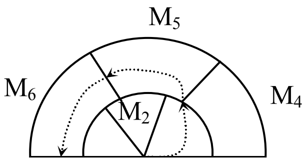


Figure 3.9: Optimal Trajectory obtained by the use of the erroneous set of models.

The interaction between states prohibits the use of local state optimisation. The solution is obtained taking recursively the adjacent pairs of states [Esteva 00], resulting in the optimal solution for the linear or quadratic performance measure of the problem.

In each pair of states the discrete transitions δ, defined for *Y* limit, the $\dot{Y}$ in this limit and the control signal *U(t)* are calculated by dynamic programming (DP) in order to minimise the global time cost. The problem is combined of three parts, the first is the cost of the first model to approach the system to the limit, the second part is the first model to cross the limit, and the third part is the second model to go through the new limit. The applied signal *U(t)* in each part is the Bang-Bang (Kirk 70) control in order to optimise the performance measure Time. In this sense the global optimisation is obtained as the local optimisation of the adjacent states.

$$\min_{k+l+j} \mid \; x_o + \sum_{i=1}^{k} \Phi_1(-i)T H_1(T)u^+ + \sum_{i=1}^{l} \Phi_1(-i)T H_1(T)u^- \geq x_{f1}$$

$$\wedge \; x_o + \sum_{i=1}^{k} \Phi_1(-i)T H_1(T)u^+ + \sum_{i=1}^{l} \Phi_1(-i)T H_1(T)u^- + \sum_{i=1}^{j} \Phi_2(-i)T H_2(T)u^- \leq x_{f2} \qquad (3.11)$$

Every pair of states need to reach one condition, the amplitude or alpha measure α, is one of the limits for the model into the alpha graph, the other are free, this means, when the amplitude of *Y* is fixed to change the model, the other parameter, the α are free and adjusted in the minimisation procedure. The iteration procedure determines the optimum taking into account the interaction between adjacent models.

Example :

The heating system. The process has the following sequence: Heat the liquid in the tank until it reaches 85 Celsius degrees. Once the temperature is reached the two additives are melted. The last step is wait for the temperature to fall until 35 degrees to be able to packaging the substance.

The goal of the control procedure is to minimise the total time of the sequence. And by in this case minimising the time, the control energy is also minimised.

Restrictions of this process, are: The melting time is fixed and 2’ time. The control action is over the heating system. The heating system has inertia for the big resistor.

The model of the heating system can be represented by the following expression:

Temperature rising: Second order model, with coefficients numerator: 0.00028, and denominator 1, 0.0029, 2.8 Exp-6 with Step response:

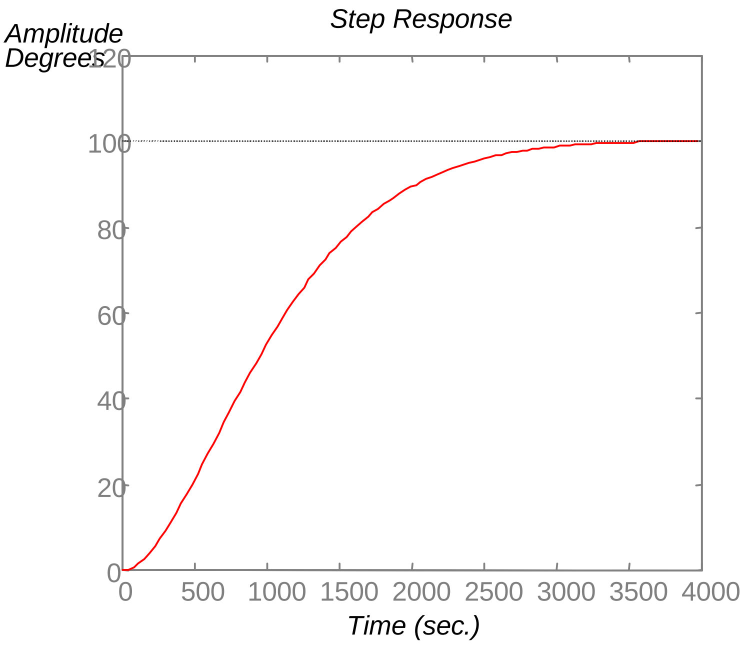


Figure 3.10 : Step response of the system

And temperature descending: Second order model, with coefficients numerator: 0.00027, and denominator 1, 0.0029, 2.7 Exp-6 due to the change of the radiation coefficient for the liquid with additives.

The process sequence can be expressed by the following expression, which is formed by the rising part of the temperature, and the descent part. The first one is composed by the rising part and approach part to the transition, at response of the minimum time problem, which gives as control the Bang-Bang solution, it imposes the two control signals applied U_{max} U_{min}. The second part is the melting liquid and free evolution without energy supply. The following formula is to be minimised, with free parameters *k, l* and *j* and gives one global minimum.

$$\min_{k+l+m} \mid A_1^{k+l+1}x_0 + \sum_{i=1}^{k}(A_1^iB_1u^+) + \sum_{i=1}^{l}(A_1^{k+i}B_1u^-) + B_1u^- \geq x_{f1}$$

$$\wedge \mid A_2^{m+1}x_{k+l} + \sum_{i=1}^{m}(A_2^iB_2u^-) + B_2u^- \geq x_{f2} \qquad (3.12)$$

A_i and B_i are the models of the temperature evolution in state space in discretised form. The initial conditions for the rising part are 0. The initial conditions to the second part are obtained from the end conditions of the first part. The x_{f1} is the temperature raised in the first state, 80 degrees, and the x_{f2} final condition 35 degrees.

Results:

If the control is at response to the sensor temperature reaching 85º the total time to reach 35º is 2878 seconds. Otherwise 2820, this is as response to the inertial temperature, which overpass the maximum temperature required.

CF=[85 .0033] CI=[85 .0034]

Switching points: `1480; 1604;2703`

Original points `:1548, 2778`

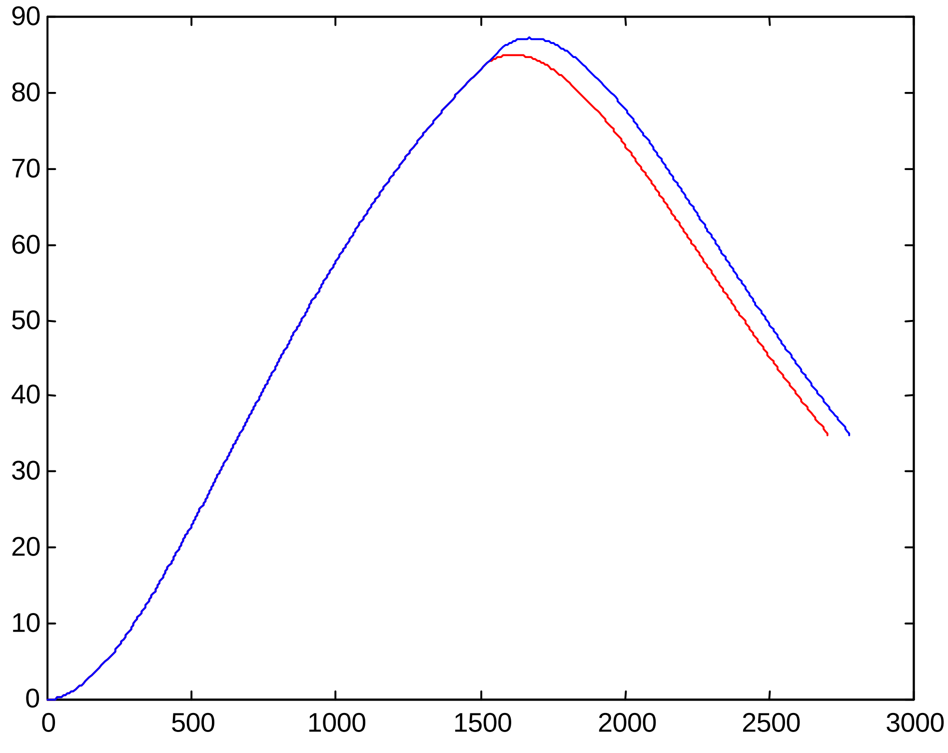


Figure 3.11: Response of the controlled system.

3.5 NON-DETERMINISM

Non-determinism in a system exist when from one state an event, transition sometimes to a different state (q_1-b-q_2, q_1-b-q_3) (figure 3.12 a). In a control sense this kind of the systems is presented when exists more than one uncontrollable event from one state (figure 3.12 b) and the perturbation not measurable forces the system to one direction. Several uncontrollable events from one state cannot help to predict the behaviour of the process. In this case the probability can be used to predict the most popular behaviour.

Figure 3.12: a) Non-deterministic system b) Non-determinism in a control sense

The non-determinism cases are pre-treated to the following procedure:

In the cases with states with more than one uncontrollable event in the controllable path, the optimisation procedure analyses all possibilities, and the result for this RchW is the worst or the best solution for the cost function. A conservative solution would pay attention to the worst solution, to assure the minimal condition to be accomplished. And the optimistic solution pay attention to the best non-deterministic Way, but taking consideration of the possibility to reach the other solution.

When it is necessary to accomplish a certain condition, the worst solution guarantees to reach the goal under the specified conditions, but the automata must to prevent all possible actions to control the system in desired direction.

EXAMPLE:

A process in a two tank system is the following (Figure 3.13):

Firstly the stage is to fill the first tank F, and the next is to add a concentrate in the water M, once the Ph is adequate, add another substance in the second tank S.

The non-determinism appear in the Ph stage, if the Ph is less, can add more concentrate, if this is too high the tank is emptied E and the process starts again. There is no one sign to determine if the Ph will be reached. It is a process clearly non-deterministic in control sense, and the controller must act in both cases.

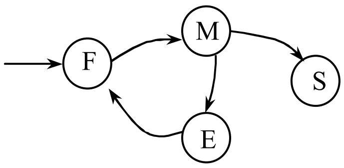


Figure 3.13: Two tank system

3.6 OPTIMISATION EXAMPLE:

A simple example to show the possibilities of the optimisation procedure over HS is shown next. Results are not very impressive due to the simplicity of the problem.

Let us consider a car with the following speed model, in Km/h :

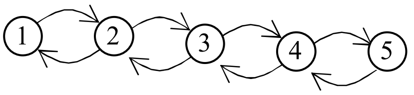
$$A = \begin{bmatrix} -.7 - G^{\wedge 3} & -1 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 5 & * G \end{bmatrix} \tag{3.13}$$

With U restricted to [0,1] and G the gear number.

The problem is to determine the best RchW and the optimal control, to drive the vehicle as fast as possible from Stop condition (initial state) to End conditions which are defined by a speed of 90Km/h and acceleration of .15m/s , which is the maximum speed to safely take the curve at 2.5 Km.

The fifth steps to solve this problem are the following:

1) The first step is to define the possible graph evolution:



2) Second step is to determine the transition conditions reachability.

The state transitions are triggered in order to maximize the acceleration, they are active when the next state presents at the same velocity, and higher acceleration.

1 to 2 condition transitions: when speed ≥28km/h and acceleration ≥ 1m/s

2 to 3 condition transitions: when speed ≥57km/h and acceleration ≥ .4m/s

3 to 4 condition transitions: when speed ≥82km/h and acceleration ≥ .24 m/s

4 to 5 condition transitions: when speed ≥115km/h and acceleration ≥ .1 m/s

The state transitions' reachability conditions are the following:

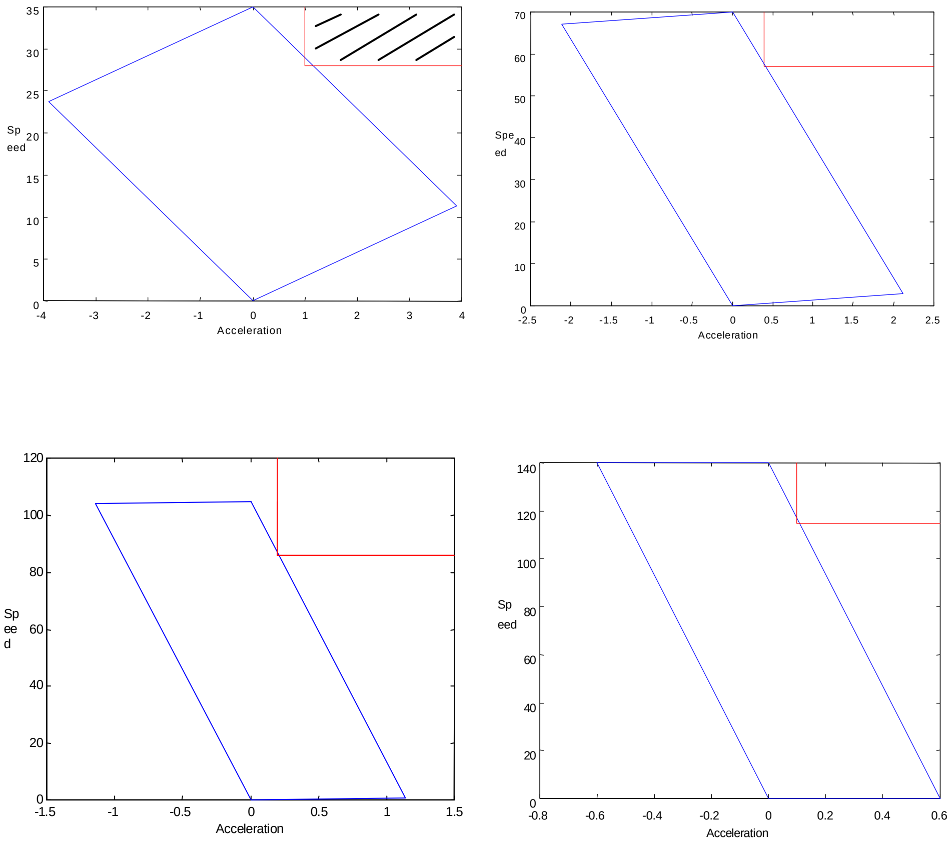


Figure 3.12: Transition conditions reachability for 1st, 2nd , 3rd and 4th gear

The graph shows that transitions, 1 to 2, 2 to 3, and 3 to 4, are reachable.

The reach set is given into these limits:

1: 3.9 m/s at 11.25 km/h, -3.9 m/s at 23.5 km/h, and maximum 35 km/s.
2: 2.12 m/s at 2.8 km/h, -2.12 m/s at 67.2 km/h, and maximum 70 km/h .
3: 1.14 m/s at .66 km/h and -1.14 m/s at 104 km/h and maximum 105 km/h.
4: .6 m/s at .18 km/h and -.6 m/s at 139 km/h and maximum 140 km/h.
5: .38 m/s at .12 km/h and -.38 m/s at 174 km/h and maximum 175 km/h.

The End condition's reachability , 90 km/h of speed and .15m/s of acceleration, can be seen in the figure 3.13.

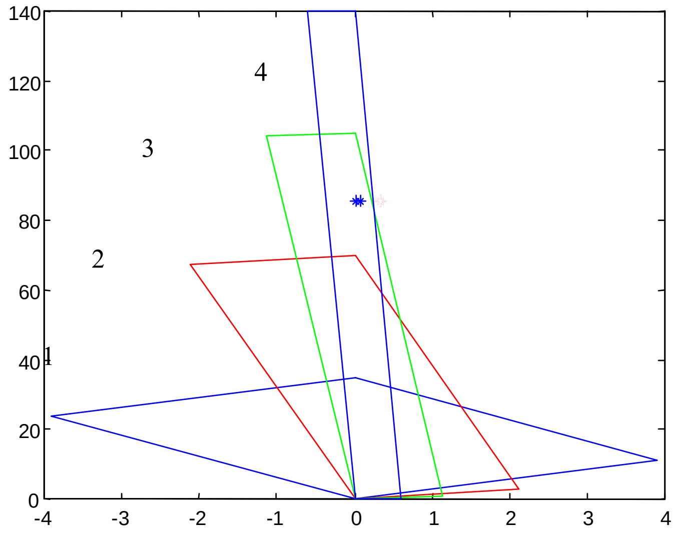


Figure 3.13 : End conditions reachability. Reach sets and End point.

In the case of multiple solution, the End point's reachability is obtained by evaluating the different solutions in order to determine the optimum. When the models have similar rising times, the optimal state will be the one having its reach set limits most distant to the End point. The Bang-Bang control forces to follow the extremes of the reach set, and distant limits express faster acceleration for the same velocity.

3) Apply the abstraction over continuous signals, to obtain the set of observable and controllable events.

The events generated by the continuous signals are reachables, this implies that they belong to the set of controllable events:

$\Sigma_c^{'}$ = {transition: 1-2, 2-3, 3-4, 4-5}

4) Step to obtain controllable languages.

TCT [Thistle 94] yields the Supremal Controllable Sublanguages to drive the system from state 1 to state X (2,3,4,or 5):

Once the reachability criteria of the states are met, the path through those states will determine the RchW.

Two possible RchW : 1 -> 2 -> 3 -> 4-> 5 -> 4 and 1 -> 2 -> 3 -> 4-> 5 -> 4 -> 3 .

The two possibilities must be analysed to determine the fastest possibility. In this case, without delays in the transitions of the states, the fastest one is 1 -> 2 -> 3 -> 4-> 5 -> 4 -> 3, because the 3th gear is faster decelerating than the 4th one.

5) Optimization method calculates the control signal U(t), which concludes that the minimum time is 85 sec.

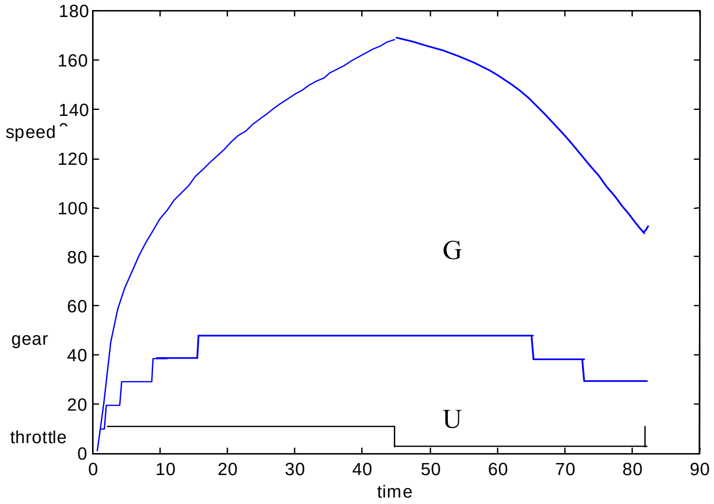


Figure 3.14: Continuous signal U(t) throttle, and discrete G gear.

Chapter 4

4.1 SUPERVISION

In industry often there are problems of losses measures or things go wrong. This can be called faults and this does not have to stops the process. The industry must detect these faults, and determine the instant and the faulty element to recover the system at the normal operation. This process can be named diagnosis and can be separated into the fault detection process and the fault isolation process.

The problem can be categorised into two classes, the anomaly when the tendency of the signals change abruptly and suddenly, and discrepancy when the value of the signal is very unstable. The first one is caused by a mechanism of the process, damage over the actuator or the process change dynamics due to the change of the process model. The second is the representation of the abnormal function, uncontrolled actuator, noise interfering, uncontrollable process.
For both cases it is necessary an accurately design, the firing alarms and inspect the system to obtain the possible cause. The time elapsed in this process is very important when the objective is to realign the system to the normal operation.

4.1.1 DIAGNOSIS APPROACHES

The problem to isolate the origin of the problem or alarm them in the manufactory industry is extensively treated. Traditional techniques or new artificial intelligence techniques applied over the system can solve a large number of these problems.

Statistical techniques are traditional methods. Artificial intelligence as fault trees, expert systems , or model in discrete event systems DES are methods which use the information of the qualitative model of the known system to determine the instant of the fault.
Other methods use a model based to diagnose the fault. Analytical redundancy methods analyse the residuals between the system and the quantitative model to measure this distance and discover the origin of this deviation.

Reasoning backwards over the qualitative or quantitative models is necessary to find the origin of the problem, but it is difficult to reason in a non complete model, the faulty behaviour must be expressed in the graph of possible behaviour.

In Alpha-model once the control part has reached, the safe , reachable and optimum path to drive the system from the Starting point to the End point, is the supervision part which must take care in higher level to assure the following proposed plan or path will be executed. Some unexpected problems must be recognised and a new plan proposed to realign the system in the expected sense.

Equally the control part works with continuous and discrete variables, the supervision part profits from this information to recognise the fault , the moment, and the causes to make a suitable reaction.

The structure of the control have two layers, the control and supervision layer (Figure 4.1) which is over and interacts with the control layer. The inputs are the continuous signals, and discrete events form the system and the events generated by the events generator, which produces events when are changed the local model.

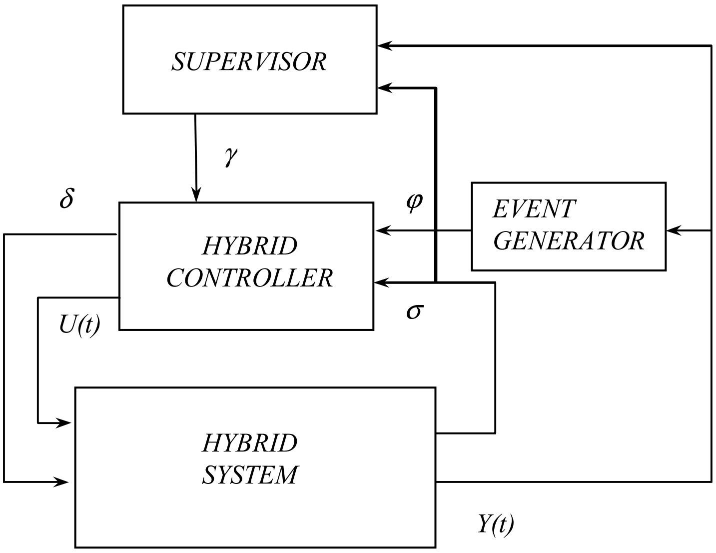


Figure 4.1 Structure of the supervised system.

In the supervisor is represented in a graph the possible evolutions of the real system. A reference model must be complete in order to recognise and situate any abnormal situation, this is the identifiability of the state. In some cases the graph cannot be complete, but parts of the graph mid-disconnected are present and give a possible solution to realign the system (Figure 4.2)

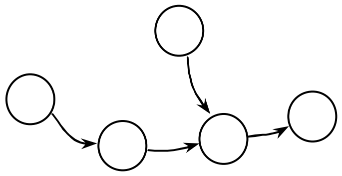


Figure 4.2 : Mid-disconnected sub-graphs.

The model is constructed by the decomposition of the Alpha-graph and the relations between them. Experimental cases, and the foresight cases can be adjoined in order to collect the majority of the possible problems.

The result taken from the supervisor block is to assign current state and assign events to realign the system toward the control block, in this case it pays attention to these signals and are inputs to the control table in order to obtain the actions to be applied over the system.

4.2 FAULT DETECTION

A fault is said to be detectable in a discrete sense, if there exists a transition in the system model that leads to detection in a finite number of steps. [Sampath Sengupta Lafortune 96] [Sampath 95]

A fault is detectable in continuous sense if the identification method determines a fault state. The analysis of the residuals identifies a fault state [Isermann 91][Frank 90][Patton 92] Qualitative models , state estimator observers and decision making are approaches to fault detection and isolation FDI.

Unobservable events may be fault events or other events, that cause changes in the system state not recorded by sensors.

The combination of the continuous and discrete methods is useful to use in Hybrid Systems for fault detection in the supervisor, the procedure is the following:

1) To detect that the continuous signal does not progress towards the transition condition.

Evolution of the continuous variables is not in right direction, this is the limit of the next transition, (figure 4.3) The response of the continuous variable is not the expected. This is possible if the continuous model is changed.

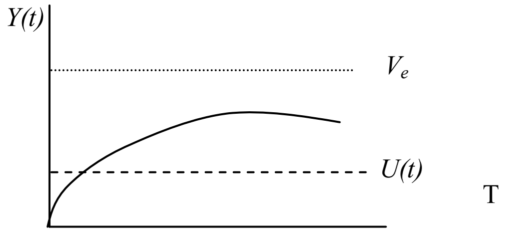


Figure 4.3: Continuous evolution

2) Analysis of the continuous signals: model identification.

Identifying the current model can inform of the current model to use and determine in which adjacent state the system is. In this case a non-observable event is done in the transition.

3) Analysis of the discrete events presented to indicates the fault into the system.

Some times can arrive a discrete event to mark the new state, a fault state. For example: too much time elapsed or no liquid in the pump, leakage, to exceed level, etc.

4) Representation of the state in the reachable sets.

System can inform the change of state in the graph of the reachable set, indicating the current state.(figure 4.4)

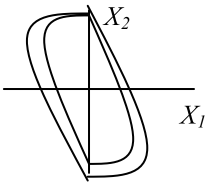


Figure 4.4: Superposed reachable States

These four steps allow detection of a wide range of the possible faults in Hybrid systems. The analysis sequence is the above, but sometimes the discrete event to indicate change of the state can arrive at any time. The combination of the continuous and discrete signals informs about the majority of the problems.

In the automatic gear example the use of the reachable sets are interesting when the fault cannot be detected by inspection of the continuous signals, for example when the wind perturbation does not allow to identify the exact model. When the state is out of the current reachable set (figure 4.5). This method determines that the current state (gear) is higher than the expected state, when the state crosses the limits of the reach set for the side of the velocity, and otherwise in the side of acceleration, the current state is a lower gear.

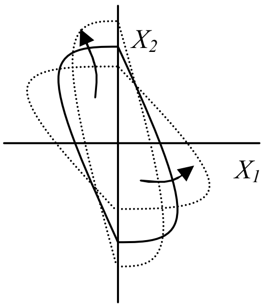


Figure 4.5: Observable faults

4.3 FAULT ISOLATION

Fault is isolable in discrete sense if there exists a transition, which gives different behaviour or next states for different faults. [Sampath 95]

A fault is isolable in continuous sense if the identification method determines a unique fault state [Gertler 91] Structured residuals, which are individually sensitive to different faults, or fixed direction residuals to the specific fault.

Model for diagnosis requires complete model behaviour, joining normal operation and faulty system behaviour and the observations of the real system. This allows to determine the In Fault state, from the specific normal Out state. (figure 4.6)

Some authors mark the premises to design a diagnosable system. [Sampath 95][Larson 97] Explaining the need to remember into the model the precedent states for the current system state, because the fault can be detected after some evolved states, the output can reflect the problem after some states of the fault state. And to design the requirements to model a system with necessary sensors to register the expressive events, which help to identify the possible Fault State. It is done with a complete behaviour graph, this can give a state explosion problem. Comparing the observed graph of the system and complete model must introduce necessary sensors to recognise different faulty situations.

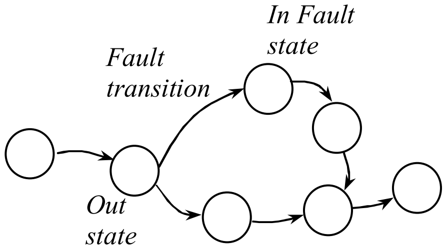


Figure 4.6: Model for diagnosis.

In Alpha-model the transition tables allows to predict the system with forward and backward analysis. The recorded events store the latest states. The objective is to find a non-deterministic transition, which gives a state with possible problems, once it is found we need to change the analysis direction (figure 4.7), the faulty direction, and to apply the latest event to the model, to recover and explain us the current state.

When the reconstructed state corresponds with the current continuous behaviour of the system, the backward research stops the research in the past states. Otherwise the possible deviation is not correct and is needed to search the bifurcation in previous states. (Figure 4.7) This observable fault state allows to the supervisory sytem to explain the fault. And at the same time can explain the previous behaviour of the system, and allow us to confirm the supposed model state is the current state.

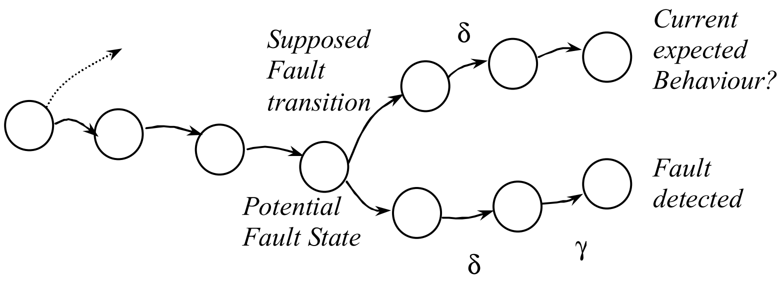


Figure 4.7: Recovery state

The use of the continuous and discrete signals, augments the possibility to recognise the abnormal situation. The analysis of the continuous signals can confirm the current model used after the backward analysis, otherwise in only discrete events is necessary a sequence of events to confirm the same. With continuous signals this process is shorter and can realign the system some times before is too late.

4.4 EXAMPLE

Suppose three tanks to produce a mixed compound, the mixing of the two components is done at specific temperature, each one at a particular temperature. The process can be realised with this schema.

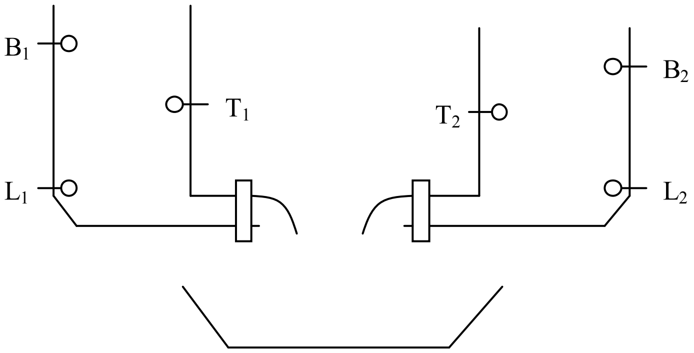


Figure 4.8 Schema of the process

The upper two tanks are filled with an individual pump and heated by the electric resistance. B_1 L_1 T_1 are the discrete signals, which indicate the levels, and the temperature reached. Due to the delay to heat the liquid both tanks are heated at same time. In order to avoid the non-determinism for the ending time, the second tank are filled with a few delay of the first tank.

The expected sequences of the process are:

L_1 L_2 B_1 B_2 T_1 T_2 L_2 L_1

Arrival of the following sequence, as an abnormal operation:

L_1 L_2 B_2 T_2

The process cannot evolve, the automata does not recognise this situation. At this instant the inference algorithm in the supervisor tries to understand this situation.

Two possible situations can be supposed, the lack of the sensor B_1 or not enough liquid is placed in the tank. A simple solution to analyse the second case, the action is restart the pump when the problem is the air in the pump or pump feeding, in some cases the pump can restart solving the problem.

The event that will arrive in this case is:

B_1

At this instant the cause is determined, the possible fault state is supposed, we need to evaluate the sequence of the presented events to determine the exact state where the process is:

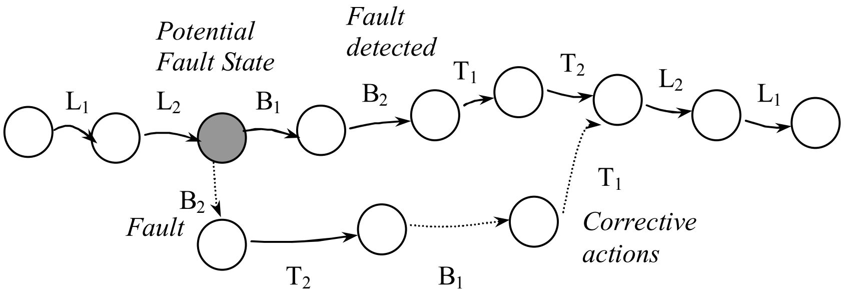


Figure 4.9 Diagram of the fault evolution

Once the current state is determined the supervisor calculates the necessary actions to drive the system to the normal operation. In this case it is necessary to heat the liquid to converge to the normal operation in the empty state.

At the end the events sequence is the following:

Faulty case:
$L_1\ L_2\ B_2\ T_2\ B_1\ T_1\ L_2\ L_1$

Normal operation where:
$L_1\ L_2\ B_1\ B_2\ T_1\ T_2\ L_2\ L_1$

Some process in fault situation needs more events than under normal operation, discover the problem and realign the system, can need more states and actions. This is a simple example to understand the problem of the identifiability of the faulty state, to recover the current state and the necessary actions to realign the system to the normal operation.

Chapter 5

5.1 APPLICATION TO THE ROBOT SOCCER

The robot dynamics is a system definitively hybrid. The movement is ordered into sections and these interact. The continuous and discrete components are present in each movement and also interact. The planning is a huge task of searching the adequate sequence of subtasks to be implemented.

For example a robot arm takes an object from the box and fits it in a machine. This work is separated into subtasks as : the approximation to the object, grasp them, change the position and fixing the object into the machine. In each step the stepper motors are controlled, these are interpreted as continuous velocity control, and discrete signals marking the starting and finishing the subtasks. And more other elements of the robot are intrinsically discrete, the grasp, the solderer etc.

How to plan, control and supervise the robot is an interesting domain for the Hybrid systems domain. The robots normally are programmed to realise a proposed task, sequencing the subtasks, a independent modus, without taking into account the interactions between the phases neither minimising some global cost. To do this requires a complete knowledge of the system, models ,relations etc. and a complex control system is needed to profit from the advantages of the Hybrid systems control techniques.

We apply Hybrid systems algorithm in a particular domain of the robot control, the robot soccer [Esteva et al.98]. The robot soccer is a platform to try and compare algorithms, from control ones to the artificial intelligence for planning. It is a sufficient stable platform which allows international competitions between research groups, which evaluates and compares these news algorithms. The competitions help to improve the new techniques and algorithms for all the subparts.

5.1.1 ROBOT DESCRIPTION

The description of the autonomous robot we are using to play soccer is represented in the following figure 5.1:

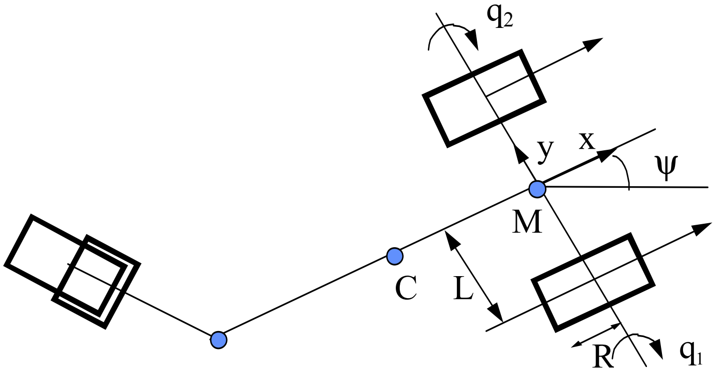


Figure 5.1: Mobile robot co-ordinates

Structural properties of kinematic and dynamic models taking into account the physical mobility restrictions of the robot, give a classification of the mobility and steerability, measured in the freedom of degree for them. In this case with two fixed drive wheels, and one or two points with a castor wheel, results a kind of robot (2,0), meaning 2 degrees of mobility and 0 of steerability.[Campion Bastin Andrea-Novel 96] As a result a non-holonomic autonomous robot, with the kinematic model is reduced to:

$$\dot{P} = \begin{bmatrix} \cos(\psi) & 0 \\ \sin(\psi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \dot{\psi} \end{bmatrix} = B(\psi)U \qquad (5.1)$$

With P as vector position of components X, Y θ, and V, ψ, linear and angular velocity. Transformation to $B*U$ to express in U the control action applied to each wheel.

The system is highly non-linear, and is controlled by two signals, the right and the left drive force into the wheels. With these two controlled signals we need to drive the robot from one position to another and so on. The hybrid nature is shown because these subtasks interact. The velocity reaching the End position affects the new positioning task. Discrete and continuous interactions appear with the continuous control driving versus discrete mark positions, ball interception, etc. Different continuous dynamics are also present, from to drive in one sense to the reverse, to accompany the ball or pushing other robot. All these are models and conditions that apply to Hybrid system Control.

Two possible methods can be applied in order to control this non-linear system: A non-linear controller able to control the system in all ranges, or a tracking controller, if it is possible to define the velocities profile for the wheels, only for the optimal computed trajectories. Otherwise and for facility we apply a non-linear controller largely treated in the literature [Zhang Mackworth 95]. The following controller for positioning is the outer loop of the piecewise linear velocity controller in the inner loop for the motors.

$$V_{lin} = V_{ref} + k1 * \sqrt{x\_error^2 + y\_error^2}$$

$$V_{ang} = \sqrt{x\_error^2 + y\_error^2} * k2 * error\_\psi_{traj} + (1 - \sqrt{x\_error^2 + y\_error^2}) * k3 * error\_\psi_f$$

(5.2)

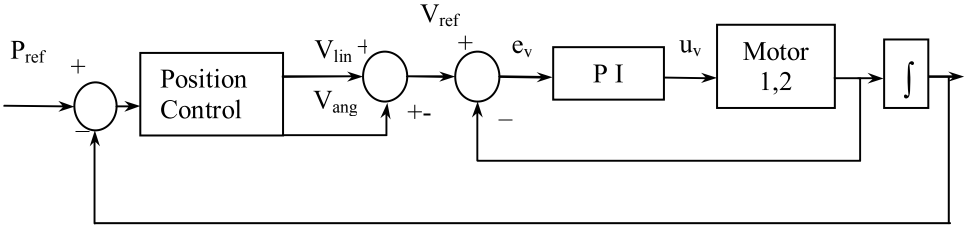


Figure 5.2: Controlled System

The robot path planning is a great problem and even more over a cost function and dynamic scenario. Dynamic scenario means the system is in a changing environment in time, this means the goal to reach at this instant can be impossible to reach at an other moment, physical restrictions can avoid this objective later. This restriction forces it to reach the goal at a particular time, the classical control goal is to stabilise the system at certain set-point without time restrictions, that are not applicable in this dynamic scenario.

The cost function to apply in this dynamic scenario is the minimum time problem, due to the necessity to assure the system to reach ball condition before an other player reaches. The exact time to reach the goal is decomposed in two subparts: The minimum time to reach the goal, as a meaning of the maximum capability of the dynamics of the system. And the second part the rest of the time, can be implemented by extra time, do nothing before the movement or attending the shock at the end of the movement.

The problem is greater when in this scenario collaborates more than one robot. A task performed by two robots is more powerful and a problem more intractable. We restrict to the passing ball problem, where the goal is the but or to overpass the Goal Keeper conditions. The playing possibilities with two players augments largely the capability to overpass the Goal Keeper conditions.

Physical limits in the game, restricts the capacity to reach the goal, dynamic models, physical parameters, and models for all the participants in the game. The goals and the control conditions, as limits, functions, etc. make ready the system to be expressed in mathematical formulas. For the solution of the equations can resolve the actions to be taken to reach the objective. The complexity of the problem can give several solutions, the methodology applied to find the optimal is, the Backwards analysis and reaching intermediate goals until the Starting point is reached.

5.2 ANALYSIS

5.2.1 PHYSICAL ANALYSIS

The analysis task uses the models and the verification procedures to obtain the possible solution to realise desired task. If some of them are able, is measured the satisfaction degree to reach the goal and the best will be applied over the running system. The analysis is applied in two phases, a planning over the situation into the running game, proposes a tactics or sequence of actions to reach the goal, and the second phase is to apply over the robot the sequence of control actions to execute the proposed movement by the planner.

For the analysis by the planner is used the backward analysis over the model scenario for the game, and for the robot control actions the optimisation algorithms over the robot model is applied.

In the planner first is analysed, if it is possible to reach the goal with only one player?, otherwise is analysed the conditions to solve this problem with two players, and also if, it is possible with the current starting conditions, positions, velocities, and time?.

The methodology to solve this problem is the following procedure:

Backwards analysis take the attention to the End state, when the Goal Keeper is overcome. In this state the goal is to avoid the interruption of the ball trajectory for the Goal Keeper. Graphically it is expressed by this distance in a graph, which is represented by the attenuated ball velocity in time with exponential decay expression, and the maximum energy applied to the Goal Keeper to attain the ball.

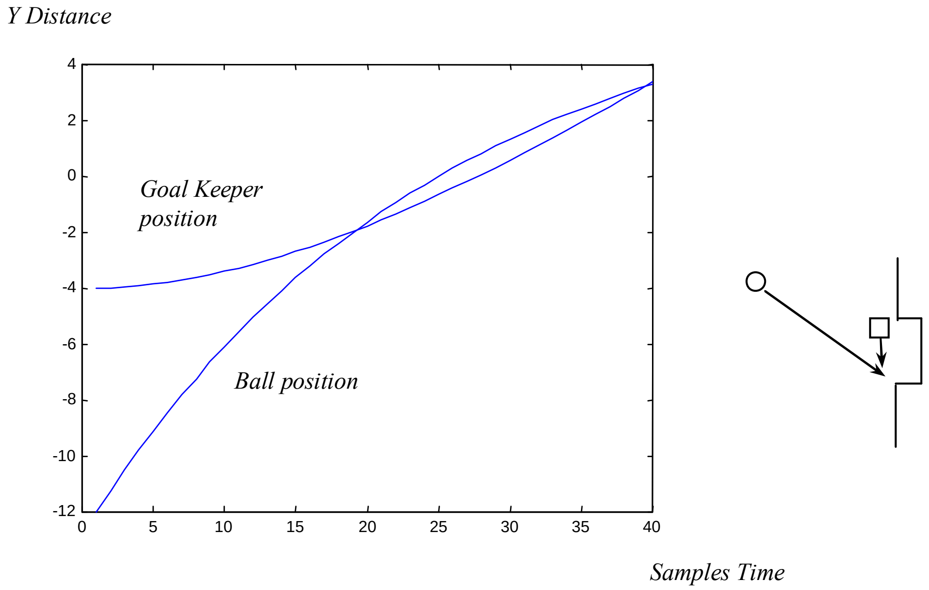


Figure 5.3 : Goal Keeper reachability of the ball

The non-intersection between these curves means the advanced position for the ball in front of the Goal Keeper and the free ball trajectory in the field. This range is the margin to reach the goal, which gives the Y margin for the goal (physical distance) . In this situation the X distance must reach the X=0 position, origin of the goal.

The margin interval is confined taking into account all the requirements: To overpass the Goal Keeper, and to reach the Y and X distance of the goal. Once this requirement is reached we propagate this interval to the previous state.

The previous state is the shock between player and the ball. From the current position of the ball with it's velocity and trajectory, we need to find a valid region which has possibilities of the inelastic shock and give this trajectory to the ball towards the goal, into the previously analysed interval.

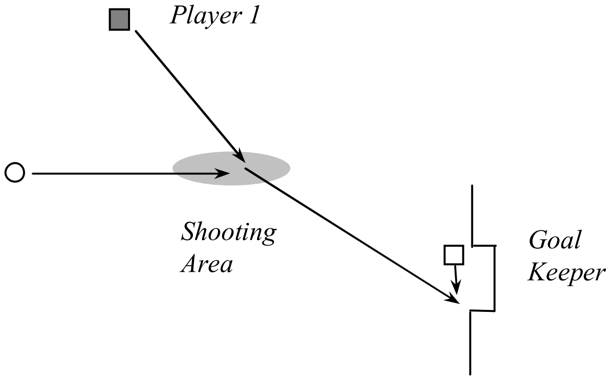


Figure 5.4 : Evolution of the ball trajectory.

At the end the request for the player is to reach this region with some requirements. A specific velocity and acceleration give to the ball the adequate velocity to overpass the Goal Keeper. If the distance between the player and the shock point is enough, the player can reach the stable velocity, and the reachable condition is reduced to the intersection of the shock conditions with the reach set of the player.

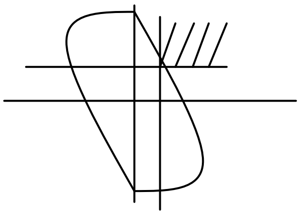


Figure 5.5: Reachability of the Shock condition.

When this condition is not reachable, this means the requirements of the velocity are not possible for this robot in particular, this requires the analysis by other robots and finally by other ways to reach the goal. Other reachable way can be the passing between two players. Other situations are analysed, taking into account the estimated position of the Goal Keeper following always the ball position. The new situation to analyse can be the following:

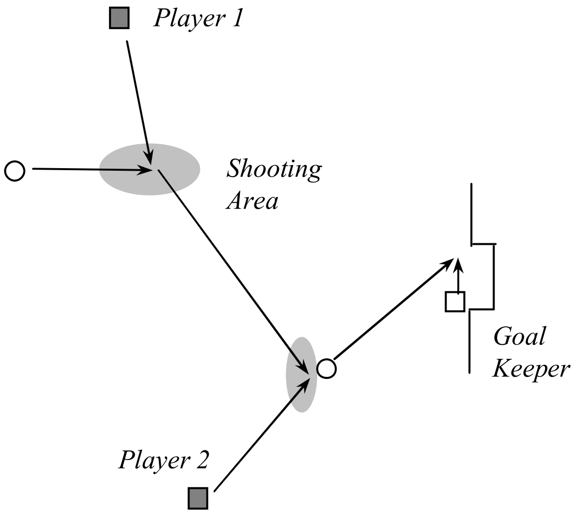


Figure 5.5: Evolution of the passing ball trajectory.

These intervals are propagated, and the time is enlarged, this makes possible more capacity to solve the problem. The problem is reduced to propagate intervals and to check if these intervals are reachable in the reach sets of the two robots.

It is not interesting to analyse situations at more than two or three states, the non-determinism of the game, the movement of the opposed players force us to avoid large predictions. Some predicted ball trajectory can be interrupted by an opposed player, meanwhile the opposed Goal Keeper behaviour is predictable.

5.2.2 IMPLEMENTATION

The implementation into the soccer team requires real time restrictions and safety conditions. These are solved restricting the searching solution over the advanced player, and reducing the viable time interval conditions by a security margin.

The analysis sequence for the states to reaching the solution is:

1- Is it possible to give the ball the V_y velocity greater than the opposed Goal Keeper at the goal position?.

This is analysed by the expression of the ball velocity by distance plus a margin.

`abs(vy) > 1.3+abs(.055*(70-yo))` (5.3)

This obtains the minimum velocity to overpass the 1.3 as the identified maximum Goal Keeper velocity. Next this limit is transformed to the previous semi-elastic shock.

`Vyj = vy*4.55-Vyp-.35` (5.4)

Resulting the necessary player velocity in the shock instant. The reach set velocity of the player solves this condition.

`Vyj < vymax` (5.5)

2- It is possible to passing the ball to the partner and to perform condition 1 ?

To analyse the passing conditions in the minimum time condition, it forces to passes the ball next to the goal in a safe distance for the Goal Keeper. Two regions are defined as possible areas for the passing ball. In this regions starts the analysis for the previous situation.

Otherwise if it decides there is no solution at this instant, the strategy is to approach the ball to the goal.

The following simulator shows the above behaviour, when the first condition of shooting the ball to the goal is not possible, the simulator shows the passing ball to the second player. The second advanced player shoot the ball against the goal when to is possible to reach the final condition, otherwise approach the ball to the goal and pass the ball to the player one.

In the figure 5.6 is shown the complete trajectory for the ball, from the starting game to the but, at reaching the goal, and the final position of the players. The passing play is used to reach the goal with the velocity and ball position conditions.

The simulator gives possible results in planning the strategy. This strategy must be defined in the control signals for the robot movement. In the next part of the chapter is converted this situation in the exact values for the robot trajectory. The restrictions given by the simulator are the initial and end position and a maximum time to execute the movement, and the mathematics obtain a minimum time and error to accomplish this specifications. If the time obtained is less than the necessary, is extra time, the movement can start in the last time possibility in order to arrive at end situation with required parameters, velocity and direction.

A controller for a following path can be placed into the robot to follow with accuracy the proposed trajectory. In this case is only necessary to planning once before the execution, also due the calculus effort which impossibilities the on-line calculation or adapting, during the movement.

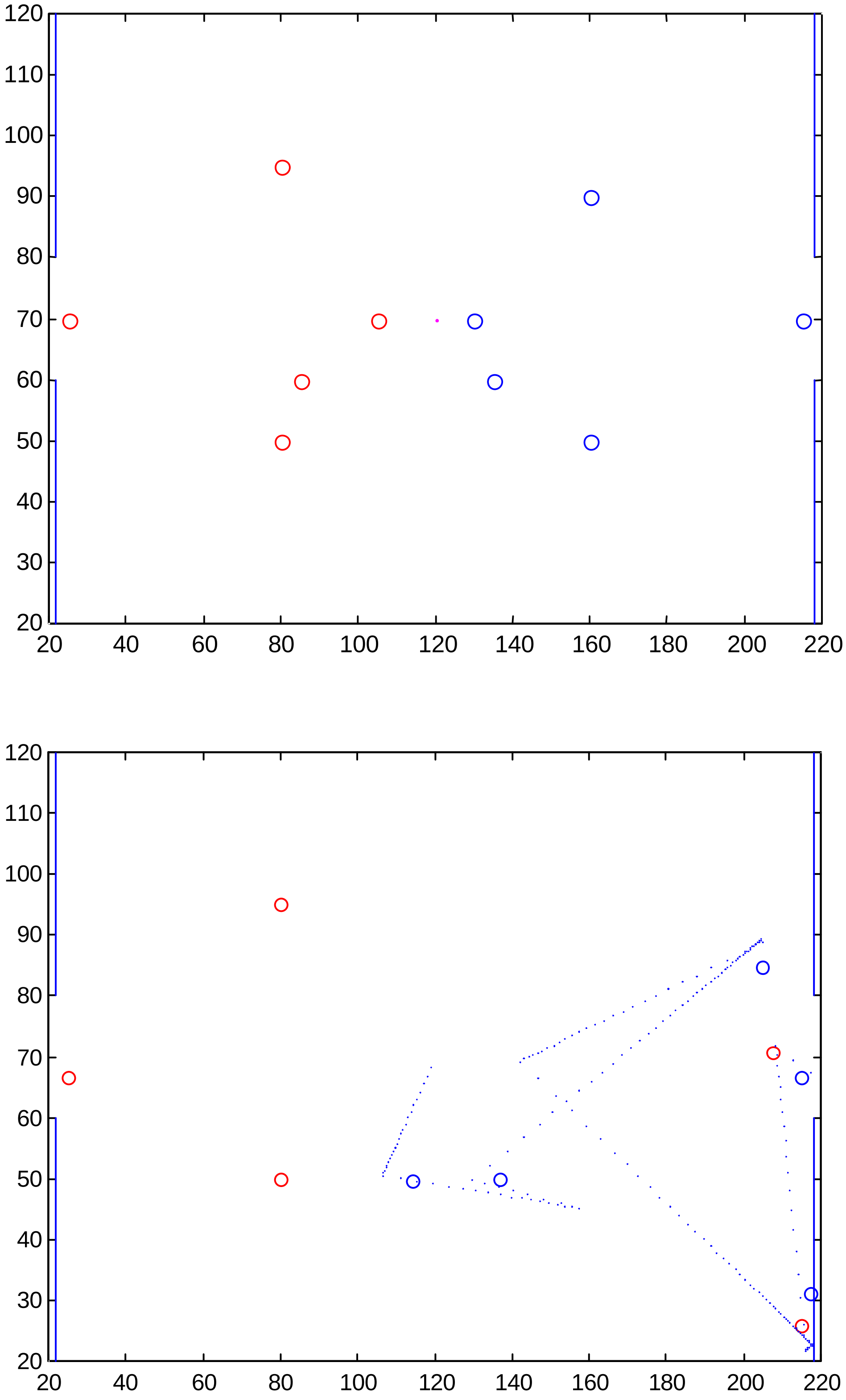


Figure 5.6 : The ball trajectory in a passing shoot.

5.3 OPTIMAL TRAJECTORIES

5.3.1 SIMULATION

The optimal trajectories for non-holonomic robot are treated by the research community [Soueres Laumond 96] and is a problem no yet solved. We obtains a near optimal solution using Dynamic Programming optimisation DP using as parameters the applied signal to control the wheels for each discretised time over the alpha model. It is a great computational effort and it is a NP Hard problem, for more than hundred time instants the problem are intractable.

The problem can be solved by different solution form: Calculus of variations over the non-linear model, or applying Dynamic Programming over the linear piecewise model. Both methods can give approximated results. Big problem is the non-global minimum solutions that can appear for non-linear models. Exploration from different initial values for the piecewise linear minimisation can assure the optimal global minimal solution is found.

A specific function modifies the non-linearity of the robot model. Multiplying the model by the $\frac{1}{(s \quad \theta)}$ give a smooth non-linear function. Anyway the approximation by piecewise linear local models is more tractable if it is only used three or four local models from the non-linear model [Esteva et al.98] and applied over the following trajectories:

Starting the robot from the origin, without velocity, and the objective position of .5 meters of X distance and Y distance, and final velocity minus one meter per second for X and Y, is a difficult trajectory to do.

The near optimal solution for the non-linear model *f(x,t)* is the solution to this minimisation function:

$$\min_{k+l+m} | x_f - f(A_1^m(A_1^1(A_1^k x_0 + \sum_{i=1}^{k}(A_1^i B_1 u_i) + B_1 u_k) + \sum_{i=1}^{l}(A_2^{k+i} B_2 u_{k+i}) + B_2 u_{k+l})$$

$$+ \sum_{i=1}^{m}(A_3^{k+l+i} B_3 u_{k+l+i}) + B_3 u_{k+l+m}) \leq \varepsilon \qquad (5.6)$$

Where the control values are the objective for the Quadratic Programming function QP, and the i,j,k values are readjusted by simulation to confine the models into their limits: α and Y_{max}
This case is the combination of three local model A_1 is the model in starting position, when the friction and inertia has the predominant effects in the dynamics. A_2 the second part is the stable dynamics at high speed. And the third part is the model at opposite direction and when the motor can recover the inertial energy to the battery.

5.3.2 RESULTS

The two optimal solution for the proposed initial and final conditions are represented in the following graphs:

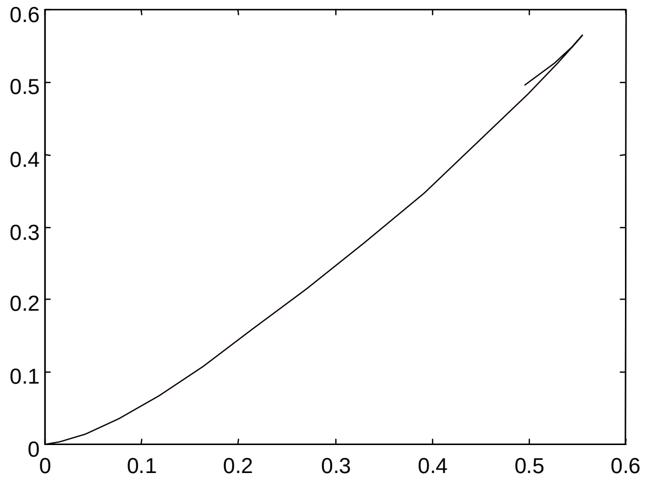


Figure 5.7: Position in 1.3 seconds of time

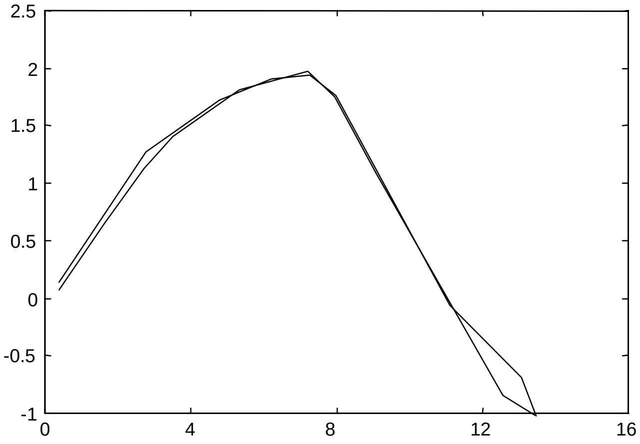


Figure 5.8 : Velocity versus time intervals .1 seconds each.

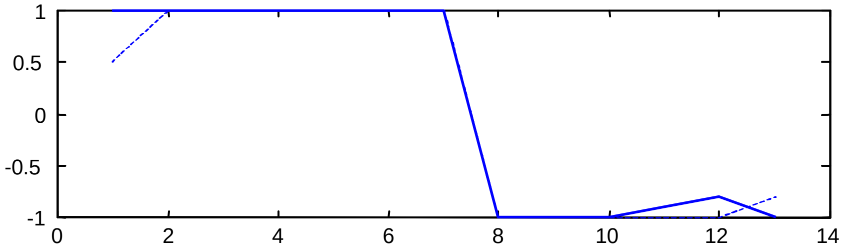
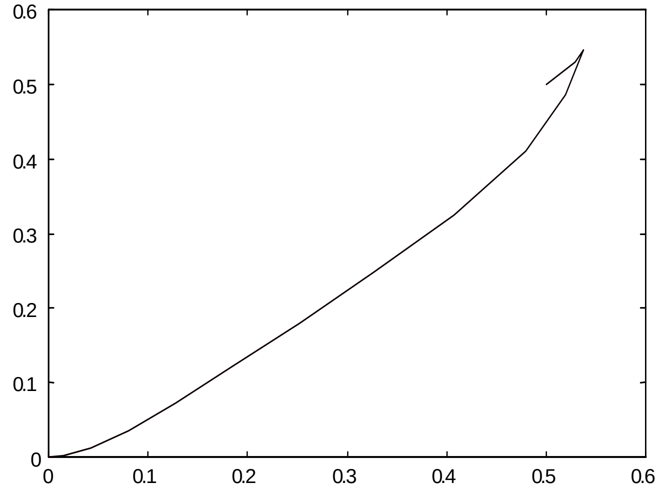


Figure 5.9 : Control signal.



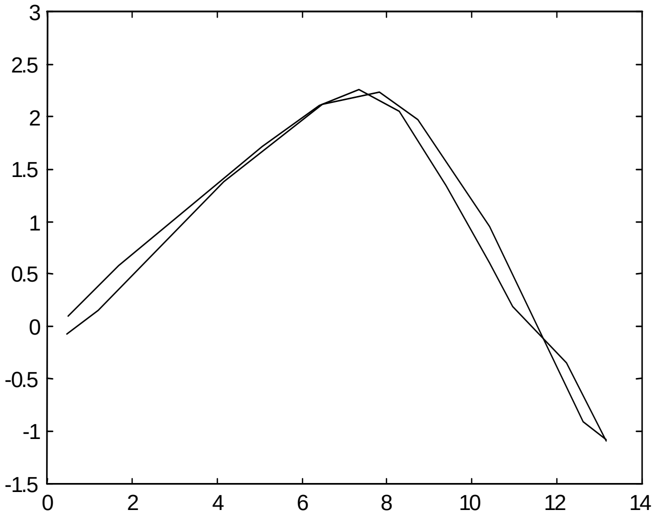
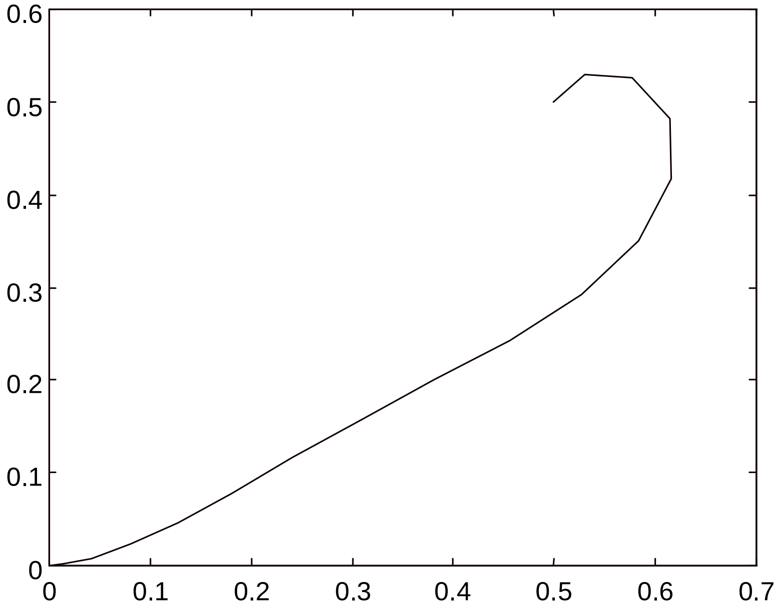


Figure 5.10: Optimal trajectories of the approximated linealised model.

Sub-optimal trajectories obtained for the approximated model. Smoothed trajectories are also able to reach the final conditions. The consumed time to cover the trajectory is 15 and 16 units of time of .1 seconds each, larger than the optimal with hard movements.



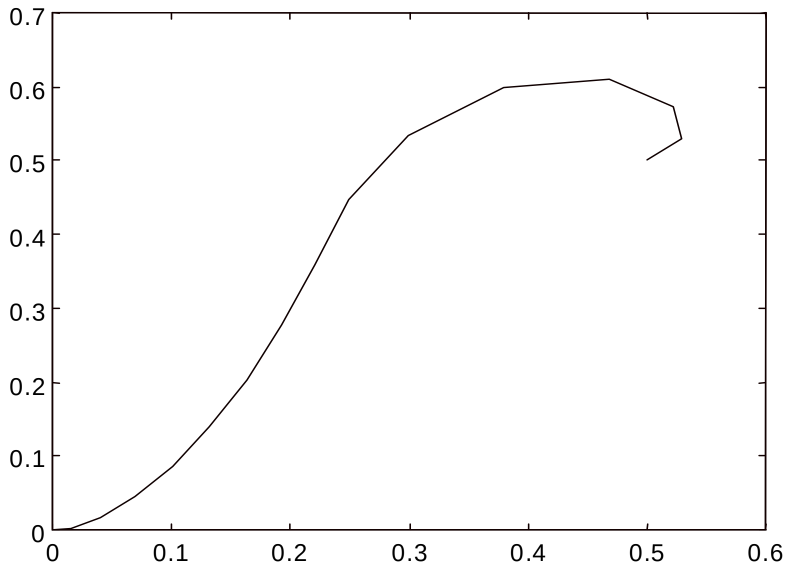


Figure 5.10 : Sub-optimal trajectories for the approximated non-linear system.

This results express, firstly the existing more than one solution, this is result from the non-linear function for the position of the robot, is a combination of *cos(ϑ)* and *sin(ϑ)* .

Secondly is showed the trajectory is combined by the strictly optimal movements, as defined in the study for minimal time trajectories for a non-holonomic robots [Soueres Laumond 96] , defines the optimal trajectory is combined by basic movements, in this case spiral, straight lines, and non-holonomic movement.

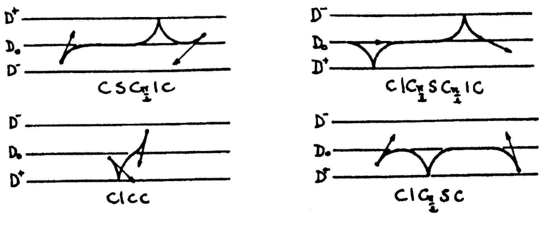


Figure 2.3: Exemples de trajectoires de type T_1

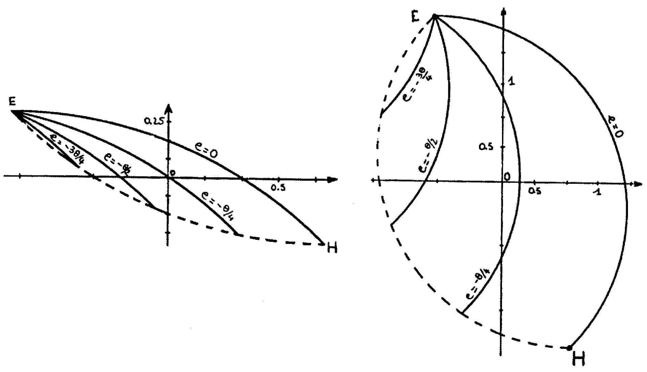


Figure 3.2: Domaine $l_a^+ l_b^- l_e^+$ ($\theta = -\frac{\pi}{4}$ à gauche) et ($\theta = -\frac{3\pi}{4}$ à droite)

Figure 5.11 Optimal trajectories for the non-linear model [Soueres Laumond 96]

5.4 COMPLETE ALPHA-GRAPH EXAMPLE

5.4.1 PROBLEM DEFINITION

A robot application pushing an object is a case where the complete alpha model is represented. It deals with different model, a non-linear model for one part of the process, and following an event, an other non-linear model is used. To optimise this process in order to minimise the time to realise the proposed task is the objective to model using the alpha model decomposition.

The process is defined as follows: The robot starts from one position distant from the object, and the objective is to pushing the object until it is inside the opposite goal. We are measurements of the system, the visual system give us the global position of the participant objects, and are known the both models of the robot, with free movement and pushing the object. To minimise the time to realise this movement, finding the trajectory and the actions to apply to the motors are the results of the optimisation procedure over the alpha-model representation.

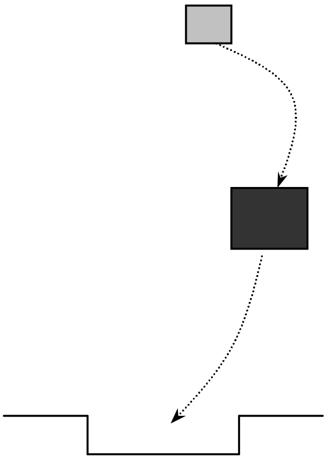


Figure 5.12: Objective trajectory to optimise.

The optimisation procedure must take account over the entire trajectory, this means to minimise the global path, from the Starting point to the end. This is necessary due the interaction between the two phases: free movement and pushing the object, the acceleration in the contact point that make influence in the second phase, the velocity in this instant must be lower than .25 m/s and non-positive acceleration to avoid rebooting effect. The position of the contact is fixed and can be predetermined in the mass center of the object. In the limits of the valid model only one parameter is predetermined the alpha or Y output measure, and the other is free and can be adapted to optimising the solution.

The joint set of the models to minimise over the cost function, give us a expected result and predicts the used models from the Starting point to the end. If the trajectory solution does not overcome the corners of the valid regions, the solution is correct, otherwise the joint set must to collect the collateral models.

The non-linear model DS_1 of the robot is the same model for the previous example:
M_1 : The starting and high acceleration model : $30/ s^2+4 s + 10$ and α limit is $\alpha >30^\circ \vee \alpha >-30^\circ$
M_2 : The stabilization model at velocity 2m/s.: $20/ s^2+5.5 s + 10$ and α limit is $\alpha >-30^\circ \wedge \alpha <30^\circ$

The model DS_2 for the robot pushing the object is (half velocity) :

M_3 : The starting and acceleration model : $16/ s^2+4 s + 10$ and α limit is $\alpha >30^\circ \vee \alpha >-30^\circ$
M_4 : The stabilization model at velocity 1m/s.: $10/ s^2+5.5 s + 10$ and α limit is $\alpha >-30^\circ \wedge \alpha <30^\circ$

The coordinates relative to the robot are :
The origin of the robot is 0,0. The position of the object is 30, 10 cm.
The size of the object is 5x5. The goal is a 60, -5, and large is 30.

5.4.2 ALPHA MODEL

The resulting model is represented by two alpha graph, with limits and models.

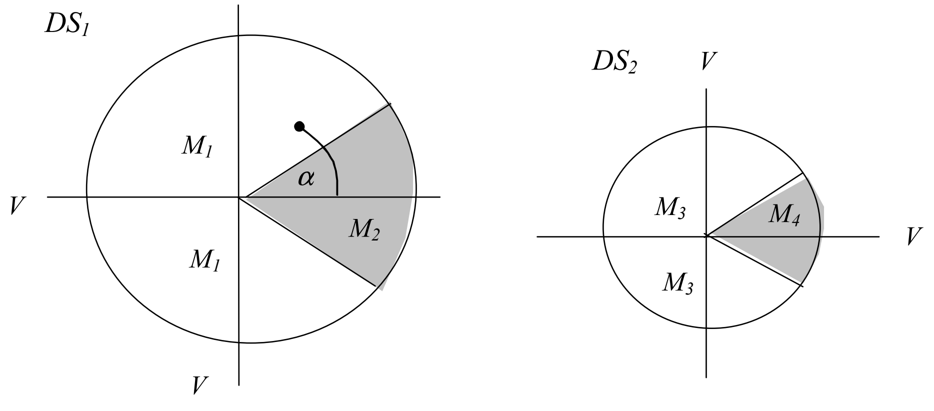


Figure 4.13 Alpha-graph representation

The minimising function is:

$$mi\,n_{k+l}\,|\,x_f - f(A_2^l(A_1^k x_0 + \sum_{i=1}^{k}(A_1^i B_1 u_i) + B_1 u_k) + \sum_{i=1}^{l}(A_2^i B_2 u_i) + B_2 u_l) \le \varepsilon$$

for the first part, free moving without object, with f the non-linear expression of the non-holonomic robot kinematics , and for the other expression for the second part:

$$mi\,n_{k+l}\,|\,x_f - f(A_4^l(A_3^k x_0 + \sum_{i=1}^{k}(A_3^i B_3 u_i) + B_3 u_k) + \sum_{i=1}^{l}(A_4^i B_4 u_i) + B_4 u_l) \le \varepsilon$$

assuming the conditions: The first acceptable error is for the position 27.5 cm. y 10 cm. x.
The second formulae the acceptable error for position 60,8. With maximal square error 1cm.

5.4.3 RESULTS

The resulting trajectory to be executed is:

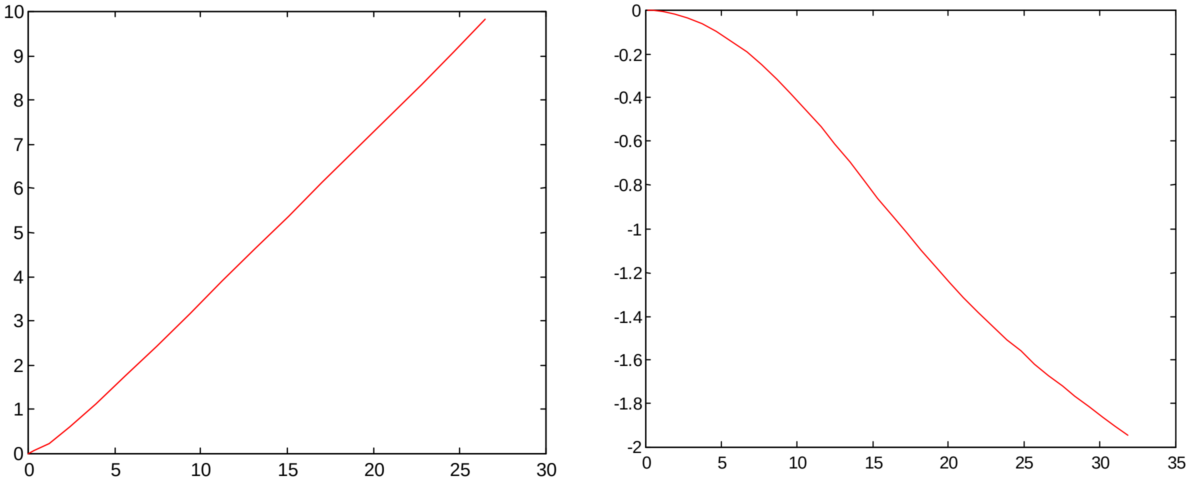


Figure 4.14 A) First part of the trajectory with free movements B) Accompaning object

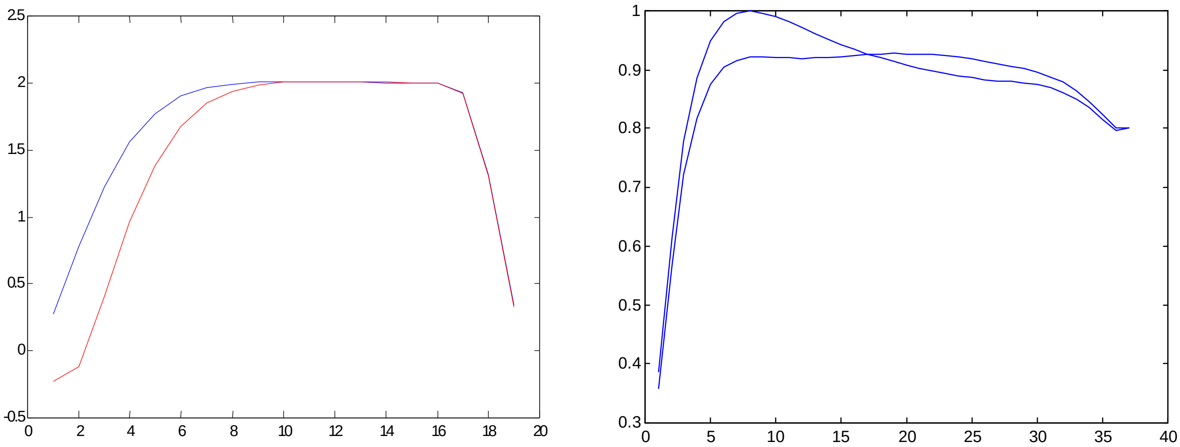


Figure 4.15 Motor velocities profile: A trajectory, Ts=.05 seg. B trajectory

And the representation over the alpha graph is:

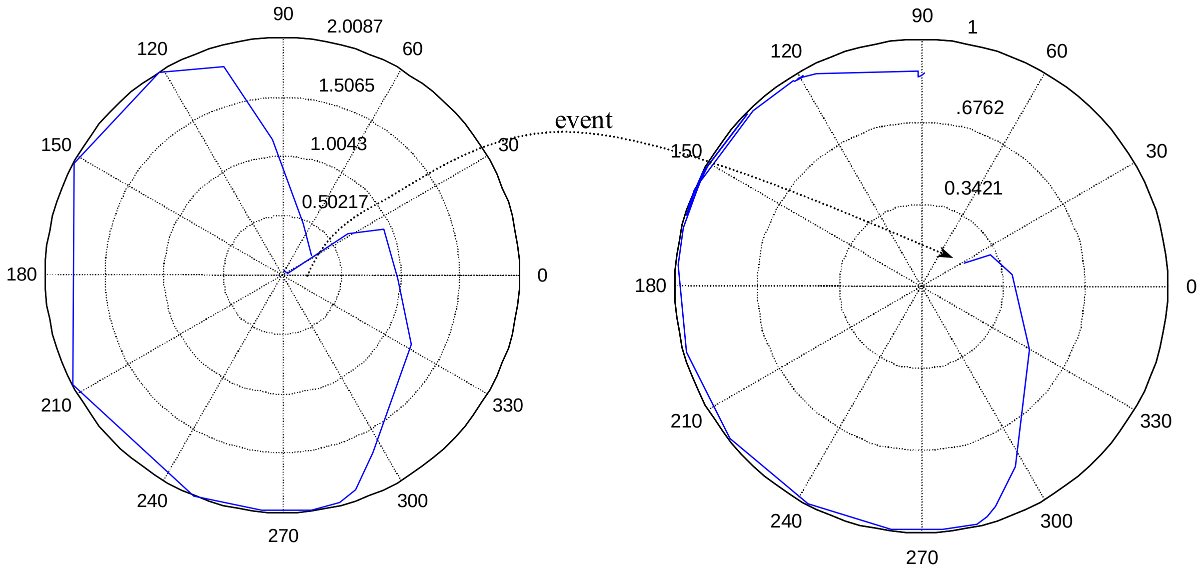


Figure 4.16 Alpha graph representation at event presented for one DC motor.

The complete trajectory is realised by the two parts, the approaching to the object, with a displacement of 27.5 , and 10 cm, to joint with the object in the axis of the gravity center. This trajectory is realised in 19 Time samples at 0.05 seg. a total of .95 seg using the two linear models. The second part of the trajectory is a displacement of 32.5 cm, and – 2 cm, to reach the opposite goal. The time needed is 37 Ts, a 1.85 seg. , that gives a global time of 2.8 seg, time to accelerate the robot, de-accelerate for the approximation to the contact, a new acceleration , and de-acceleration for the entry to the goal.

Only non-hard movements can be computed by this methodology, due the possible non-optimal solutions given by the f function, a non-linear function depending of *sinuns* and *cosinus* of the orientation angle of the robot. In this case a further analysis can conclude if these are the optimal solution, due the used algorithm to find the solution is a gradient form, and the validity can be studied by the exploration from other initial values to search the solution.

Chapter 6

6.1 CONCLUSIONS

An important area of automatic control theory is the theory of Hybrid systems. Typically it can use both process automata and regulators for the control of production processes. When the continuous and discrete parts are not optimised, the controller structure, and communications are not the most adequate. Yet, new knowledge may be gained by the users every time the control technology is applied.

Nowadays, these theories are well applied in simple Hybrid systems such as tanks, chemical reactions, etc. Their application over a large systems, as car , highways, or distillation column, are now under development and research . It is needed to further distinguish the purely continuous and discrete parts and their interconnections to make it useful for treating large systems. Thus, Hybrid systems theories will be able to optimise the system's performance.

It is a huge, and sometimes impossible task to find a mathematical expression for the behaviour of all complete system. For the success of Hybrid system, there is a need for an easy implementation for real processes. The identification task, the modelling relations , the validation, and verification, are complicated and difficult task and currently only one can be done by control expert, a kind of expertise which does not exist in many industries.

The objective of this work is to propose a tool, a software tool, which helps the user to model and interconnect the process with the computer. This work contributes to the understanding of the simple Hybrid systems, and how to optimise them to some expected results. For practical processes, this necessity is present but almost never attained. The purpose of this research is to reduce the effort and make it feasible to use the technology of Hybrid systems in new projects. Rather than leave it to rest on the books and the computers.

In this case the algorithms will be applied to robotics – a domain where improvements are well accepted – it is expected to find a simple repetitive processes for which the extra effort in complexity can be compensated by some cost reductions. It may be also interesting to implement some control optimisation to processes such as fuel injection, DC-DC converters etc.

The most important contribution of this work , is the Alpha approximation for non-linear systems with high dynamics While this kind of process is not typical, but in this case the Alpha approximation is the best linear approximation to use.

With the modelling stage performed OFF-line , this methodology can be implemented even for fast systems. The ON-line part running over the process is capable to control and conduct the system to desired objective. The discrete actions to apply for the system can be implemented with a look-up control table. And a simple optimisation procedure can be applied to obtain the continuous control signals depending on the state vector in the switching state instant.

A lot of work is needed until this method will become a standard practice for control engineers. A unique global theory, a unique methodology, a standard method, have to be adapted, similar to what we know in other areas, e.g. state space form, Grafcet, etc. Different groups need to agree to one acceptable theory and methodology to overpass this subject and go into other topics.

Appendix A

VERIFICATION WITH HYTECH

Hytech is a research tool built in Berkeley, and is used to analyse communication protocols. Particularities of this tool, is the capability to work with intervals, the continuous evolution confined in a range of progression rates, an other is the capability to work with several continuous dynamics with different rates in the same model. This part has some problems in the calculus, but as a research tool it is always under development and solving the problems.

Programming in Hytech language is needed to use a high level language over the C Unix primitives. Which is the operating system which supports the macros and directives of the language. Also it is possible to make models in a graphical interface autograph A2tg, which helps to design the model in states and transitions.

The analysis is necessary to be programmed in a Hytech language, the questions to be analysed must express the methodology to be analysed, Forward ,Backward or both. The program takes little time to analyse a question over the model, for example in a two tanks model in a second time can find the solution.

Hytech can use a symbolic variable in this model, it can be used for design purposes, parameters to be adjusted to accomplish certain conditions. Also it can measure the time ,for example minimum or maximum time between two events, the rate progression for the continuous signals are proportional to the time, and this can be counted.

Restriction of this tool is the necessity of the model for the continuous dynamics in a rate proportional to the time, K* Time. Reset variables are possible at a desired state. This force to model a first or second order model in several states with rates confined in intervals. The modelling error can be confined but is progressive with the modelling procedure.

Next is shown an example of the possibility's modelling with Hytech.
For this example a filling tank is used, with a pump a drain valve and a level detector.
There is a delay between the level of the water and the actuation over the pump.

The analysis will tell us if the level in the tank reaches the top. It is necessary to design the position to place the sensor for the level.

The model of the tank is composed of three states, delay state, ramp state, and the stabilization state, named: ret_on, on_0, on_1. The range intervals of the level progression, give a combination of the lower and upper limits of the possible values of the level in time. All the information is analysed in order to assure the reachability of the proposed condition.

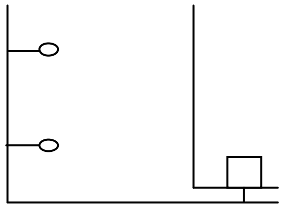
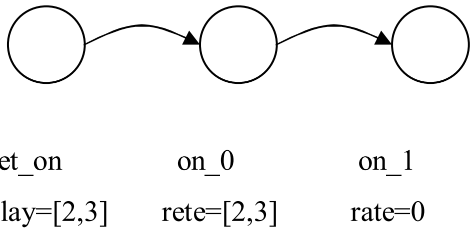


Figure A.1: Schema of the process



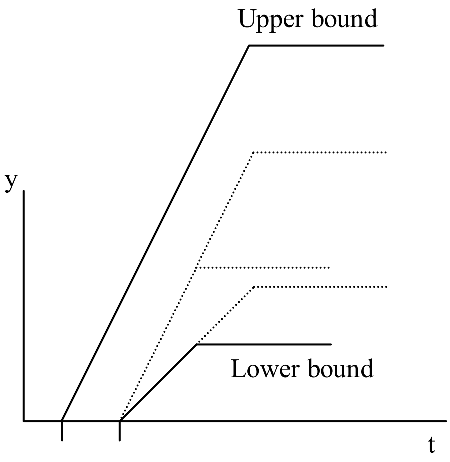


Figure A.2: Limits for the models

```
-- Model water

var
     x,z,y : stopwatch;

automaton model

synclabs:open, close;

initially on_0 & y=10 & x=0 & z=0;

loc ret_on: while z<=5 wait {dz in [2,3], dx=0}
     when z=5 do {z'=0} goto on_0;

loc on_0: while x<=10 wait {dy in [2,3], dx in [1,3/2], dz=0}
     when x=10 do {x' = 0} goto on_1;
     when True sync close do {x' = 0} goto ret_off;

loc on_1: while x=0 wait {dy=0, dx=0}
     when True sync close goto ret_off;

loc ret_off: while z<=5 wait {dz in [2/3,2], dx=0}
     when z=5 do {z'=0} goto off_0;

loc off_0: while x<=10 wait {dy in [-3,-2], dx in [1,3/2], dz=0}
     when x=10 do {x' = 0} goto off_1;
     when True sync open do {x' = 0} goto ret_on;

loc off_1: while x=0 wait {dy=0, dx=0,dz=0}
     when True sync open goto ret_on;
end
```

Control by sensor level measure, and action over the pump.

```
automaton controller

synclabs:open, close;

initially enjegat;

loc enjegat: while y<=16 wait{}
     when y=16 sync close goto parat;

loc parat: while y>=6 wait{}
     when y=6 sync open goto enjegat;
end
```

Analysis, the initial and final conditions and the reachability question.

```
init_reg := loc[water] = on_0 & y=10 & z=0 & x=0;
final_reg :=  y<=3 | y>=19;


b_reachable := reach backward from final_reg endreach;
print b_reachable;

if empty(b_reachable & init_reg)
  then
    prints "Impossible";
  else
    prints "Can do";
endif;

print trace to (b_reachable & init_reg ) using b_reachable;
```

Solution:

```
Number of iterations required for reachability: 3
Location: off_1.parat
     x = 0   & y >= 19
Location: off_0.parat
     x <= 10   & y >= 19
Location: ret_off.parat
     y >= z + 14   & z <= 5   & y >= 12
Location: on_1.enjegat
     x = 0   & y <= 3
|
     x = 0   & y = 16
Location: on_0.enjegat
     x <= 10   & y <= 3
|
     z <= 2   & 3x <= y + 14   & y <= 16
|
     3x <= y + 14   & 3y <= 4x + 8
Location: ret_on.enjegat
     3z + 2y <= 21   & z <= 5   & y <= 16
|
     9x <= 2z + 3y + 32   & y <= 16   & z <= 5

Can do
====== Generating trace to specified target region ========
Time: 0.00
Location: ret_on.enjegat
     x = 0   & z = 0   & y = 10
---------------
 VIA 2.50 time units
---------------
Time: 2.50
```

```
Location: ret_on.enjegat
     x = 0   & z = 5   & y = 10
-------------------------------
  VIA:
-------------------------------
Time: 2.50
Location: on_0.enjegat
     x = 0   & z = 0   & y = 10
---------------
 VIA 2.00 time units
---------------
Time: 4.50
Location: on_0.enjegat
     x = 2   & z = 0   & y = 16
-------------------------------
  VIA: close
-------------------------------
Time: 4.50
Location: ret_off.parat
     x = 0   & z = 0   & y = 16
 ============ End of trace generation ============


================================================================
Max memory used =     0 pages =        0 bytes =   0.00 MB
Time spent       =        0.10u +       0.02s =        0.12 sec
total
================================================================
```

REFERENCES

[Asarin Maler Pnueli 95] Reachability analysis of Dynamical Systems having Piecewise Constant Derivatives. Theoretical C Science 138

[Alur 90] R. Alur, C.Corcoubetis, D.Dill. Model-Checking for real time systems. Proc 5th IEEE Simp. on Logic in Comp. Science 1990

[Alur 91] R. Alur. Techniques of Automatic Verification of Real-Time Systems. Ph.D Thesis, Standfor Univ. 1991

[Alur 95] R.Alur et al. The Algorithmic Analysis of Hybrid systems. Theor. C. Science 138-3

[Alur Corcoubertis Henzinger Ho 93] Hybrid Automata: an algorithmic approach to the specification and verification of Hybrid systems. LNCS 736 Springer-Verlag 1993.

[Alur Dill 94] A theory of timed automata. Theoretical Computer Science 126 1994.

[Alur Henzinger 92] R.Alur T.Henzinger. Logics and Models of Real Time: A Survey. Technical Report TR 92-1262 Dept. Comp. Scien. Cornell Univ.

[R. Alur R. Kurshan 96] Timing analysis in COSPAN. LNCS 1066

[Antsaklis Lemmon 93] P.Antsaklis J.Stiver M.Lemmon. Hybrid systems Modeling and Autonomous Control Systems. Lect. Notes in Comp. Science 736. Springer-Verlag

[Antsaklis 95] X.Yang P.Antsaklis. On the supremal Controllable sublanguage in the Discrete-Event Model of Nondeterministic Hybrid Control Systems. IEEE Trans. on Aut. Control 40-12

[Beneviste 90] A.Beneviste P.Le Gernic. Hybrid Dynamical Systems Theory and the SIGNAL Language. IEEE Trans. on Aut. Control 35-5

[J. Bengtsson K. Larsen F. Larsson P.petterson W. Yi 96] UPPAAL: A tool suite for automatic Verification of Real Time Systems. LNCS 1066

[R. Bellman 66] Dynamic Programming and modern control. Academic press 66

[Billings 87] S.Billings, W.Voon. Piecewise linear identification of non-linear systems. Int. J. Control 46-2

[Billings 89] S. Billings, S. Chen. Extended model set, global data and threshold model identification of severely non-linear systems. Int. J. Control 50-5

[Branicky 94] M. Branicky V. Borkar S.Mitter. A Unified Framework for Hybrid Control. 33 Conf. on Decision and Control.

[Branicky 95] M.Branicky S.Mitter. Algorithms for Optimal Hybrid Control. 34 CDC New Orleans 95

[Branicky 98] M.Branicky V.Borkar S.Mitter. A unified Framework for Hybrid Control: Model and Optimal Control Theory. IEEE Trans. on Aut. Control 43-1

[D.Brajnik Clancy 97] Control of Hybrid Systems using Qualitative Simulation Qualitative. Reasoning. QR-97 Italy

[Cassandras Pepyne 01] Optimal control of a Class of Hybrid Systems. IEEE Trans. A. Control 46-3

[Caines 95] P.Caines J.Wei. The hierarchical lattices of a finite machine. Systems and Control Letters 25.

[G.Campion G.Bastin B.Andrea-Novel 96] Structural properties and classification of kinematic and dynamic models of Wheeled Mobile Robots. IEEE T.on Robotics and Aut. V-12.

[Clarke 80] E.M.Clarke E.A.Emersón. Automatic verification of finite state concurrent systems using temporal logic specification. 10 ACM Symposium on Principles of Programming

[Clark 81] Design and synthesis of synchronisation skeletons using branching-time temporal logic. LNCS 131 Springer-Verlag 1981

[Daoutidis 96] A.Kumar P.Daoutidis. State-Space Realizations of Linear Differential-Algebraic Equation Systems with Control-dependent State Space.IEEE Trans. Aut.Control 41-2

[Delchamps 90] D. Delchamps. Stabilizing a Linear Systems with Quantized State Feedback. IEEE Trans.on Aut Control 35.

[Deshpande 94] A.Deshpande. Control of Hybrid systems. PhD Thesis. Elec. Eng. and Comp. Sciencies. Univ Berkeley 94.

[Deshpande Varaiya 95] A. Deshpande and P.Varaiya. Viable Control of Hybrid systems. Lectures Notes in Comp. Sciences 999, Springer-Verlag, 95

[Deshpande Gollu Varaiya 99] The Shift programming language. IEEE Trans. A. Control March 99

[S. Esteva et al. 98] Description of RogiTeam. Lectures Notes on AI 1410. pp286-294 1998

[S. Esteva 99] Hybrid systems modelling using State Space regime decomposition. STAR99. Barcelona.

[S. Esteva P.ll de la Rosa Aguilar-Martin 00] Optimal path in Hybrid Dynamical Systems. CACSD 2000. UK. Proceedings IEE.

[Flauss Halla 97] Structural analysis of Hybrid Systems modelled by Flow Nets. ECC 97 Brussels

[Frank 90] Fault diagnosis in dynamical systems using analytical and knowledge based redundancy. Automatica 26(3)

[Gertler 91] Structured residuals. Analytic redundancy methods in failure dtection and isolation. Safeprocess 91 Baden Baden

[A. Glua F. Dileseare 91] Supervisory design using Petri Nets. 30 CDC Brighton 91

[Grossman 95] R.Grossman R.Larsón. An Algebraic Approach to Hybrid systems. Theor. Comp. Scienc. 138-1

[Guckenheimer 95] J.Guckenheimer 95. A Robust Hybrid stabilization strategy for equilibria. IEEE Trans. Aut. Control 40-2

[Henzinger 91] T. Henzinger. The Temporal Specification and Verification of Real-Time Systems. PhD Thesis. U. Stanford. 91

[Henzinger 96] The theory of Hybrid Automata. Proc 11 IEEE Symp. Logic in Comp. Science 1996

[Henzinger 98] T.Henzinger P.Ho H.Wong-Toi. Algorithmic Analysis of Nonlinear Hybrid systems. IEEE Trans. on Aut. Control. 43-4

[Henzinger Ho 95] Hytech: The Cornell hybrid technology tool. LNCS 999

[T. Henzinger H. Wong-Toi 95] Linear Phase Portrait approximations for Non-linear Hybrid Systems. LNCS 939

[Hilshorst 92] R.Hilshorst. Supervisory Control of Mode-Switch Processes. Thesis.U.Twentee.

[Iserman 91] Process Fault diagnosis based on process model knowledge. Trans. of the ASME 113 Dec 91.

[Johansen 94] T. Johansen. Operating Regime based Process Modeling and Identification. PhD Thesis. U. of Trondheim 94.

[Johansen 95] T. Johansen, A.Foss. Identification of non-linear system structure and parameters using operating regime decomposition. Automatica 31-2

[T.A.Johanssen B.Foss 95] Identification of Non-linear System Structure and parameters using operating Regime Decomposition. Automatica 31.

[Kohn Nerode 94] W. Kohn A.Nerode. Multiple Agent Hybrid Control: Carrier Manifolds and Chattering Approximations to Optimal Control. 33 Conf. CDC

[Kohn 95] W. Kohn A.Nerode. Viability in Hybrid systems. T. Comp. Science 138-1

[D. Kirk 70] Optimal control theory. Pretice Hall Pub. 70

[B. Kuipers B.Shults 94] Reasoning in logic about Continuous System. Proc. Knowledge representation and Reasoning KR94

[A. Kurhanski Varaiya 99] Ellipsoidal techniques for reachability analysis. LNCS 1790 Springer Verlag 2000

[Laferriere Pappas Sastry] O-minimal Hybrid System. Tech. Repport UCB/ERL M98/29 U. Berkeley 98

[Lafortune 92] S. Lafortune, S.Chung. Limited Lookahead Policies in Supervisory Control of Discrete Event Systems. IEEE Trans. on Aut. Control 37-12

[Lafortune 98] R.Sengupta S.Lafortune. An Optimal Control Theory for Discrete Event Systems. SIAM J. Control Optim. 36-2

[Lafortune Sampath 00] A Discrete Event System approach to Failure Diagnosis: Theory and application. Workshop on Principles of Diagnosis DX-00

[M. Larsson 98] Analysis of diagnosis properties using Discrete Event Dynamic System. CDC98

[M. Larsson 97] On Modelling and Diagnosis of Discrete Event Dynamical Systems. Ph.D. U. Linkoping 97

[M.D.Lemmon P.Antsaklis 93] Event identification in Hybrid Control System. 32CDC 93

[Lennartsón 94] B. Lennartsón B.Edgard M.Tittus. Hybrid systems in Process Control, 33 Conf. Dec. Control 94

[Lunze Nixdorf Richter 97] Hybrid modelling of continuous variable system with applications to Supervisory Control. ECC97 Brussels

[Lunze Nixdorf Richter 01] Process Supervision by means of Hybrid model. J. of Process Control 11

[O. Maler 98] A Unified approach for studing Discretized continuous Dynamical Systems 37 CDC Tampa 98

[McManis 94] McManis,Varaiya. Suspension automata: A decidable class of hybrid automata. Workshop Computer-Aided Verification 1994.

[Murray 94] R. Murray-Smith. A Local Model Network Approach to Nonlinear Modelling. PhD Thesis. U. of Strathclyde, Glasgow.

[R. Murray-Smith F.A.Johanssen R. Shorten 97] State Space modelling using Operating Regime Decomposition and local models. LNCS 1360

[Nerode Khon 93] A.Nerode W.Khon. Models for Hybrid systems: Automata, Topologies, Controllability, Observability. Lectures Notes in Comp. Sciences 736

[Nerode Khon 95] W. Khon A. Nerode. Viability in Hybrid systems. Theoretical Computer Science 138

[X. Nicollin A. Olivero J. Sifakis S. Yovine 93] An approach to the description and analysing of Hybrid Systems. LNCS 736

[Passino 94] K. Passino A.Michel P.Antsaklis. Lyapunov stability of a Class of Discrete Event Systems IEEE Transactions on Aut. Control 39-2

[Patton Chen 92] Robustness in model based Fault diagnosis. Concise Enciclopedia of Simulation and Modelling. Pergamon Press 92.

[Peleties DeCarlo 93] P.Peleties R.DeCarlo . A Modeling Strategy for Hybrid systems Based on Event Structures. Discrete Event Dynamic Systems: Theory and Applications3.

[Pettersón Lennartsón 95] S. Pettersón B. Lennartsón. Time-Optimal Control for a Class of Hybrid systems. European Control Conf. 95

[N.B.O.L Pettit P. Wellstead 95] Analysing Piecewise Linear Dinamical Systems. IEEE Control S. Magazine 15-5

[Pnueli 97] A.Pnueli. The Temporal Logic of Programs. FOCS 77 IEEE 97

[Puri 94] A.Puri and Varaiya. Decidable of Hybrid systems with rectangular differential inclusions. Workshop Computer-Aided Verification 1994.

[Puri Varaiya 94] A. Puri P. Varaiya. Verification of Hybrid systems using Abstractions. Hybrid Systems Workshop.

[Puri 95] A.Puri. Theory of Hybrid systems and Discrete Event Systems. PhD Thesis. U. of California at Berkeley. 95

[Pontryagin Pinch 93] Optimal control and the Calculus of Variation. Oxford Univ. Press New York 93

[Ramadge 89] P.Ramadge. Some Tractable Supervisory Control Problems for Discrete-Event Systems Modeled by Buchi Automata. IEEE Trans. on Aut. Control 34-1

[Ramadge 90] P. Ramadge. On the Periodicity of Symbolic Observations of Piecewise Smooth Discrete-Time Systems. IEEE Trans. on Aut. Control 35-7

[M. Sampath 95] A discrete Event System approach to Failure Diagnosis. Ph.D.U.Michigan 95

[Sampath Sengupta Lafortune 96] Failure Diagnosis using Discrete Event Models. IEEE Trans. A Control 96-4

[Sóntag 81] E.Sóntag. Nonlinear Regulation: The Piecewise Linear Approach. IEEE Trans. on Aut. Control 26-2

[Sóntag 81] E. Sóntag. Nonlinear Regulation: The Piecewise Linear Approach. IEEE Trans. Aut. Control 26-2

[P.Soueres J.Laumond 96] Shortest paths synthesis for a Car-like Robots. IEEE Trans. A Control V41-5

[Stiver Antsaklis93] J. Stiver P.Antsaklis DES Supervisor Design for Hybrid Control, 31 Annal Allerton Conf. on Communic. Control and Computers. 93

[Tagaki Sugeno 85] Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems Man and Cybernetics Vol 15 pp 116-132

[Thistle 94] Manual TCT

[Tittus 94] M.Tittus, B.Egardt. Control-Law Synthesis for Linear Hybrid systems. 33 CDC 94

[Tittus Edgard 94] M.Tittus B.Edgard. Control-Design for Linear Hybrid systems. Tech Report. Control Eng. Lab. Chalmers Univ.Sweden

[Tsisiklis 89] J.Tsitsiklis. On the Control of Discrete-Event Dynamical Systems. Math. Control Signals Systems 89-2

[Utkin78] Sliding mode and their application in Variable Structure System. Mir Publishers Moscow 78

[Wong 96] K.Wong W.Wonham. Hierarchical Control of Discrete-Event Systems. Disc. Event Dyn. Systems: Theory and App.

[Wonham 87] W.Wonham P.Ramadge. Supervisory Control of a Class of Discrete Events Processes. SIAM J Control Optim. 25-1

[Wonham 88] W.Wonham P.Ramadge. On the Supremal Controllable Sublanguages of a Given Language. SIAM J. Control Optim. 25-2

[Yovine 97] Kronos: A verification tool for Real-time Systems. Int. J of Software V1 Springer Verlag 1997

[Y. Zang A.Mackworth 95] Real Time control of soccer playing robots using off-board vision. IEEE Int. C. on System Man and Cybernetics V-4

[Zhong 90] H. Zhong W. Murray. On the Consistency of Hierarchical Supervision in Discrete-Event Systems. IEEE Trans. on Aut. Control 35-10