Incremental Controller Networks: a comparative study between two self-organising non-linear controllers

Eric Ronco and Peter J. Gawthrop
Centre for System and Control
Department of Mechanical Engineering
University of Glasgow
ericr@mech.gla.ac.uk & peterg@mech.gla.ac.uk

Technical Report: CSC-97011

November 13, 1997

Abstract

Two self-organising controller networks are presented in this study. The "Clustered Controller Network" (CCN) uses a spatial clustering approach to select the controllers at each instant. In the other gated controller network, the "Models-Controller Network" (MCN), it is the performance of the model attached to each controller which is used to achieve the controller selection. An algorithm to automaticly conctrust the architecture of both networks is described. It makes the two schemes self-organising.

Different examples of control of non-linear systems are considered in order to illustrate the behaviour of the ICCN and the IMCN. It makes clear that both these schemes are performing much better than a single adaptive controller. The two main advantages of the ICCN over the IMCN concern the possibilities to use any controller as a building block of its network architecture and to apply the ICCN for modelling purpose. However the ICCN appears to have serious problems to cope with non-linear systems having more than a single variable implying a non-linear behaviour. The IMCN does not suffer from this trouble. This high sensitivity to the clustering space order is the main drawback limiting the use of the ICCN and therefore makes the IMCN a much more suitable approach to control a wide range of non-linear systems.

1 Introduction

For the purposes of control, it is essential that the chosen class of models is *transparent* in the sense that the model structure and parameters may be interpreted in the context of control system design. For example, neural networks are widely used for system modelling purposes because of their ability to represent non-linear functions. However, most of the neural networks develop unclear representations of the system; therefore system analysis can not be carried out easily and thus the neural model is less useful for control purposes. This black box modelling is the main feature that restricts the use of neural networks for system modelling and control.

However, there are a couple of neural networks that appears to be very suitable for modelling oriented control purposes. This is due to their use of multiple linear models to approximate the behaviour of a system. From each of the linear models it is straight forward to design a controller. Therefore the non-linear overall model developed by those networks can be easily transformed into a non-linear controller. This is perhaps the only general and systematic approach to designing a non-linear controller out of a non-linear model.

One of these neural networks is the “Local Model Network” (LMN) introduced in (Poggio and Girosi, 1990) and further extended for modelling and control purposes by (Johansen and Foss, 1993; Johansen and Foss, 1992). The control version of the LMN is the “Local Controller Network” (LCN). The idea underlying the other network was introduced in (Middleton *et al.*, 1988) and further extended in (Morse, 1990; Morse *et al.*, 1992; Weller and Goodwin, 1994) and coined as the “hysteresis switching algorithm”. This algorithm aims at achieving stability whereas, the “Multiple Switched Model” (MSM) extensively studied by (Narendra *et al.*, 1995; Narendra and Balakrishan, 1997), but closely related to the former algorithm, is used for improving the control performance whilst dealing with systems having their parameters changing quickly through time (e.g. non-linear systems).

The basic idea of the LCN and MSM is to develop and use various controllers at different operating regions of the system. These algorithms differ mainly by the method used for the selection of the controllers at each instant. In the LMN, the controllers are selected according to a spatial clustering of the operating space whereas in the LMN the selection of the controllers is clustering free.

These two LCN and MSM have important advantages in common. Their properties can be easily extracted since linear theory can be applied to analyse each of the linear controllers composing the network. “Learning” is extremely quick due to the use of regression methods (e.g. least squares) for the estimation of the model-controller parameters. This makes these approaches very suitable for online control purposes. Another advantage of these algorithms is that they do not suffer from the “stability-plasticity dilemma” which is a basic design problem for learning machine as emphasised by (Carpenter and Grossberg, 1988): while the model is adapting to an operating region of the system it is forgetting previous adaptations regarding other regions. The LCN and MSM are not exposed to this dilemma because each model-controller is specially adapted for a different operating region of the system. Hence, and as highlighted by (Narendra *et al.*, 1995), these two schemes can be adapted for different discontinuities of the system.

However, to make the LCN and MSM self-organising requires a general and systematic method to automatically construct their network architecture. This is the purpose of this study to develop such an automatic network construction algorithm.

The outline of this study is as follows. The next section compares, in the frame of non-linear control, a linear adaptive controller and a multiple linear controller. The latter controller embeds the basic mechanisms of the LCN and MSM. The automatic network construction method developed in this study is then described. This method implies significant modifications of the LCN algorithm. The modified algorithm is called the “Clustered Controller Network” (CCN). The automatic network construction algorithm is referred to as the “Incremental Network Construction” (ICN) and the algorithm embedding the INC is called the “Incremental Clustered

Controller Network" (ICCN). The following section describes the incremental network construction method applied to the "Model-Controller Network" (MCN) (a modified version of the MSM). The results obtained by the ICCN and the IMCN whilst controlling various non-linear systems are then compared. The results are recalled during the conclusion to discuss the efficiency of these two approaches.

2 Adaptive linear control versus multiple linear control

The purpose of this section is to briefly introduce the basis of the multiple-linear control approach. The control of a non-linear system is considered. This system is described by the following equation

$$sy = 2.5u - 2sin(y) \tag{1}$$

where $s \equiv \frac{d}{dt}$ is a differential operator, $sin(y)$ is the system's non-linearity and $0 \leq y \leq pi$.

This function is non monotonic since a change of sign occurs around the operating condition $y = \frac{1}{2}\pi$. This makes this system a difficult problem to control using the "Model Reference Adaptive Controller" (MRAC) used in this study. This MRAC is an indirect design method, therefore the controller design is based on a local model of the non-linear system. Each of the first order linear model used for the local modelling of the system (1) is of the following form

$$s\hat{y} = bu - ay + c \tag{2}$$

where c is a constant that makes the above equation non-homogeneous. This is important in the case of a model network since the local models linearise the system at any position (at equilibrium or not).

From each of the local models one can easily work out the controllers equation. To do so a model of the closed loop system has to be specified. It was arbitrarily decided that the system will have to settle down in 3sec with an error inferior to 0.01% of the desired set point r. This leads to the following reference closed loop model:

$$s\bar{y} = 1.5351(r - y) \tag{3}$$

where 1.5351 is the reference model parameter obtained with $tol = 1\%r$ and the settling time $t_s = 3s$.

Now, from the equalisation of the system model (2) and the reference model (3) we determine the linear equation describing the various local controllers composing the CCN (see for further details about this control design method chapter 2 in (Ronco, 1997)):

$$u = \frac{1}{b}(1.5351r - y(1.5351 - a) + c) \tag{4}$$

You see from this equation that the control design is only a matter of determining the value of the model parameters a, b and c. The singular value decomposition method is used in this study for the estimation of these parameters (see chapter 2 in (Ronco, 1997)).

We can generalise this control design method for any SISO system of order N:

$$u = \frac{1}{b_N}\left(c - \sum_{i=1}^{N-1} b_i s^{i-1} u + \sum_{i=1}^{N} \theta_i^r s^{N-i} r - s^{N-i} y(\theta_i^r - a_i)\right) \quad (5)$$

The control of a non-linear system is often achieved through the use of a linear controller designed from a linearisation of the system. The resulting controller is only valid for a local region of the system. This is shown is figure 1. The non-linearity of system (1) and its linearisation used for the controller design are depicted in the bottom of figure 1. The top subplot depicts the control performance of this controller concerning two consecutive control sequences. During the first control sequence we see that the control performance is correct i.e. a small steady state error is achieved after a quite accurate matching of the desired transient system output. This was expected since the system's non-linearity is quiet accurately approximated by its linearisation in the range $y[0 \quad 0.63]$ (where 0.63 is the control goal). However, it is not sensible to use the same controller to drive the system in the range $y[0.63 \quad \pi]$ (see the second control sequence in figure 1). In this range the linearisation of the system diverges significantly from its non-linearity.

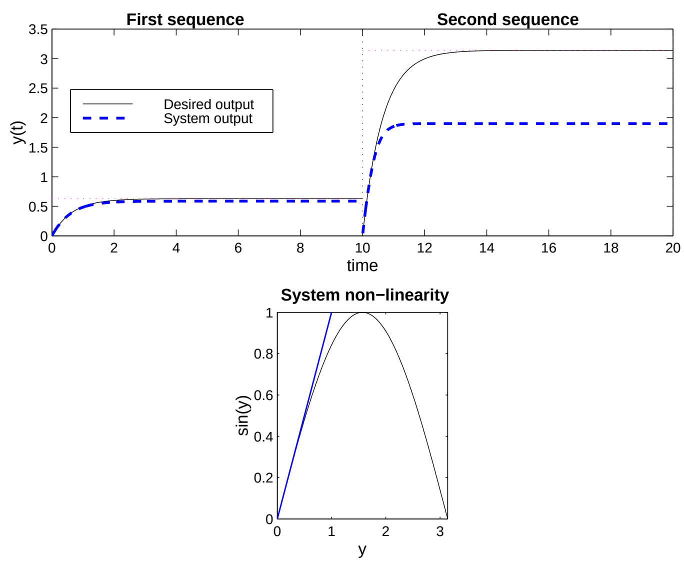

Figure 1: Performance of a linear controller

The top subplot depicts the control performance of the controller (designed from the linearisation of system (1) achieved around $y = 0$) concerning two consecutive control sequences. During the first and second control sequences a desired transient (see plain line curves) has to be achieved whilst respectively driving the system (see dotted line curves) toward the desired position $y = \frac{\pi}{5} = 0.63$ and $y = \pi$. The bottom subplot shows the system's non-linearity in the range $0 \leq y \leq pi$ and its linearisation used for the controller design.

Hence, a linear controller has a certain region of validity beyond which its performance becomes poor. One standard way to overcome this problem is to continually adapt the identification (i.e. the linearisation) of the system and thus the controller; this is conventional adaptive control. Such a method can only be effective if the dynamics of the system are changing smoothly and quite slowly through time. Therefore, if the function is discontinuous adaptive control can not be applied. In addition, the slowness of such an adaptation may result in a large transient error (Narendra *et al.*, 1995). This is illustrated in figure 2. The adaptive controller has been preadapted from a linearisation of the system around $y = 0$ (see left bottom plot of this figure). Hence the controller performs well in the operating region where the linearisation is valid. In the second control sequence the system has to move from $y = 0$ to $y = \pi$. We can see that from the top plot of figure 2 that the adaptive controller is eventually going to reach a small steady state error. However, the transient is much slower than the desired one. A more serious problem occurs during the third control sequence. While during the second control sequence the controller became adapted for the control of the system around $y = \pi$ it forgot its former adaptation corresponding to the linearisation of the system around $y = 0$. This resulted in a control performance worse than during the second sequence. This effect is a basic machine learning problem referred as the "stability-plasticity dilemma" (Carpenter and Grossberg, 1988).

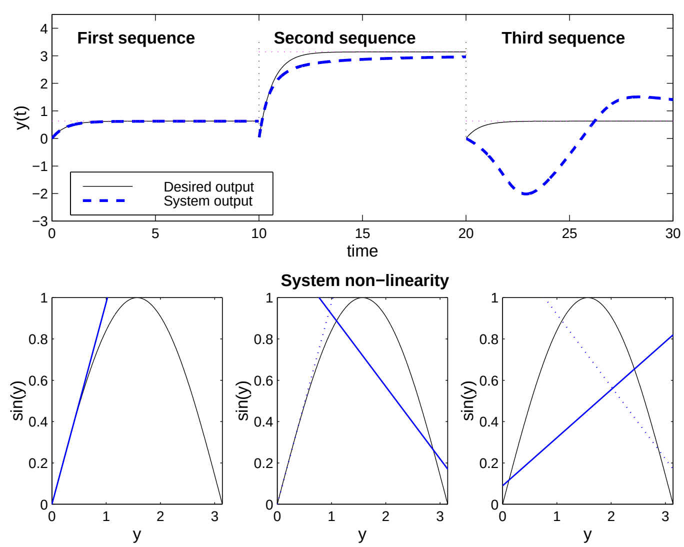

Figure 2: Performance of a linear adaptive controller

The top subplot depicts the control performance of an adaptive controller (preadapted from the system (1) linearisation achieved around $y = 0$) concerning three consecutive control sequences. During the first, second and third control sequences a same desired transient (see plain line graphs) has to be achieved whilst respectively driving the system (see dotted line graphs) toward the desired position $y = \frac{\pi}{5} = 0.63$, $y = \pi$ and $y = \frac{\pi}{5} = 0.63$. The three bottom subplots depict the system's non-linearity in the range $0 \leq y \leq \pi$ and its linearisations achieved by the adaptive controller at the last stage of each control sequence (see plain straight line; the dotted lines correspond to the system linearisation at the first stage of each control sequence).

A simple way to avoid the "stability-plasticity dilemma" is to use a number of controllers each valid for a different operating region of the system. To show the effectiveness of this scheme, the results obtained whilst using only two different controllers active each for a different half of the whole operating range of the system have been depicted (see figure 3). The controllers obtained at the last stage of the first and second adaptive control sequences previously described have been used (see bottom of the figure 3). The control performance although not perfect (the second controller is not very well adapted to its operating region) overcomes the stability-plasticity dilemma and the transient is as quick as desired.

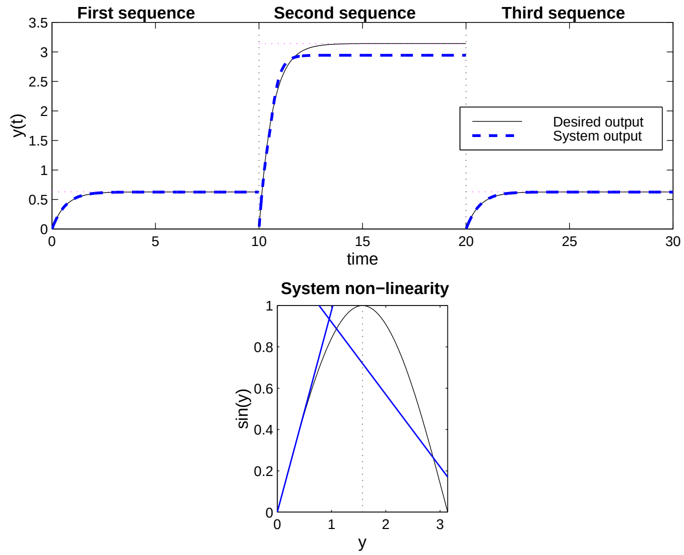

Figure 3: Performance of two linear controllers for the control of system (1).
The top subplot depicts the control performance of this scheme concerning three consecutive control sequences. During the first, second and third control sequences a desired transient (see plain line graphs) has to be achieved whilst respectively driving the system (see dotted line graphs) toward the desired position $y = \frac{\pi}{5} = 0.63$, $y = \pi$ and $y = \frac{\pi}{5} = 0.63$. Each controller is active for a different half of the operating range of the system. The bottom subplot shows the system's non-linearity in the range $0 \leq y \leq \pi$ and the two linearisations used to design the controllers (see plain straight lines).

Hence both problems of an adaptive controller can be avoided using a multiple linear controllers scheme to control non-linear systems. However there are few problems to sort out before applying systematicly this scheme. Solutions to these problems are exposed in the next section.

3 Clustered Controller Network

The development of an automatic network construction algorithm is difficult to achieve if one maintains the architecture of the Local Controller Network (LCN). Therefore, this algorithm has been simplified whilst developing the Clustered Controller Network (CCN). This section aims at describing this algorithm. Note that the modelling version of the CCN is closely related to it since one model is linked to each controller in the CCN. We could say that a Clustered Models Network (CMN) lies inside a CCN. Hence, although we are mainly going to deal with the CCN bear in mind that the basic features of this algorithm applies also to a CMN.

Similarly to the LMN, the selection of the controllers in the CCN is achieved through a clustering of the operating space. As discussed in (Ronco *et al.*, 1996*a*) and extensively in (Ronco, 1997) (chapter 1), the use of radial basis functions (rbfs) for the clustering is most effective for single dimensional space. A compromise between approximation and clustering quality has to be found for clustering spaces of dimensions superior to one. This problem is non trivial and its complexity increases quickly with the dimension of the space. This is why the clustering should always be performed on a single dimension space. The ideal situation would be to cluster on quantities which remain one dimensional for high-order systems; for example the control error. Such a quantity that could be relevant for most of the SISO systems has not yet been found. Hence, the choice of this quantity is an important issue in applying the CCNs.

In most of the cases studied so far the clustering has been achieved on the space shaped by y_{t-1} (the system output at time t-1). The clustering of this single quantity is only effective for cases where the system output is the main variable inducing non-linearity in the system. For other cases, where other variables can imply non-linear behaviour (e.g. the velocity of the system), to perform a multi-dimensional clustering should be more relevant even if we bear in mind the difficulty of performing a multi-dimensional clustering (see above discussion).

However, there are other features that further justify the clustering on a single quantity rather than on a multiple one. Those advantages are related to the fact that in the single dimension case it is straightforward to determine the neighbourhood of the operating condition Φ. The neighbourhood always implies the two controllers attached to the two rbf surrounding the operating condition (see the graph entitled "single dimensional clustering in figure 4). This neighbourhood has a predominant effect on the quality of interpolation between controllers that is necessary to smooth the behaviour of the controller network. This interpolation is straight forward in the single dimension clustering case. In the multi-dimension case the neighbourhood is much more unclear. In the two dimensional case plotted in figure 4 at least six rbfs can be considered as being in the neighbourhood of the operating condition Φ. Only a few of them are going to be relevant for this operating condition. Interpolating between all of these neighbour controllers can lead to catastrophic results if some of them are not fitted for the operating condition. In addition, this unclear neighbourhood makes the behaviour of the controller network difficult to interpret. In the single dimension case the analysis of the network of controllers can be reduced locally to a single controller without losing any of the significance of the analysis. This can not be done in the multi dimensional case. There are too many interactions between controllers. This interaction also makes the parameters identification of the local controllers difficult whereas this is straight forward in the single dimension case. A regression method (e.g. least squares) can be applied to speed up the adaptation and to ensure a convergence towards a solution (in the least error squares sense). It is often required to apply a gradient method to identify the parameters of the controllers composing the network that involves a multi dimensional operating space. This makes the adaptation slow and does not ensure to reach a satisfactory solution. This involvement of the majority of the controllers, to determine the output of the network, tends also to make the computation intensive. In the single dimension case only the two neighbours are involved in the network output at each instant. This makes the computation very quick or at least does not relate it to the number of controllers composing the network. A last advantage, as we are going to see in the next section, is that the clustering of a single quantity facilitates considerably the automatic network construction. It is clear from these advantages that, whatever the characteristics of the SISO system is, it is recommended to cluster on a single quantity rather than on a multiple one.

In the CCN each rbf is characterised by a centre and a "single dimension width" (see figure 5). The selection of each rbf is performed using a "winner takes all" like method. Each time we select the two rbfs in the neighbourhood of the operating condition Φ. This is achieved by computing D for each rbf.

$$D_i = |centre_i - \Phi| - width_i \tag{6}$$

The rbf_i having the smallest D is selected. The other rbf in the neighbourhood of Φ is simply

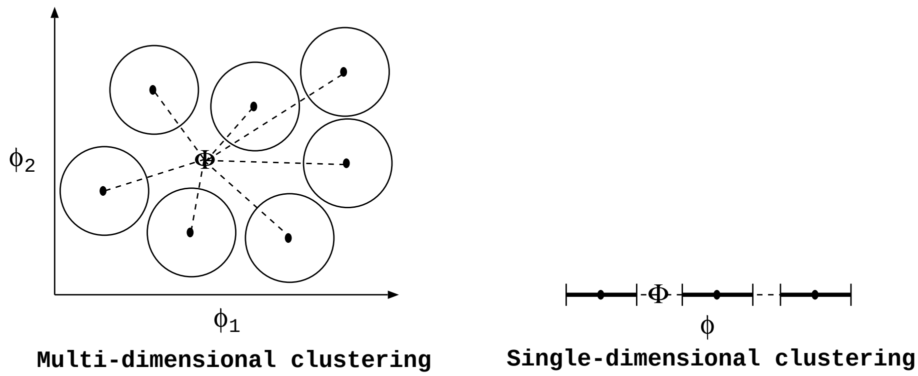

Figure 4: Multiple vs single dimension space clustering

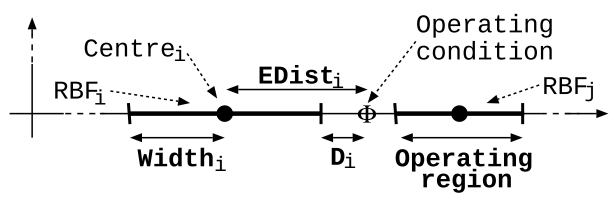

Figure 5: Components of the radial basis functions used for the clustering of a single dimension space.
Two radial basis functions are depicted. Each one is composed of a centre and a width.

the neighbour of rbf_i in the opposite side of Φ: the neighbour of rbf_i is $rbf_{i+sign(\Phi-center_i)}$. Note that this neighbour relationship between rbfs leads to an architecture that could be related to a Kohonen's neural network (see (Kohonen, 1982) for details about this neural network).

Now, by relating a different controller to each rbf we end up with a Clustered Controller Network (CCN) (see figure 6). Since at each instant only two rbfs are selected, the two controllers attached to those rbfs are activated as well. The selection of two controllers at each instant enables an interpolation between their outputs. This is important to avoid sharp controller switching and thus non-smooth behaviour of the CCN. Hence, the CCN output u corresponds to a weighted sum of the two selected controllers' output:

$$u = u_i w_i + u_j w_j \tag{7}$$

where w_i and w_j are the two weighting parameters respectively corresponding to the activity of rbf_i and rbf_j. The determination of the activity of each rbf is achieved according to the block diagram depicted in figure 7.

The following sigmoid like function is used to perform the interpolation.

$$Act_i = \frac{1}{1 + e^{pD_i}} \tag{8}$$

where $p = \alpha \frac{1}{|center_j - center_i|}$ is used to maintain the same shape of the sigmoid function (see figure 8) whatever the size of the single dimension operating space is. This is to avoid the normalisation of the clustering space (as it is achieved when applying the local models network or local controller network (see for instance (Johansen and Foss, 1993)). The parameter α in

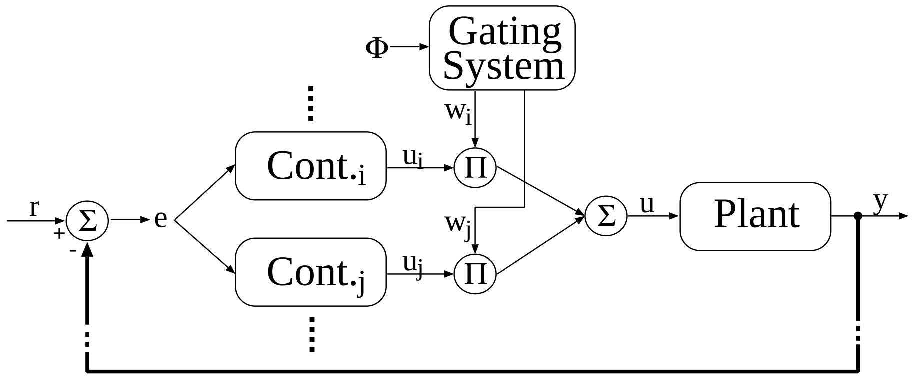

Figure 6: The controller network closed loop system.
The output of each selected controller u is weighted by the activity w of its connected rbf. Hence, the output of the network is a sum of the weighted output of the two selected controllers.

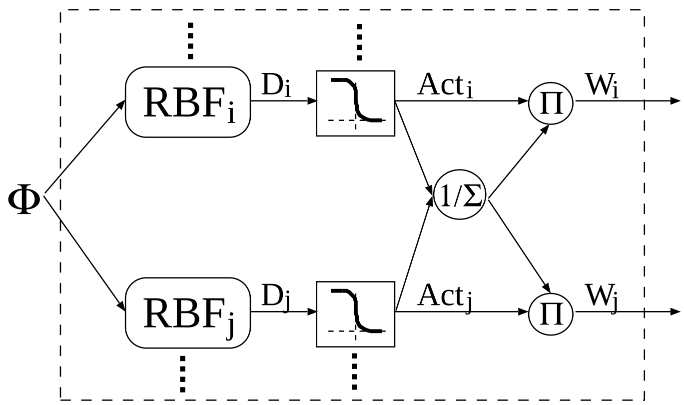

Figure 7: The gating system
The two rbfs in the neighbourhood of the operating condition Φ are activated. The other rbfs are kept at rest. Each rbf has got a certain activity w depending on its region of activity and the position of the operating condition Φ. A fixed sigmoid like function is used to enable an interpolation between their activity.

the variable p is the coefficient that drives the sharpness of the sigmoid. $\alpha = 10$ was used in this study, which leads to the sigmoid shape depicted in figure 8.

From figure 8, and taking the rbf_i for example, you can see that the activation "Act_i" of the rbf_i will tend to 1 when the operating condition Φ will be inside the activity region $|center_i| + |width_i|$ of the rbf_i, whereas the activity of the rbf_i will tend to 0 when the operating condition Φ will be outside of its activity region. To make sure that the sum of the activity of the two rbf is 1, the activation "Act" of each rbf is weighted by the activations' sum of the two rbfs. The activity w is then determined from the weighted activation of each rbf. The activity w of each rbf will be the one used to weight the output of its connected controller.

As previously said, this approach could be used also for the clustered models network. This latter case is depicted for clarity (see figure 9).

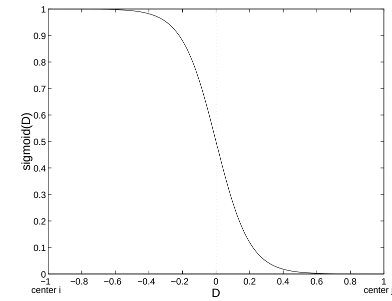

Figure 8: Shape of the sigmoid function used for the interpolation between controllers' output

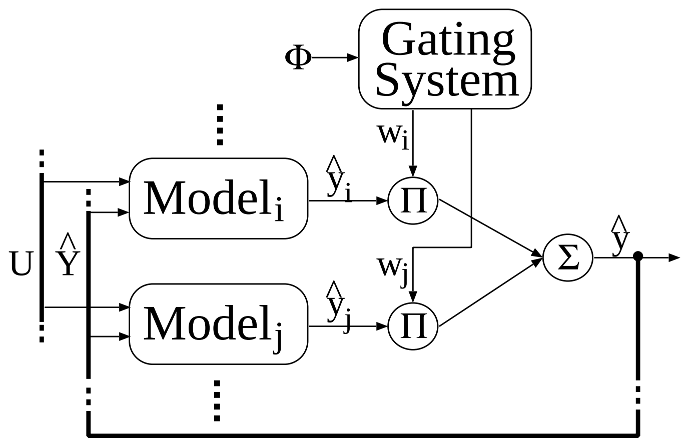

Figure 9: The clustered models network.

The output of each selected model $\hat{y}$ is weighted by the activity w of its respective rbf. Hence, at each instant, the output of the network is a sum of the weighted output of the two selected models.

4 Incremental Clustered Controller Network

The purpose of this section is to describe the main method uses in this study to determine the architecture of the Clustered Controller Network (CCN): the number of controllers and their region of activity. Bear in mind again, that although we are going to deal with the CCN, most of the features that are going to be exposed here applies also to the Clustered Models Network (CMN).

As discussed in (Ronco, 1997) (chapter 1) an homogeneous spread of the basis functions is not a sensible approach because the non-linearity of a system and the input-output samples are not homogeneously spread. A small region can be highly non-linear whereas a large region can be almost linear. Few samples can be available in a large region whereas a large amount of samples could be distributed into a very small region. Both these problems would affect the parameter approximation quality of the controllers composing the network. Therefore it is vital to find out the required number of controllers and their region of validity in order to efficiently apply a controller network. Note that during the literature review preformed in (Ronco, 1997) (chapter 2) it was emphasised that there is no general and systematic approach to solve that clustering problem.

To determine the architecture of a CCN without any a priori knowledge about the system is indeed a non-trivial problem. Our first attempt to develop a network construction was very much inspired from some biological insights. This method is exposed next, followed by a section dedicated to an extended version of this method that should be applicable for most cases.

4.1 Progressive Control Design

It is clear that the human system never learn how to perform a task at once but tends to progressively learn the complexity of a task (Ronco, 1994; Szilas and Ronco, 1995). (Ronco, 1994; Szilas and Ronco, 1995) referred to this learning strategy as "the progressive complexity learning". This learning strategy has been applied in (Ronco *et al.*, 1996*b*; Ronco *et al.*, 1996*c*) to determine the controller network architecture required to balance the inverted pendulum (see figure 10) around the angle $\theta = 0$. The inverted pendulum can be described by the following equation:

$$\ddot{\theta} = \frac{\frac{1}{2} mgl.\sin\left(\theta\left(t-1\right)\right) + \tau}{\frac{1}{3} ml^2} \tag{9}$$

It is a simple problem if the pendulum is initialised near $\theta = 0$ but it is non-trivial if the pendulum is initialised upside down i.e. $\theta_0 = 180$. The latter case was studied.

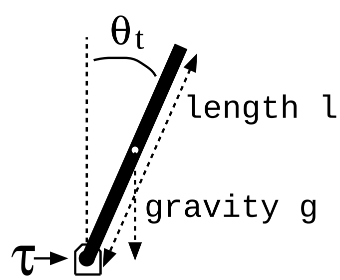

Figure 10: The inverted pendulum

To determine the network architecture a "progressive control design" was performed. The underlying idea of this algorithm is to incrementally build the CCN whilst the initial transient control error e ($e = y_0 - r$ where y_0 is the initial system output and r is the set point, the control goal) is progressively increased. In other words, instead of trying to control the pendulum initialised upside down at once, a progressive increase of the complexity of the control problem is achieved. Different successive control sequences, that differed only in the initial position of the pendulum, were performed. The first control sequence was the simplest since the pendulum was initialised very close to the control goal $\theta = 0$. The distance (i.e. the initial error e) between the set point r and its initial position y_0 was increased during the following stages. The last stage of the progressive control design was the most difficult one since the distance e between the set point r and the initial position y_0 was the maximum (see figure 11 for some of those stages).

This learning strategy does not solve the problem of the network construction but it constitutes a simple and very efficient method which ensure to drive the system in most of its

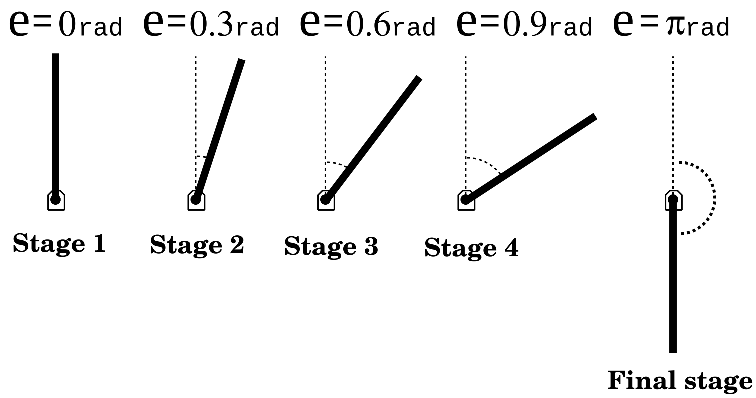

Figure 11: Stages of the progressive control design applied to the inverted pendulum

operating conditions. On this basis it becomes straight forward to construct the CCN. During the control of the system one looks at the actual control error. If the control error of the CCN exceeds a certain threshold it means that the control is unsatisfactory. The control is efficient enough in other cases. Hence, each time the control becomes unsatisfactory a controller is added to the network. Otherwise, when the control is efficient enough, the activated controller (only one controller is activated each time) is updated as well as its connected basis function so that its region of activity covers the actual operating condition and the others for which it was previously valid (see next section for more details about the way the controllers are added to the network and the updating of the controllers design and their operating region).

This fairly simple method makes the network construction straightforward. However this method can not be widely applied since not all the systems can be initialised at any position. Thus, in this study this approach has been generalised to make it applicable to most of the SISO systems. This method is exposed in the next section.

4.2 Incremental Network Construction

As just shown, the problem of the network construction can be easily solved if one can drive the system into most of its operating conditions. This is indeed difficult to perform in the case of a multi-dimensional clustering. The number of possibilities increases very quickly with the system order. One way to reduce this problem would be to consider only operating conditions where the system is in equilibrium. This would be too restrictive since many systems have very few equilibrium points (e.g. unstable systems). Hence, this constitutes another argument for constraining the clustering to a single quantity.

In most cases it is trivial to cover all the operating conditions involved by a single quantity. The network construction can be achieved on-line whilst controlling the system. At each instant, two actions are possible depending on the control performance achieved by the CCN. The control accuracy is reflected by the control error $\bar{e} = \bar{y} - y$ (where $\bar{y}$ is the desired output given by a reference closed loop model) achieved by the CCN. If this error can not be determined the modelling error $\hat{e} = y - \hat{y}$ can be used as an indirect measure of the control efficiency. This can be done since each controller is designed from a model. Hence to be general let us consider a control efficiency criteria ϵ rather than the control error $\bar{e}$ or the modelling error $\hat{e}$. There are two possible actions to modify the architecture of the CCN:

- *If* $\epsilon \leq threshold$:
  To facilitate the updating of the control design only one controller must be selected at each instant among the two controllers selected by the CCN. If the operating condition belongs to the region of activity of one controller this latter is automaticly selected. In the other case, the most valid controller for the current operating condition must be selected. To do so we can not use the control error because it is not possible to test the control accuracy performed by each controller (only one control signal can be inputed to the plant at each instant). The distance between the operating condition and the rbf attached to each controller is not a relevant criteria to determine which of the two controllers is the most effective. Therefore we use the modelling error $\hat{e}$ as an indirect measure of the control efficiency. One selects the controller having its connected model performing the best modelling of the actual input-output sample of the system behaviour.
  The selected controller is then updated in order to integrate the new sample and to be adapted to small changes in the environment. In addition, the rbf connected to the selected controller is updated in order it covers the current operating condition as well as previous ones for which the controller was previously valid.

- *If* $\epsilon > threshold$:
  A sampling of the system is achieved around the actual operating condition (this is usually preferable than continuing the control that can become totally inaccurate if the change of parameters value related to the operating condition is abrupt e.g. case of a discontinuous function). From those samples a local model of the system is determined. A linear controller is then designed from this model. Any conventional control design methods can be used (e.g. pole placement, Model Reference Adaptive Controller. The latter is used in this study. Details about this control design approach can be found in the (Ronco, 1997) (chapter 1). A rbf is centred on the current operating condition and its width is set to almost zero. The new controller-rbf pair is finally added to the CCN.

These features are the basis of the "Incremental Network Construction" (INC). This algorithm has been first introduced in (Ronco and Gawthrop, submitted*b*). A pruning feature has been recently added to the INC to remove the controllers that are not (or are unlikely to be) robust enough (see (Ronco and Gawthrop, submitted*a*; Ronco and Gawthrop, 1997)). It happens that some controllers perform very badly at some operating conditions. This can be interpreted as a lack of robustness that could lead to unstable behaviour of the overall controller. There are two cases involving a pruning. We prune the controllers that have not learned enough i.e. the number of samples encountered is less than the number of parameters of the model attached to the controller. We also prune the controllers which on average have a modelling error greater than twice the threshold used for the network construction. This is indeed arbitrary but sufficient in most cases to remove the undesirable controllers. A better criteria could be found in the singularity vector expressed by the Singular Value Decomposition (SVD) method used for the models approximation (see (Ronco, 1997) (chapter 2) for details). However this possibility has not been sufficiently investigate to be exposed here.

After a while, when most of the operating conditions have been reviewed it is suggested to switch from this learning stage to a generalisation where the architecture of the network is no longer updated. The controllers should remain adaptable in case some small changes occur in the environment. The advantage of this generalisation stage is that it is much quicker than the incremental one.

Although this algorithm could look dedicated for the construction of the CCN architecture, it is straight forward to apply it for the construction of the Clustered Models Network. It is even simpler since no control design will be required. The only difficulty without a controller is to have a method to efficiently sample the plant, but this is a known problem in system identification.

Note that in the context of gating through spatial clustering, this network construction algorithm is the only one developed so far which takes into account the capability of the models/controllers to determine their operating region. This is indeed an important feature since this ensures that each model/controller is valid for its operating region. This is however a common feature with another neural network which also applies a gating approach to select different computing modules at each instant: the "Hierarchical Mixture of Experts" algorithm developed by (Jacobs and Jordan, 1993) and extensively described in (Ronco, 1997) (chapter 1).

4.3 Illustration

In order to illustrate the capability of the INC associated to the CCN we will consider the control of the first order system (1) used in the first section of this article. We recall the function representing this system:

$$sy = 2.5u - 2sin(y) \tag{10}$$

where $s \equiv \frac{d}{dt}$, $sin(y)$ is the system's non-linearity and $0 \leq y \leq \pi$.

Each of the first order linear controllers composing the CCN are described by the following equation which was derived in section 2:

$$u = \frac{1}{b}(1.5351r - y(1.5351 - a)) \tag{11}$$

where b and a are the two parameters of a first order local model of the system.

We wish to design a CCN adapted for the control of the system (1) in its full operating range, that is to say $y[0 \quad \pi]$. The control adaptation to this system by the CCN consisted in a series of five control sequences with respective set points $\pi * 1/5$, $\pi * 2/5$, $\pi * 3/5$, $\pi * 4/5$ and π. This simple learning strategy is important to ensure that the system has been properly excited and therefore driven in most of its possible states. The INC was applied during this learning stage. The threshold determining the control efficiency performed by the CCN was set to 0.01 i.e. we wish the CCN to never make a control error $\bar{e} = \bar{y} - y$ superior to 0.01. The INC assigned five controllers to the CCN.

To test its generalisation capability, the CCN has been applied to control system (1) on three different control situations depending on three different set points (i.e. control goals). The three control sequences are characterised respectively by a set point $r = pi/5$, $r = pi$ and $r = pi/5$. This is the same control problem used in the first section of this article to evaluate the efficiency of an adaptive controller whilst controlling the non-linear system (1).

The results obtained from the CCN are depicted in figure 12. In the bottom subplot of this figure you can see the five local linearisations of the system used for the design of the controllers.

It is clear that, if one limits each linearisation to its operating region (see the operating regions of each controller on the x axis), the overall model is a perfect match of the non-linear system. This is indeed the reason why the performance of the CCN is so good (see the top sub-plot of figure 12). The system output (dashed line) matches perfectly the desired transient (plain line). Note that on the y axis you can see the operating region of each of the controllers.

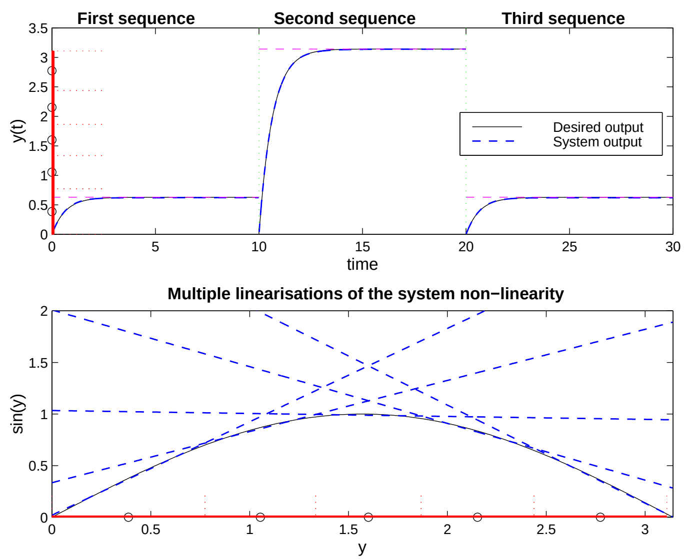

Figure 12: Performance of the CCN whilst controlling system (1).

The top subplot depicts the control performance of this scheme concerning three consecutive control sequences. During the first, second and third control sequences a desired transient (see plain line graphs) has to be achieved whilst respectively driving the system (see dotted line graphs) toward the desired position $y = \frac{\pi}{5} = 0.63$, $y = \pi$ and $y = \frac{\pi}{5} = 0.63$. Five controllers are active at a different operating region depicted on the y axis. The bottom subplot shows the system's non-linearity in the range $0 \leq y \leq \pi$ and the five local linearisations of the system used to design the controllers (see plain straight lines).

These results are far better than those obtained from the single adaptive controller (see figure 2). More importantly this shows that although the ICCN is a simple method it is very efficient. The CCN is not subject to the stability-plasticity dilemma and it is valid for the full operating range of the system.

4.4 Concluding remarks

The architecture of the CCN and CMN have been presented. The CCN and CMN are respectively modified versions of the LCN and LMN. The main difference is that the clustering is achieved on a single quantity in the CCN whereas the LCN clusters on multiple quantities. There are several important advantages arising from this simple clustering. Among them is the facility to determine the neighbourhood of an operating condition. This leads to a better interpolation capability of the network as well as a straight forward understanding of the network activity. The activation of no more than two models/controllers each time has also the advantage of implying very few computations.

Another important advantage of clustering on a single quantity is that it highly simplifies the problem of developing an automatic architecture construction of the network of models/controllers. The "Incremental Network Construction" (INC) algorithm was developed and described in this study. This algorithm constructs the network architecture on-line according to the performance of the network. It determines the number of models/controllers required to model/control an unknown system as well as the operating region of each model/controller. The INC therefore gives a complete autonomy to the CCN/CMN.

However, in case that more than a single quantity involves important non-linear behaviour in the system, the clustering on a single quantity may not be efficient enough. Rather than clustering on multiple quantities one should use an other approach to select the models/controllers. A clustering free approach is described next.

5 Incremental Model-Controller Network

The "Multiple Switched Model" (MSM) has been extensively studied in (Narendra *et al.*, 1995; Narendra and Balakrishan, 1997) (See (Ronco, 1997) (chapter 1) for details about this algorithm). The MSM is a network of model-controller pairs where each controller is designed from its connected model (See figure 13). There are various possibilities while composing the network. One can use solely fixed or adaptive model-controller pairs or a combination of them. The authors argue that the best compromise is obtained by using a certain number of fixed models plus an adaptive and reinitialisable one. The best control performance and stability results have been obtained from this scheme.

The selection of the controllers is achieved according to the performance of their connected model. The network of models can therefore be interpreted as a gating system (See figure 14). This is a clustering free approach where at each instant, the selected controller i is the one having its connected model i minimising the index $J_i(t)$:

$$J_i(t) = \alpha \hat{e}_i^2(t) + \beta \int_0^t e^{-\lambda(t-\tau)} \hat{e}_i^2(\tau) d\tau \tag{12}$$

where $\hat{e}$ is the modelling error, $\alpha \geq 0$, $\beta > 0$ and $\lambda > 0$ are designed parameters. $\alpha \geq 0$ and $\beta > 0$ respectively influence the instantaneous and long term memory of the index. This index can be thought of a first order filter.

The properties of a simplified version of the MSM is investigated in this study: the "Model-Controller Network" (MCN). The index used takes only into consideration the integral of the error over the immediately preceding interval of T units. This gives

$$J_i(t) = \int_{t-1}^t \hat{e}_i \tag{13}$$

This index should effective enough in this study since we are not considering any system affected by disturbances and compared to the one used in the MSM it has the advantage of requiring no setting of parameters.

All the model-controller pairs are adaptive but with no forgetting factor so that their adaptability will diminish with the increase of data reviewed. To have all controllers adaptive is not computationally demanding since only one controller is adapted each time (the one selected). However the number of controllers will significantly affect the computing time as at each instant all the models must be activated to determine their performance. This actually

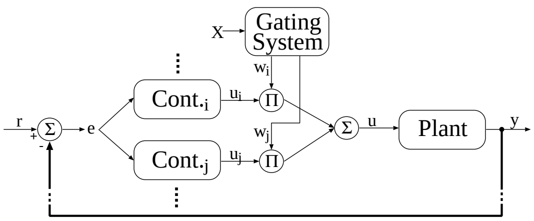

Figure 13: The controller network.
A network of feedback controllers. A gating system is used to select only one controller at each instant. Hence, the output of the network is always reduced to the output of a single controller.

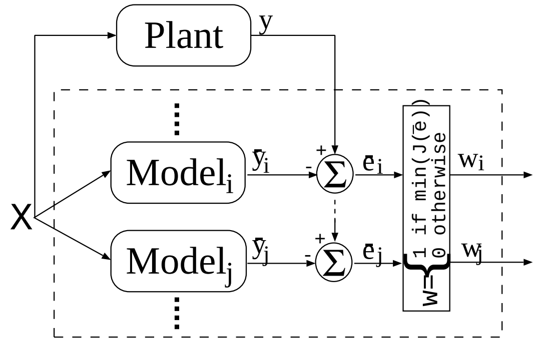

Figure 14: The gating system of the multiple switched models.
It is composed of plant models each one connected to a controller. The weight w_i of a model i equal 1 if $min(J_i)$ or 0 otherwise. Hence, this scheme implies the selection of only one controller at each instant

implies far more computations than the selection of the controller using the clustering approach described in the previous section.

Note that the "hysteresis" feature, introduced in (Middleton *et al.*, 1988) and use also in the MSM, is not involved in the MCN. The hysteresis is a time delay between the selection of a new controller and its activation. This is to avoid instability that could arise from a quick change of controllers. Since we are here only dealing with system having simple dynamics, this feature should be of no use in our case.

In (Narendra *et al.*, 1995; Narendra and Balakrishan, 1997) the parameters of the models are initialised at a certain distance from their desired value. Different preinitialisations enable the convergence of the model towards different solutions. However, a random initialisation of the model should not ensure that the scheme will converge toward a solution. Moreover, the number of controllers required to control an unknown system can not be determined a priori. To use a high number of controllers to overcome this problem is certainly not a satisfactory solution at least because of the high number of computations that it would implies. Hence, here again the overall modelling and control of the plant should be significantly facilitated by an incremental construction of the network of model-control pairs.

The general idea of the INC can be applied to the MCN. That is to say that the MCN can be constructed according to the on-line control performance of the controller network. Actually, the incremental process is simplified since only the controller attached to the model performing the best according to the index (13) is selected at each instant rather than the two neighbours of the operating condition in the CCN. However this method can only be used for control purposes since the modelling performance of each model can only be determine a posteriori. There are two possible actions according to the actual control error $\bar{e}$:

- *If* $\bar{e} \leq threshold$:
  The selected controller is updated in order to integrate the actual system input-output sample and to be adapted to small changes in the environment.

- *If* $\bar{e} > threshold$:

A sampling of the system is achieved around the actual operating condition (this is usually preferable instead of continuing the control which can become totally inaccurate if the change of parameters relating the operating condition is abrupt e.g. case of a discontinuous function). From those samples a local model of the system is determined. A linear controller is then designed from this model. Similar to the CCN, a Model Reference Adaptive Controller design method is used in this study (see (Ronco, 1997) (chapter 2) for details about this control design method). The new model-controller pair is finally added to the MCN.

Similar to the CCN, there is in the INC a pruning of the controllers that are not performing well enough i.e. their average control error is greater than twice the threshold. Indeed this should not occur very often since a controller is added as soon as the control error exceeds the threshold. However it can happen that a controller has not learned enough (i.e. the controller has not reviewed enough system input-output samples). A controller must be selected to learn. The fact that the controller is almost never selected means that it is of no use for the network. It may be useless because it is not adapted well enough. Otherwise its connected model should have performed well at some operating conditions and this would have implied the selection of this controller. Hence, this is a simple but very effective way of determining what are the undesirable controllers in the network. Note that, without any mean of biological mimetism, this method is applied by biological systems to prune neurons. Neurons with low activity tend to degenerate (or at least to be allocated to other tasks).

As well as with the CCN, it is suggested to apply the INC only during a learning stage. After a while, when the system has been driven in most of its possible states, the INC should be removed. It is not so much to speed up the process but to ensure a fixed structure of the controller network necessary to be implemented to control real systems. By removing the INC one makes the MCN entering a generalisation stage.

Note also that a similar technique than the INC is reported in (Narendra *et al.*, 1995). However, their method contains no pruning feature. This pruning should be vital to remove inadapted controllers that are very likely to imply unstable behaviours in the MSM if they come to be activated due to disturbances.

5.1 Illustration

The control capability of the IMCN is going to be illustrated according to the non-linear system (1) used previously to illustrate the control capability of the ICCN. Hence the INC is applied to the MSM during the same learning stage than the one used for the creation of the CCN. As a result six controllers were designed (instead of five in case of the ICCN). Each of the first order linear controllers composing the MCN are described by the following equation that has been derived in section 2:

$$u = \frac{1}{b}(1.5351r - y(1.5351 - a)) \tag{14}$$

where b and a are the two parameters of a first order local model of the system (1).

The results obtained from the IMCN during the same generalisation stage as for the ICCN are depicted in figure 15. In the bottom subplot of this figure you can see the six local linearisations of the system obtained during the INC that were used for the indirect design of the controllers. These linearisations very accurately cover the system's non-linearity. This implies that the overall model developed by the IMCN is very accurate. This is why the overall controller performs so well. As for the CCN, the transient of the system (dashed line) matches perfectly the desired transient (plain line).

Hence, here again we see that the INC lead to a perfect network construction that involved perfect control of the system.

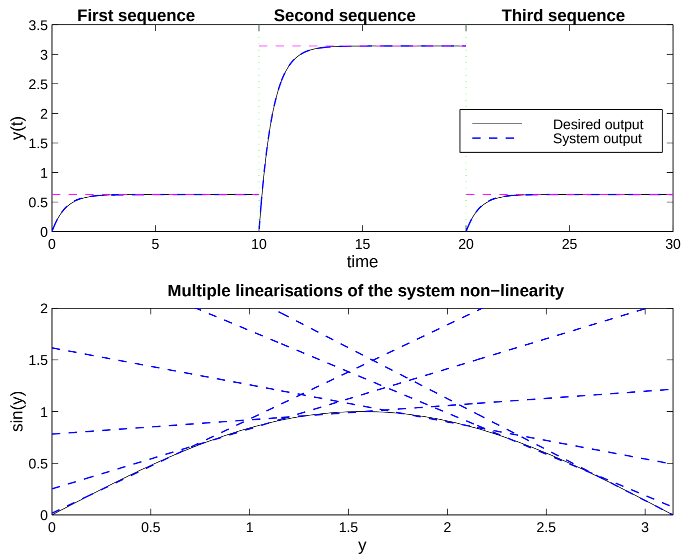

Figure 15: Performance of the IMCN whilst controlling system (1).
The top subplot depicts the control performance of this scheme concerning three consecutive control sequences. During the first, second and third control sequences a desired transient (see plain line graphs) has to be achieved whilst respectively driving the system (see dotted line graphs) toward the desired position $y = \frac{\pi}{5} = 0.63$, $y = \pi$ and $y = \frac{\pi}{5} = 0.63$. Six controllers are active for various operating regions. The bottom subplot shows the system's non-linearity in the range $0 \leq y \leq \pi$ and the six local linearisations of the system used to design the controllers (see plain straight lines).

5.2 Concluding remarks

A simplified version of the MSM extensively studied in (Narendra and Balakrishan, 1997; Narendra *et al.*, 1995) has been described in this section: the "Model-Controller Network" (MCN). The modelling performance index used to select the controllers is much simpler than the one used in the MSM. Another difference is that in the MCN all the model-controller pairs are adaptive. Referring to the stability results obtained and described in (Narendra and Balakrishan, 1997), an important advantage of this scheme is the insurance of stability. However one must be careful with this result. The robustness of the multiple controllers scheme should be of prime interest since few inadapted but stable controllers could lead to an unstable overall behaviour of the network. This is why a pruning of inconsistent approximations (e.g. model developed from singular matrices) is necessary to avoid the above problem.

Note that the MSM lacks the possibility of interpolating between controller outputs. This is essential to smooth the behaviour of the network. This possibility has not been investigated. This should be a future area of research.

We have seen that the INC is a very efficient technique to construct the MCN architecture. The results obtained by this scheme were similar to the one obtained by the ICCN. Since the MCN avoids the clustering problem of the CCN without adding significant drawbacks we could argue that for control purposes the IMCN is a better solution than the ICCN. However to discuss this further, in the next section will be compared these two approaches on a couple of different control problems.

6 Illustration

So far a quite simple non-linear control problem has been used to illustrate the behaviour of both the ICCN and the IMCN. The system (1) was first order and stable. The illustration is going to be extended next by considering a second order unstable system: the inverted pendulum. In the following section will be illustrated the problem of using a CCN to control a system where more than one variable implies non-linearity into the system.

6.1 Control of a second order unstable non-linear system

In this section we are going to consider the control of the inverted pendulum (see figure 10as an example of a second order unstable non-linear system. The equation of the inverted pendulum is the following:

$$\ddot{\theta} = \frac{\frac{1}{2}mgl.\sin\left(\theta\left(t-1\right)\right)+\tau}{\frac{1}{3}ml^2} \tag{15}$$

where $\ddot{\theta}$ is the angular acceleration of the pendulum and θ is the angle of the pendulum. τ is the torque in Newton meters applied to the pendulum by the controller. The mass of the pendulum is $m = 0.1kg$, $l = 0.5m$ is its length and $g = 0.81$ is the gravity. The Euler method is chosen for the integration.

This problem is trivial if the control is performed whilst the pendulum angle $\theta \leq \pm 12^o$ since in this region the pendulum behaves linearly. It becomes much more difficult to control the pendulum as θ is diverging from this region. The task of the controller will be to stabilise the pendulum at three different angles that are respectively $\theta = \frac{1}{4}\pi = 0.78rads$, $\theta = \frac{1}{2}\pi = 1.57rads$ and $\theta = \frac{3}{4}\pi = 2.35rads$. These three pendulum positions are depicted in figure 16. These three control sequences last 10sec each.

For each control sequence, the controllers are required to track a transient characterised by no overshoot, a settling time of 5sec and a tolerance error of 1% of the set point r. This specification gives rise to the following the 2nd order linear adaptive controllers (i.e. Model Reference Adaptive Controller):

$$u = \frac{1}{b_2}\left(-b_1 u + \theta_1^r sr - (\theta_1^r - a_1)sy + \theta_2^r r - (\theta_2^r - a_2)y + c\right) \tag{16}$$

where b_1, b_2, a_1 , a_2 and c are the parameters of a second order local model of the system and $\theta_1^r = 1.5$ and $\theta_2^r = 1.84$ are the parameters of the reference closed loop model (see (Ronco, 1997) (chapter 2) for details).

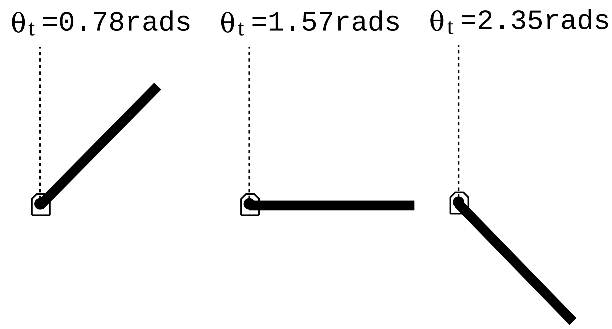

Figure 16: The three desired positions of the inverted pendulum

The same learning strategy as the one described earlier has been used to properly excite the plant. That is to say that the learning stage consisted in five control sequences with respective set points $r = 2.5 * 1/5$, $r = 2.5 * 2/5$, $r = 2.5 * 3/5$, $r = 2.5 * 4/5$ and $r = 2.5$. The same transients as during the generalisation stage (see above) have to be tracked during this stage.

At the end of the learning stage the INC had attributed five controllers to the ICCN and nine to the IMCN. The results obtained by a single adaptive controller, an ICCN and an IMCN are depicted in figure 17. It is clear that the two controller networks (see graphs IMCN and ICCN transient) perform much better than the single adaptive controller (see graph MRAC transient). The adaptive controller controls the system with a very slow transient and is suffering the stability-plasticity dilemma (otherwise it would have performed well during the second control sequence). The performance of the two controller networks are comparable. During the first two control sequences they meet the control requirement: the system settles down in about 5sec, there is no overshoot and the steady state error is less than 1% of the set point. During the third control sequence the actual transient is slightly slower than the desired one but the control requirement are almost meet.

The transient of the controlled input are plotted in figure 18 to show more clearly the behaviour of the controller networks. We can see that they do not always behave very smoothly. During the first control sequence the ICCN makes a sharp change of behaviour around $t = 5sec$. This is a very undesirable behaviour which could have easily yielded instability. This suggests that a controller is inadapted around the region $y = 2.35$. By observing the operating region of the controllers (see table 1) we see that the controller #5 is the one active for this region. One could easily investigate the stability and robustness of this controller. Depending on the results one could have decided to remove this controller. It was not removed because it was assumed that this behaviour was due to an inadaptation of this controller (a significant change between the parameters of its connected model occurred after this controller became activated). The ICCN was reapplied to the control problem to validate this hypothesis. The results are depicted in figure 19. The undesirable sharp behaviour has disappeared. The control is more accurate.

This example was only meant to show how simple it is to understand the behaviour of the ICCN since each controller has a clear region of activity defined by its attached rbf. This is unfortunately not the case of the IMCN. We can not know easily the activity region of each controller. If the IMCN behaves undesirably at a certain operating region it is difficult to know which controller is responsible. For instance, during the last control sequence, the IMCN

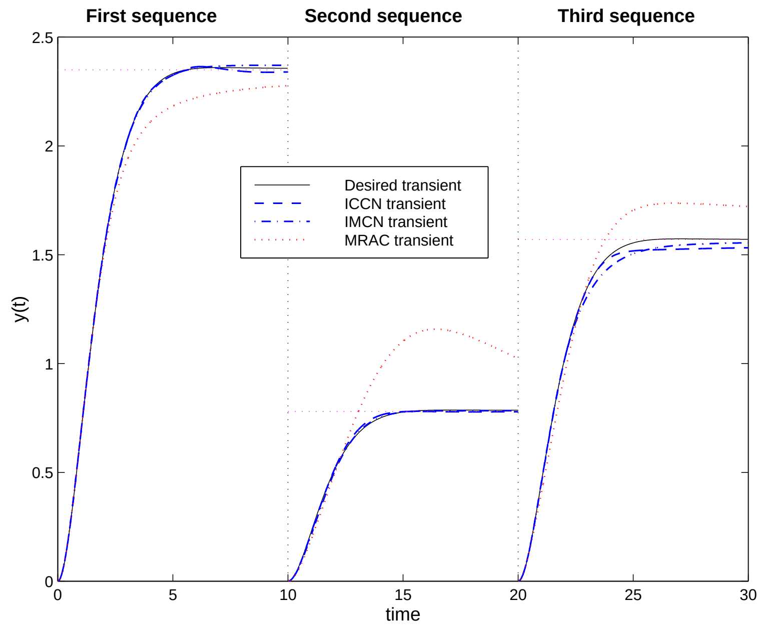

Figure 17: Control performance of a single adaptive controller, an ICCN and an IMCN whilst controlling the inverted pendulum.
Three different control sequences of 10sec each had to be achieved. During the first, second and third control sequences a desired transient (see plain line graphs) has to be achieved whilst respectively driving the system (see respective dashed line graphs) toward the desired angular position $\theta = \frac{1}{4}\pi = 0.78rads$, $\theta = \frac{1}{2}\pi = 1.57rads$ and $\theta = \frac{3}{4}\pi = 2.35rads$.

Table 1: Operating region of the controllers composing the ICCN

Controller #	Operating region
1	$0 < \theta < 0.8201$
2	$0.8201 < \theta < 0.9562$
3	$0.9563 < \theta < 1.5500$
4	$1.5502 < \theta < 2.3528$
5	$2.3531 < \theta < 2.5445$

changes abruptly its behaviour approximatively at $t \approx 23sec$. At this time the operating region $y = \theta \approx 1.45$. This last information is irrelevant for the determination of the activated controller. To determine the controller responsible for this undesirable behaviour each controller has been associated to a rbf. The operating regions are given in table 2. Note that the region of activity of each controller is only determined during the generalisation stage.

First, one will notice that there are a lot of overlaps. More importantly there are some controllers (#4, #5 and #8) associated with a very small operating region. This could mean that there are of no use for the control problem. Regarding the controllers operating around the region $y = \theta \approx 1.45$, we can now see that they are two possible candidates: controller #3 and #6. The values of the model parameter b_1 associated with the control input u (which highly influences the control design) are depicted in the last row of table 2. We actually know from the pendulum equation (15) that this parameter is $\frac{1}{\frac{1}{3}ml^2} = 120.4819$. Most of this models parameter

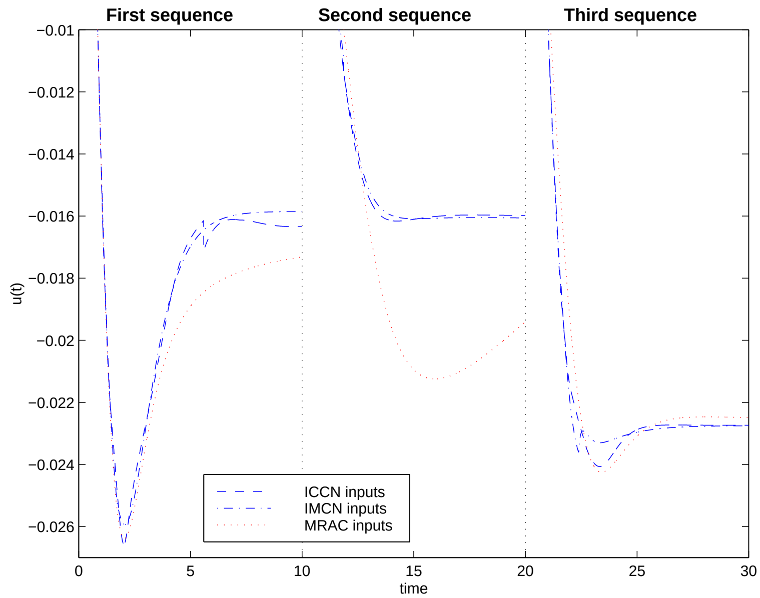

Figure 18: Transient controlled input of a single adaptive controller, an ICCN and an IMCN. These transients are associated to their control performances depicted in figure 17.

Table 2: Operating region of the controllers composing the IMCN

Controller #	Operating region	Size	b_1
1	$0 < \theta < 1.2952$	1.156692	113.0253
2	$0.4239 < \theta < 0.8026$	0.378670	116.3043
3	$0.1314 < \theta < 1.5548$	1.423441	98.1966
4	$0.2561 < \theta < 0.2593$	0.003202	129.0845
5	$0.2771 < \theta < 0.2853$	0.008208	118.6738
6	$0.7095 < \theta < 2.3705$	1.661001	134.7457
7	$0.4637 < \theta < 1.3316$	0.867870	116.4107
8	$0 < \theta < 0$	0	115.3445
9	$0.1176 < \theta < 1.0255$	0.907872	121.2787

value b_1 are fluctuating around 120.4819 except for the models attached to the controllers #3 and #6 which have respectively a parameter value $b_1 = 98.1966$ and $b_1 = 134.7457$. One should notice also that these two controllers have the two largest operating regions. This means those controllers are active at operating regions that are very different from each other and thus disrupt the adaptation of these controllers. This problem has occurred because during the generalisation stage no controller is added even if the network does not perform well. Instead of forcing the use of the two unfitted controllers #3 and #6 for the operating region around $y = \theta \approx 1.45$ a new controller should have been added. Although this inadaptation does not disrupt significantly the control, for other situations this could have lead to unstable behaviour of the IMCN. This suggests that one should make sure that most of the operating conditions have been reviewed before ending the INC and switching from the learning stage to the generalisation stage. Indeed, this recommendation holds also for the CCN.

In brief, the main point stressed by this control example is that the association of activity

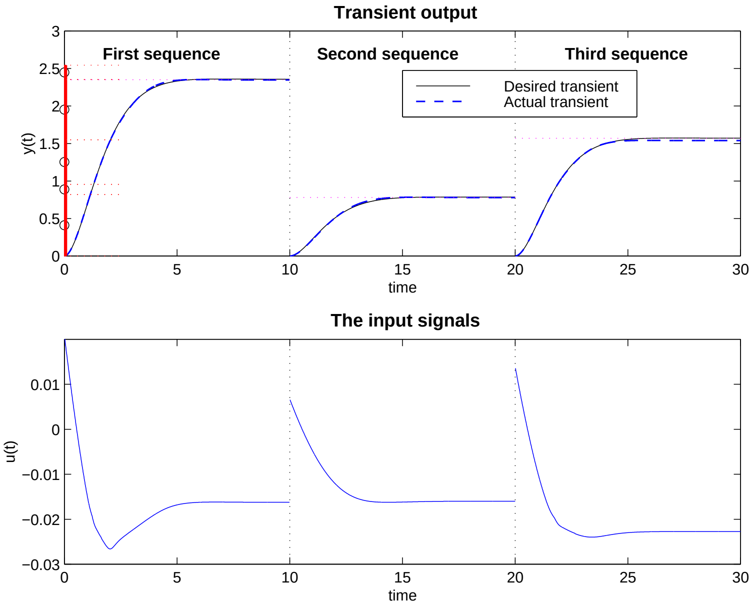

Figure 19: Control performance of the ICCN whilst controlling the inverted pendulum a second time.
The top plot depicts the system transients obtained whilst controlling the pendulum toward the desired angular position $\theta = \frac{1}{4}\pi = 0.78rads$, $\theta = \frac{1}{2}\pi = 1.57rads$ and $\theta = \frac{3}{4}\pi = 2.35rads$. The bottom plot depicts the transient controlled input.

regions to controllers (even if those regions are not used for the selection of the controllers) highly simplifies the understanding of the network behaviour. We have also seen that the ICCN and the IMCN perform equally well and much better than a single adaptive controller whilst controlling a second order system having a unique variable implying a non-linear behaviour. In the next section are presented some results obtained using the ICCN and the IMCN whilst controlling a second order system having two variables implying non-linear behaviours.

6.2 Control of a second order system having two states implying non-linear behaviours

This section illustrates the behaviour of the ICCN and IMCN whilst controlling a second order system having two states implying a non-linear behaviour. The arbitrary system used is described by the following second order equation:

$$s^2 y = 10u + 2\sin(y) + \cos(2sy) \tag{17}$$

You see from this equation that the system non-linearity not only depends on the system output y but also on the velocity of the system sy. This implies significantly different behaviour at the beginning of the control (fast velocity) and at the end of the control when the system is settling down.

The control problem consists of three different control sequences differing only on the set points that are respectively $r = \frac{1}{3}\pi = 1.0472$, $r = \frac{1}{2}\pi = 1.5708$ and $r = \pi$. The 2nd order linear controllers used here correspond to equation (16).

After a learning stage similar to the one described in the previous section, the INC allocated six controller to the ICCN and 10 to the IMCN. These numbers of controllers have a significant impact on the control performances of the ICCN and IMCN (see figure 20 and figure 21). The IMCN performs very well (perfect match of the IMCN transient with the Desired transient). This contrasts with the poor control performance of the ICCN (and of the single adaptive controller).

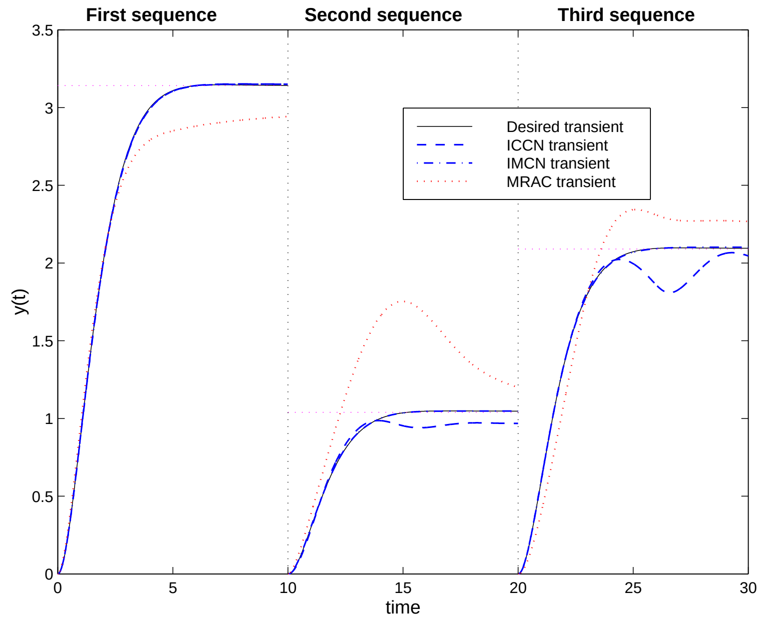

Figure 20: Control performance of a single adaptive controller, an ICCN and an IMCN whilst controlling system (17).
Three different control sequences of 10sec each had to be achieved. During the first, second and third control sequences a desired transient (see plain line graphs) had to be followed whilst respectively driving the system (see respective dashed line graphs) toward the desired position $y = \frac{1}{3}\pi = 1.0472$, $y = \frac{1}{2}\pi = 1.5708$ and $y = \pi$.

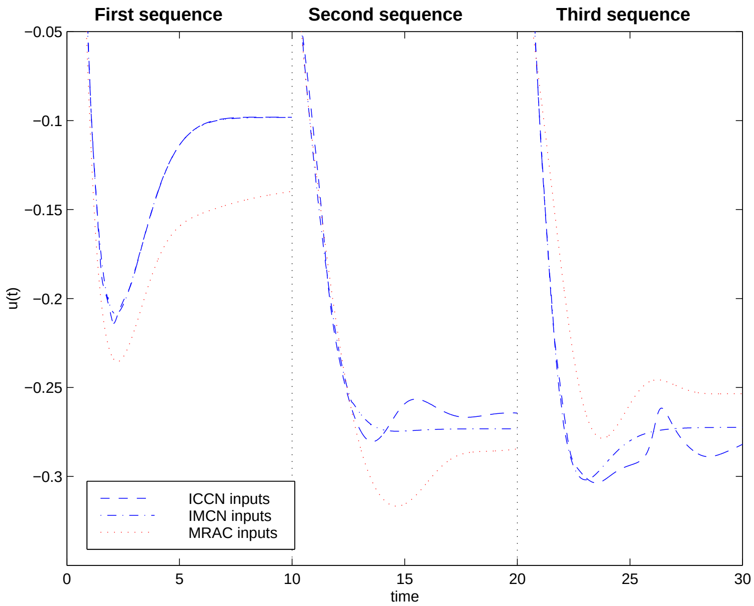

Figure 21: Transient controlled input of a single adaptive controller, an ICCN and an IMCN associated to their control performances depicted in figure 20.

These results were expected since the ICCN clustered on a single dimension (i.e. the system output) which is in this case not the only one involving a non-linear behaviour in the system. At the same operating region, the behaviour is significantly different depending on whether the system is accelerating (e.g. start of a control sequence) or settling down (end of a control sequence). The ICCN performs well to control the system with the set point $r = \pi$ because it encountered situations around $y = \pi$ involving a settling down of the system but no situation involving its acceleration. During the other control sequences there are operating regions where both situations are encountered and a unique controller has to cope with them. Note that the results are improved if the same ICCN is controlling for a second time system (17) but they are still inaccurate (see figure 22).

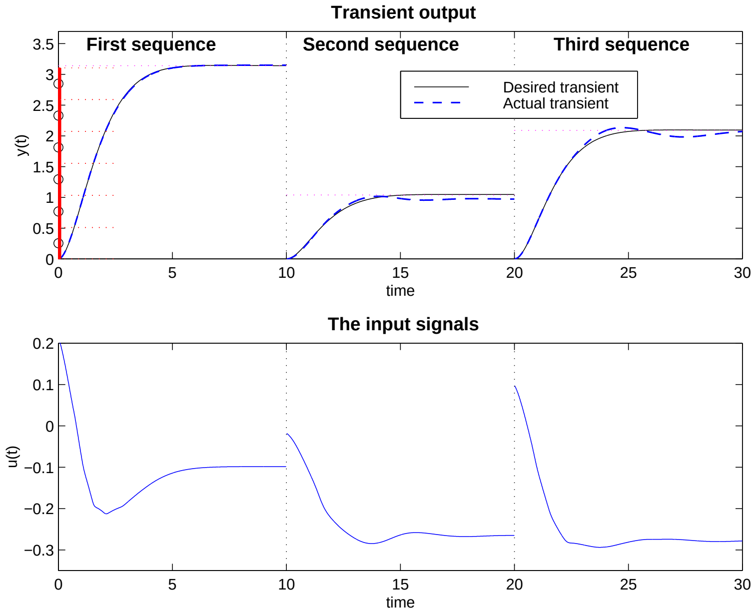

Figure 22: Control performance obtained whilst controlling system (17) a second time. The top plot depicts the system transients obtained whilst controlling the system toward the desired position $y = \frac{1}{3}\pi = 1.0472$, $y = \frac{1}{2}\pi = 1.5708$ and $y = \pi$. The bottom plot depicted the transient controlled input.

The operating regions of each of the controllers composing the ICCN are summarised in table 3. These operating regions differ totally from the ones involved by the IMCN (see table 2). Most of the controllers of the IMCN have a large operating region that is overlapping with many others. This means that there are many controllers involved in the same operating condition y. A two dimensional clustering should clarify the behaviour of the IMCN. This was done by clustering the controllers according to the system output y and its velocity sy (see figure 23). This way, each controller appears to have a square region of activity over the operating space. The fact that we can not identify clearly what is the operating region of each controller indicates that a highly complex clustering (certainly not square) is taking place especially in the region $y[0 \quad 1]$ $sy[0 \quad 0.6]$ where more than half of the controllers are overlapping. This is the only way to cope with the highly non-linear behaviour of system (17). This complex network behaviour would be difficult to obtain using a spatial clustering approach for the selection of the controllers. It is clear from these results that the IMCN is a much more suitable approach than the ICCN for the control of highly non-linear systems.

Table 3: Operating region of the controllers composing the ICCN

Controller #	Operating region	Size
1	$-0.000001 < y < 0.510308$	0.510309
2	$0.510318 < y < 1.034262$	0.523944
3	$1.034274 < y < 1.554180$	0.519906
4	$1.554212 < y < 2.072184$	0.517972
5	$2.072217 < y < 2.589818$	0.517601
6	$2.590953 < y < 3.106402$	0.515449

Table 4: Operating region of the controllers composing the IMCN

Controller #	Operating region	Size
1	$0.000616 < y < 0.600437$	0.599821
2	$0.000134 < y < 0.853710$	0.853576
3	$0.031017 < y < 1.047788$	1.016772
4	$0.002177 < y < 0.923805$	0.921628
5	$0.000000 < y < 1.411980$	1.411980
6	$0.322573 < y < 2.102542$	1.779969
7	$0.090697 < y < 2.008967$	1.918270
8	$0.276597 < y < 1.036315$	0.759718
9	$0.606431 < y < 1.832973$	1.226541
10	$1.838017 < y < 3.151625$	1.313608

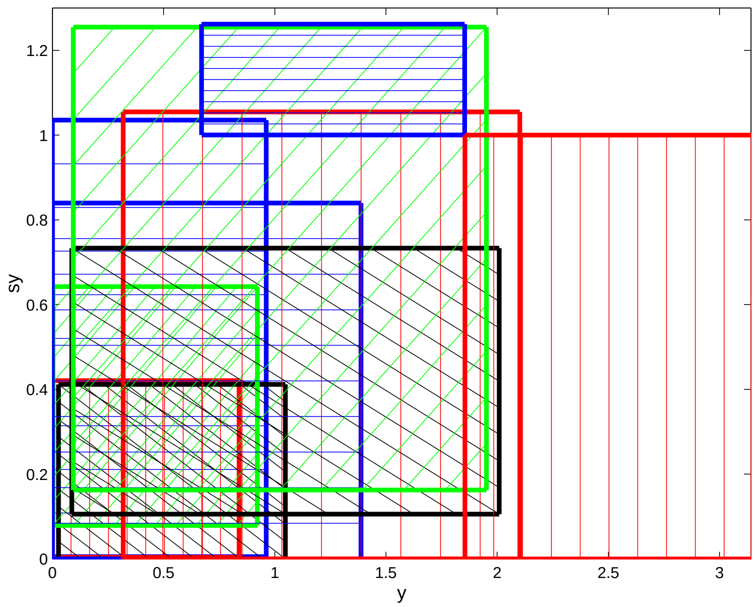

Figure 23: Two dimensional operating regions of the controllers composing the IMCN

7 Conclusion

In this study was presented two controller networks which have in common a gating system enabling them to select at each instant the currently valid local controllers of the plant. The local controllers composing these gated controller networks are Model Reference Adaptive Controllers (MRAC). A detailed description of such a controller and its properties can be found in (Ronco, 1997) (chapter 2). Note that other kind of local controllers might be used as building block of the controller networks.

One of the gated controller networks uses a spatial clustering approach to select the controllers at each instant: the "Clustered Controller Network" (CCN) that shares many common features with the "Local Controller Network" (LCN) developed by (Johansen and Foss, 1992; Johansen and Foss, 1993). The main difference in the CCN is that, whatever the system characteristics, the clustering used for the selection of the controllers is achieved on a single quantity. This simplification is justified by several important advantages. The main advantages are the simplification and speed-up of the controllers selection, the facilitation of the local plant approximations, the clarification of the overall behaviour of the CCN and the simplification of the construction of the network architecture. The latter advantage enabled us to develop a fairly simple algorithm to automatically construct the network architecture of the CCN: the "Incremental Network Construction" (INC). The CCN together with the ICN lead to the "Incremental Clustered Controller Network" (ICCN).

The other gated controller network is clustering free in the sense that instead of clustering the operating space for the selection of the controllers it uses the modelling performance of the models attached to each of the controllers to achieve the controller selection. This approach has been called the "Model-Controller Network" (MCN). The MCN is similar to the so called "Multiple Switched Models" (Narendra *et al.*, 1995; Narendra and Balakrishan, 1997). The ICN, used for the construction of the CCN, has been adapted to construct the MCN. From the ICN and the MCN arises the "Incremental Model-Controller Network" (IMCN).

An important feature of the ICN in both the CCN and MCN is the pruning of inadapted controllers. Fairly simple heuristics are used to determine whether a controller is well adapted or not. The average control error is one of the criteria. A more sensible criteria is the number of input-output samples reviewed by a controller which, particularly in the case of the MCN, indicates the inadaptation (or at least useless) of a controller. However, the singularity of the matrix should be a much more relevant criteria to determine the adaptation quality of a controller. This information is contained in the singular vector given by the SVD whilst approximating the parameters of the models associated with the controllers (see (Ronco, 1997) (chapter 2) for details).

Both the ICCN and the IMCN offer several important advantages compared to other approaches. Since the controllers composing these networks are linear, it should be simple to analyse locally the properties of each controller and deduce general properties regarding the overall behaviour of the network. For the same reason it is straightforward to determine the parameter values of the controllers as linear regression methods can be applied. These two very important advantages are missing in most of the other non-linear control schemes. Another advantage is the possibility for these controller networks to be adapted to the full operating range of conditions involved by a non-linear system. This is why these schemes are not sensitive to the stability-plasticity dilemma that largely affects a single adaptive controller. A last important advantage is that the INC makes the CCN and the MCN two self-organising approaches. The only system knowledge required to apply the ICCN and the IMCN is the system order. These advantages highlight the high potential of the controller networks for the control of non-linear systems.

Different examples of control of non-linear systems have been considered in order to illustrate the behaviour of the ICCN and the IMCN. It is very clear that both these schemes are performing much better than a single adaptive controller. Perfect matching of desired transients have been obtained whilst controlling highly non-linear systems. Moreover, the INC allocated an optimum number of controllers to the two controller networks. We have also seen that the understanding of the networks behaviour was straightforward due to the knowledge of the region of activity of each controller.

However the ICCN appears to have serious problems in coping with non-linear systems having more than a single variable implying a non-linear behaviour. The IMCN does not have such a problem. This high sensitivity to the clustering space order is the main drawback limiting the use of the ICCN or any other approaches involving an operating space clustering for the controllers selection.

This does not imply to reject the ICCN. It can be useful for situations where the IMCN can not be applied. It is clear that the controllers composing the IMCN must be designed from a local system model. Without a model one cannot select the controllers and therefore cannot use the IMCN. For instance it may be difficult to apply this approach using model based predictive controllers as building blocks of the IMCN. This is inconvenient since such a controller is a powerful approach to handle systems with complex dynamics. In addition, it is not possible to use the IMCN for modelling purposes. Hence, the ICCN has a wide range of applicability over the IMCN.

Moreover, the spatial clustering itself has the important advantage of clarifying the understanding of the controller networks' behaviour. This is why rbfs have been attached to the controllers composing the CCN as well as for the MCN. In the MCN the rbfs are not used for the selection of the controllers as in the former case but are simply used to facilitate the understanding of its behaviour. A powerful approach would consist of combining these two gating schemes (the spatial clustering and the clustering free approach). The number of controllers in a MCN could be rather large depending of the non-linearity and the order of the system. In such a case, for the controller selection, the activation at each instant of all the models could involve too many computations to be feasible. One way to overcome this problem would be to activate only the models which have an operating region covering the actual operating condition (the clustering on a single quantity would be satisfactory in this case). This constitutes one of our future works in this area.

References

Carpenter, G. A. and S. Grossberg (1988). The art of adaptive pattern recognition by a self-organising neural network. *IEEE Computer* **21**(3), 77–88.

Jacobs, R.A. and M.I. Jordan (1993). Learning piecewise control strategies in a modular neural network architecture. *IEEE Transaction on Systems, Man, and Cybernetics* **23**(2), 337–345.

Johansen, T. A. and B. A. Foss (1992). A narmax model representation for adaptive control based on local model. *Modeling, Identification, and control* **13**(1), 25–39.

Johansen, T. A. and B. A. Foss (1993). Constructing NARMAX models using ARMAX models. *Int. J. Control* **58**, 1125–1153.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics* **43**, 59–69.

Middleton, R. H., G. C. Goodwin, D. J. Hill and D. Q. Mayne (1988). Design issues in adaptive control. *IEEE Transaction on Automatic Control* **33**(1), 50–58.

Morse, A. S. (1990). Toward a unified theory of parameter adaptive control–tunability. *IEEE Transaction on Automatic Control* **35**(9), 1002–1012.

Morse, A. S., D. Q. Mayne and G. C. Goodwin (1992). Applications of hysteresis switching in parameter adaptive control. *IEEE Transaction on Automatic Control* **37**(9), 1343–1354.

Narendra, K. S. and J. Balakrishan (1997). Adaptive control using multiple models. *IEEE transactions on Automatic Control* **42**(2), 171–187.

Narendra, Kumpati S., Jeyendran Balakrishnan and Kemal M. Ciliz (1995). Adaptation and learning using multiple models, switching, and tuning. *IEEE Control Systems* **3**, 37–51.

Poggio, T. and F. Girosi (1990). Networks for approximation and learning. *Proceedings of the IEEE* **78**, 1481–1497.

Ronco, Eric (1994). Apprentissage a complexite progressive dans les systemes connexionnistes. Master's thesis. Institut National Polytechnique de Grenoble. Grenoble, France.

Ronco, Eric (1997). Incremental Polynomial Controller Networks: two self-organising non-linear controllers. PhD thesis. Faculty of Mechanical Engineering, University of Glasgow.

Ronco, Eric and Peter J. Gawthrop (1997). Incremental model reference adaptive polynomial controllers network. In: *Proceeding of the IEEE Conference on Decision and Control (CDC'97).* (In Press).

Ronco, Eric and Peter J. Gawthrop (submitted*a*). Incremental model reference adaptive polynomial controllers network. *International Journal of Systems Science.*

Ronco, Eric and Peter J. Gawthrop (submitted*b*). Polynomial models network for system modelling and control. *Neural Computing Survey*.

Ronco, Eric, Henrik Gollee and Peter J. Gawthrop (1996*a*). Modular neural network and self decomposition. Technical Report CSC-96012. Centre for system and control, U. of glasgow, UK. Available at www.mech.gla.ac.uk/~ericr/pub/mnnsd.ps.Z.

Ronco, Eric, Peter J. Gawthrop and Mohamed Abderrahim (1996*b*). Progressive local control. In: *World Automatic Conference*. pp. 637–642.

Ronco, Eric, Peter J. Gawthrop and Yasmine Mather (1996*c*). Incremental modular controllers network. In: *Proceeding of the International Conference on Intelligent and Cognitive Systems (ICICS'96)*.

Szilas, N. and E. Ronco (1995). Action for learning in non-symbolic systems. In: *Europeen Conference on Cognitive Science*.

Weller, S. R. and G. C. Goodwin (1994). Hysteresis switching adaptive control of linear multivariable systems. *IEEE Transaction on Automatic Control* **39**(7), 1360–1375.