Nonlinear Control Strategies Incorporating Input-State-Output Models

Konstantinos Dimopoulos

City Liberal Studies

Affiliated Institution of the University of Sheffield

Thessaloniki 546 24, Greece

k.dimopoulos@city.academic.gr

***Abstract* - Dynamic recurrent neural networks have become popular for use in non-linear control due to recent developments concerning their use in control strategies such as the Internal Model Control (IMC) and Global Linearisation Control (GLC). In both strategies a neural network is used to model the non-linear plant, and from this model a control rule is formulated so that the cascaded system of plant-model-controller becomes linear in behaviour. Investigators have referred to IMC as an input-output approach and GLC as an input-state-output approach. It has been speculated that there exist and equivalence in these two strategies.**

In this paper we examine both control strategies and demonstrate the conditions under which the two are equivalent. The conditions are tested in simulations by implementing the two strategies to control a simple non-linear plant, using first an exact model of the plant, then a model that contains some errors and finally with a dynamic recurrent neural network that has been trained as a model of the plant.

***Index Terms* - Internal Model Control, Globally Linearising Control, Dynamic Recurrent Neural Networks, Non-linear systems, Differential Geometry.**

I. Internal Model Control

Internal Model Control (IMC) first appeared in the 80's as a result of the dissatisfaction of the then available control system design methods and their ability to effectively deal with process control applications, and of the availability of increased-powered computer hardware. The basic IMC structure was extensively examined as a linear control strategy by Garcia and Morari in 1982 [1], and further as a non-linear controller in [2].

A. IMC Formulation

Consider the classic control structure of Fig. 1(a). In this structure the error due to feedback is calculated as:

$$e = y_{\text{ref}} - y_{\text{plant}} = y_{\text{ref}} + y_{\text{model}} - y_{\text{plant}} - y_{\text{model}} \quad (1)$$

Equation (1) transforms our system of **Fig. 1**(a) to an equivalent form illustrated in **Fig. 1**(b). By setting as a new controller $G_c = \dfrac{C}{1 + C \cdot G_m}$, the IMC structure is formulated. Because the block diagrams illustrated in **Fig. 1** are equivalent, the advantages of the transformation are not apparent at a first glance. In reality according to [1], the controller of the form of G_c is much easier to design than C, and the IMC structure allows for robustness to be included in a very explicit manner.

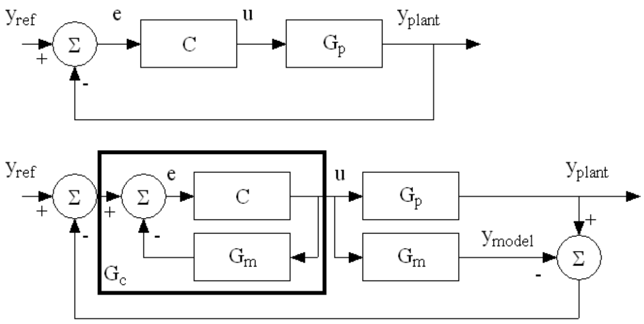

Fig. 1 (a) Classical control system, (b) IMC structure

In order for the IMC structure to work though, it is necessary for certain conditions to hold [1]:

1) The plant model is an exact match to the plant.

2) The inverse of the model exists and it is stable.

If these conditions hold, then the IMC structure has three very important properties:

1) *Dual Stability Criterion:* when the model is exact then the stability of both controller and plant is sufficient to guarantee overall system stability.

2) *Perfect Control:* when the model is exact, and the controller is the inverse of the model, then the control is perfect and the plant output is equal to the reference signal.

3) *Zero offset:* if the model is exact, the controller is the inverse of the model and the overall system is stable, then there will be no offset in the plant output if the reference signal is asymptotically constant.

In IMC the controller G_c is formulated in two steps. The first involves the construction of the inverse of the model. In the second step, an additional linear filter is incorporated to provide the system with robustness in the face of possible modelling errors, by reducing the loop gain. Additionally, it serves the purpose of transferring the error signal to the appropriate input range for the inverse.

In literature [2] three ways have been proposed for formulating the inverse. First it is possible to use a different network to act as the inverse of the original network model [3]. This procedure though involves additional training, and furthermore it does not guarantee that the inverse will completely cancel out the plant/model dynamics, as it is possible for modelling errors to exist in the inverse network, accumulating to the possible modelling errors in the model.

A second way of formulating an inverse is through numerical computation methods. These methods can prove to be computationally expensive, and difficult to implement in real time [1, 2].

Finally, it is possible to analytically design the inverse from the model using the Hirschorn inverse. Although this method is more straightforward and easy to implement than training an inverse network and using numerical computation, it has inherent problems of stability and is not easy to implement in real systems [4, 5]. This problem can possibly be overcome, by considering an input-state-output approach in designing and implementing the IMC. Besides the inherent problems involved, we will be using the Hirschorn inverse to formulate the IMC controller.

B. The Hirschorn Inverse

Consider the following control affine non-linear system:

$$\left.\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned}\right\} \tag{2}$$

Assuming that this system has relative order *r*, then according to the definition of relative order [6]:

$$\begin{aligned} & y(t) = h(x(t)) \\ & y^{(1)}(t) = L_f h(x(t)) \\ & \quad \vdots \\ & y^{(k)}(t) = L_f^k h(x(t)), \quad \forall k < r \\ & \quad \vdots \\ & y^{(r)}(t_0) = L_f^r h(x^0) + L_g L_f^{r-1} h(x^0) u(t_0) \end{aligned} \tag{3}$$

where $y^{(k)}(t)$ is the k^{th} derivative of $y(t)$, $L_f h(x)$ is the Lie derivative of $h(x)$ in the direction of $f(x)$. The left inverse problem is to analytically formulate a system that when fed with $y(t)$ in its input, it will produce $u(t)$ at its output (i.e. the input of (2) for which y is produced). Hirschorn [7] has showed that the left inverse of (2), if it exists, it is given by the following ordinary differential equations:

$$\left.\begin{aligned} \dot{z} &= F(z) + G(z)\upsilon \\ w &= H(z) + K(z)\upsilon \end{aligned}\right\} \tag{4}$$

where $K(z) = \dfrac{1}{L_g L_f^{r-1} h(z)}$, $H(z) = -L_f^r h(z) \cdot K(z)$, $G(z) = K(z) \cdot g(z)$ and $F(z) = f(z) + H(z) \cdot g(z)$.

It is important to notice that the plant state vector x of (2) and the inverse state vector z of (4) have the same dimensions.

Let us consider the left inverse in (4). Assuming that the plant and inverse state vectors are equal, then working out the substitutions we get:

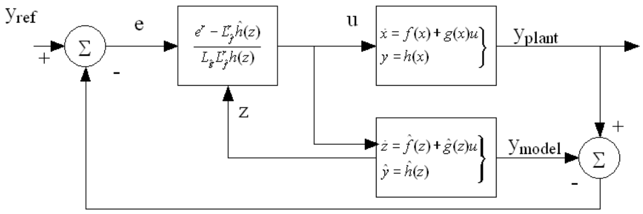

Fig. 2 IMC with Hirschorn controller

$$\left.\begin{aligned} \dot{x} &= f(x) + g(x)\frac{\upsilon - L_f^r f(x)}{L_g L_f^{r-1} h(x)} \\ w &= \frac{\upsilon - L_f^r f(x)}{L_g L_f^{r-1} h(x)} \end{aligned}\right\}$$

Let the input to the inverse be the r^{th} derivative of the output of (2), where r is the relative order of the plant. Then according to (3):

$$\upsilon = y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x) \cdot u \tag{5}$$

which transforms (4) to:

$$\left.\begin{aligned} \dot{x} &= f(x) + g(x)u \\ w &= u \end{aligned}\right\}$$

Therefore the left inverse of a system, is the same system but with the input given by:

$$u = \frac{y_{ref}^{(r)} - L_f^r f(x)}{L_g L_f^{r-1} h(x)} \tag{6}$$

Alternatively, it is possible to say that the inverting input is given by (6), with the states being fed-back from the linearised model or plant. This structure is shown in Fig. 2. Again for the IMC to work we have assumed that the plant states equal the model states at any time. This is an appropriate assumption according to the *dual stability criterion* as it also insures system stability. Assuming that the structure of Fig. 2 holds, and that the *perfect control criterion* is true, then in Fig. 2, $x=z$, $f(x) = \hat{f}(x)$, $g(x) = \hat{g}(x)$ and $h(x) = \hat{h}(x)$. The feedback will always be zero. Thus the error e will be equal to the reference signal, and therefore $e^{(r)} = y_{ref}^{(r)}$. Then because of (3) and (6) we get that the r^{th} derivative of the plant output is given by:

$$\begin{aligned} & y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x) \frac{y_{ref}^{(r)} - L_f^r f(x)}{L_g L_f^{r-1} h(x)} \Rightarrow y^{(r)} = y_{ref}^{(r)} \Rightarrow \\ & s^r y = s^r y_{ref} - s^{r-1}(y_{ref}(0) - y(0)) - \ldots - (y_{ref}^{r-1}(0) - y^{r-1}(0)) \Rightarrow \\ & y = s^r \frac{1}{s^r} y_{ref} - s^{r-1} \frac{1}{s^r} a_1 - \ldots - \frac{1}{s^r} a_{r-1} \end{aligned}$$

where a_1 to a_{r-1} are constants that depend on initial conditions. Assuming appropriate initial conditions these can be set to 0. Here is where the problem of the Hirschorn inverse originates, since in the IMC structure there will be r zero-pole cancellations at the origin.

This would not be a problem if the model equals the plant exactly, but in reality the model will approximate the plant to some degree of accuracy and therefore some small errors will exist. Furthermore, truncation errors in the procedure can move the system towards instability since it is marginally stable. More importantly, the Hirschorn inverse requires the reference signal to be differentiated r times. While in theory this does not pose any problems, in real systems accurate differentiation is difficult.

Let us consider the case where some small errors in the model exist, as well as errors in calculating the derivatives in the Hirschorn inverse. In this case, the model state vector will approximate the plant state vector. Hence we can define the error in the plant and model outputs as:

$$
\begin{aligned}
y-\hat{y} &= \varepsilon_o = h(x)-\hat{h}(z) \\
\dot{y}-\dot{\hat{y}} &= \varepsilon_1 = L_f h(x) - L_{\hat{f}}\hat{h}(z) \\
\vdots \quad & \quad \vdots \qquad \vdots \\
y^{(r-1)}-\hat{y}^{(r-1)} &= \varepsilon_{r-1} = L_f^{(r-1)}h(x) - L_{\hat{f}}^{(r-1)}\hat{h}(z) \\
y^{(r)}-\hat{y}^{(r-1)} &= \varepsilon_r = \\
&= L_f^r h(x) + L_g L_f^{(r-1)}h(x)\cdot u - L_{\hat{f}}^r\hat{h}(z) - L_{\hat{g}}L_{\hat{f}}^{(r-1)}\hat{h}(z)\cdot u = \\
&= \left[L_f^r h(x) - L_{\hat{f}}^r\hat{h}(z)\right] + \left[L_g L_f^{(r-1)}h(x) - L_{\hat{g}}L_{\hat{f}}^{(r-1)}\hat{h}(z)\right]\cdot u = \\
&= \qquad \varepsilon_{r_1} \qquad + \qquad \varepsilon_{r_2}\cdot u
\end{aligned}
\tag{7}
$$

Since $L_g L_f^{(r-1)}h(x) - L_{\hat{g}}L_{\hat{f}}^{(r-1)}\hat{h}(z) = \varepsilon_{r_2}$, which with some manipulation we can get

$$
\frac{L_g L_f^{(r-1)}h(x)}{L_{\hat{g}}L_{\hat{f}}^{(r-1)}\hat{h}(z)} = 1 + \frac{\varepsilon_{r_2}}{L_{\hat{g}}L_{\hat{f}}^{(r-1)}\hat{h}(z)}
\tag{8}
$$

Additionally because there are errors in calculating the derivative of y_{ref}, the Hirschorn inverse controller will be $\dfrac{y_{ref}^{(r)} - \varepsilon_o^{(r)} + \varepsilon_d - L_{\hat{f}}^r\hat{h}(z)}{L_{\hat{g}}L_{\hat{f}}^{(r-1)}\hat{h}(z)}$ where ε_d is the error of the derivative. Therefore the r^{th} derivative of the plant output will be:

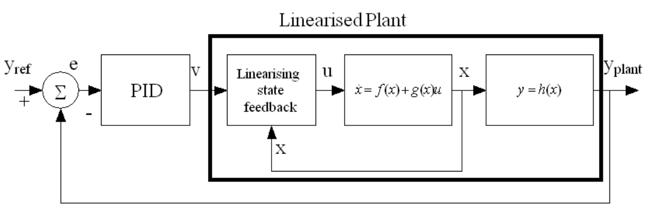

Fig. 3: The GLC structure

$$
\begin{aligned}
y^{(r)} &= L_f^r h(x) + L_g L_f^{(r-1)}h(x)\cdot\frac{y_{ref}^{(r)} - \varepsilon_o^{(r)} + \varepsilon_d - L_{\hat{f}}^r\hat{h}(z)}{L_{\hat{g}}L_{\hat{f}}^{(r-1)}\hat{h}(z)} = \\
&= L_f^r h(x) + \left[y_{ref}^{(r)} - \varepsilon_o^{(r)} + \varepsilon_d - L_{\hat{f}}^r\hat{h}(z)\right]\cdot\left[1 + \frac{\varepsilon_{r_2}}{L_{\hat{g}}L_{\hat{f}}^{(r-1)}\hat{h}(z)}\right] = \\
&= L_f^r h(x) + y_{ref}^{(r)} - \varepsilon_o^{(r)} + \varepsilon_d - L_{\hat{f}}^r\hat{h}(z) + \varepsilon_{r_2}\frac{y_{ref}^{(r)} - \varepsilon_o^{(r)} + \varepsilon_d - L_{\hat{f}}^r\hat{h}(z)}{L_{\hat{g}}L_{\hat{f}}^{(r-1)}\hat{h}(z)} = \\
&= L_f^r h(x) - L_{\hat{f}}^r\hat{h}(z) + y_{ref}^{(r)} - \varepsilon_o^{(r)} + \varepsilon_d + \varepsilon_{r_2}\cdot u = \\
&= \varepsilon_{r_1} + \varepsilon_{r_2}\cdot u + y_{ref}^{(r)} - \varepsilon_o^{(r)} + \varepsilon_d = \varepsilon_r + y_{ref}^{(r)} - \varepsilon_o^{(r)} + \varepsilon_d = \\
&= y_{ref}^{(r)} + \varepsilon_d \Rightarrow \\
s^r y &= s^r y_{ref} + \varepsilon_d \Rightarrow y = y_{ref} + \frac{1}{s^r}\varepsilon_d
\end{aligned}
$$

This is a very important result, since it shows that although the model is approximating the plant dynamics, the modelling errors are cancelled in the IMC procedure because of the feedback loop. It is important to note that these results are for models that approximate the dynamics correctly, and that the model correctly identifies the relative order. Should this not be the case, the above error cancellations would not be possible. Whatever the case, any errors in calculating the derivatives in the Hirschorn inverse controller, are integrated in the plant output.

By accepting that small model / plant mismatches are possible, we contradict the first criterion of the IMC. According to the *dual stability criterion* insuring that the model equals the plant insures stability of the strategy. In reality this is not possible, and therefore small errors in the model exist since the model will always be an approximation to the plant. The better the model approximates the plant, the smaller the errors. Although these errors will eventually cancel out due to the feedback in the strategy, the problem of internal stability still remains. Partially, this can be overcome with the selection of an appropriate linear filter that will be placed in front of the Hirschorn controller. Economou et.al. [2] provide guidelines for the design of this filter. One important factor is that it must have relative order equal to that of the plant. This conclusion is drawn from the fact that the filter will provide the strategy with desired linear dynamics that the non-linear plant will be linearised to. Finally, the filter will transfer the error signal to the appropriate input range for the inverse.

II. Globally Linearising Control

An alternative to IMC is Globally Linearising Control [4, 5, 8-11]. In this control strategy instead of using an inverted model to cancel out the plant dynamics, static state feedback is used in an internal feedback loop in order to linearise the plant to some desired linear dynamics. Then the resulting structure is placed inside a classic control loop where a linear controller is used to control the now linearised plant. This structure can be seen in Fig. 3. If the plant's state equations are known, the linearising state feedback law is given by:

$$u = \frac{v - L_f^r h(x) - a_{r-1} L_f^{r-1} h(x) - \ldots - a_0 h(x)}{L_g L_f^{r-1} h(x)} \tag{9}$$

In the case where they are not, a model is used to provide approximations. Therefore assuming that this is the case, the linearising structure takes the form of Fig. 4.

Let us assume that there exist same small errors in the model. Therefore $x \simeq z$, $f(x) \simeq \hat{f}(z)$, $g(x) \simeq \hat{g}(z)$ and $h(x) \simeq \hat{h}(z)$. In this case the feedback linearising law is given by

$$u = \frac{v - L_{\hat{f}}^r \hat{h}(z) - a_{r-1} L_{\hat{f}}^{r-1} \hat{h}(z) - \ldots - a_0 \hat{h}(z)}{L_{\hat{g}} L_{\hat{f}}^{r-1} \hat{h}(z)} \tag{10}$$

Combining the above equation with (3) the r^{th} derivative of the plant output can be calculated as:

$$y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x) \cdot \frac{v - L_{\hat{f}}^r \hat{h}(z) - a_{r-1} L_{\hat{f}}^{r-1} \hat{h}(z) - \ldots - a_0 \hat{h}(z)}{L_{\hat{g}} L_{\hat{f}}^{r-1} \hat{h}(z)}$$

Substituting (8) into the above yields:

$$\begin{aligned} y^{(r)} &= L_f^r h(x) + \left[1 + \frac{\varepsilon_{r_2}}{L_{\hat{g}} L_{\hat{f}}^{(r-1)} \hat{h}(z)} \right] \cdot \left[v - L_{\hat{f}}^r \hat{h}(z) - a_{r-1} L_{\hat{f}}^{r-1} \hat{h}(z) - \ldots - a_0 \hat{h}(z) \right] = \\ &= L_f^r h(x) + v - L_{\hat{f}}^r \hat{h}(z) - a_{r-1} L_{\hat{f}}^{r-1} \hat{h}(z) - \ldots - a_0 \hat{h}(z) + \varepsilon_{r_2} \cdot u = \\ &= \varepsilon_r + v - a_{r-1} L_{\hat{f}}^{r-1} \hat{h}(z) - \ldots - a_0 \hat{h}(z) \end{aligned}$$

Solving the equations in (7) for $\hat{y}$ and considering equation (3) we can transform the above to:

$$\begin{aligned} &\left. \begin{aligned} & y^{(r)} = \varepsilon_r + v - a_{r-1} \hat{y}^{(r-1)} - \ldots - a_0 \hat{y} \\ & \varepsilon_r = \varepsilon_0^{(r)} \\ & \hat{y} = \varepsilon_0 - y \end{aligned} \right\} \Rightarrow \\ & s^r y = s^r \varepsilon_0 + v - (a_{r-1} s^{r-1} + \ldots + a_0)(\varepsilon_0 - y) \Rightarrow \\ & (s^r + a_{r-1} s^{r-1} + \ldots + a_0) y = (s^r + a_{r-1} s^{r-1} + \ldots + a_0) \varepsilon_0 + v \Rightarrow \\ & y = \varepsilon_0 + \frac{1}{s^r + a_{r-1} s^{r-1} + \ldots + a_0} v \end{aligned} \tag{11}$$

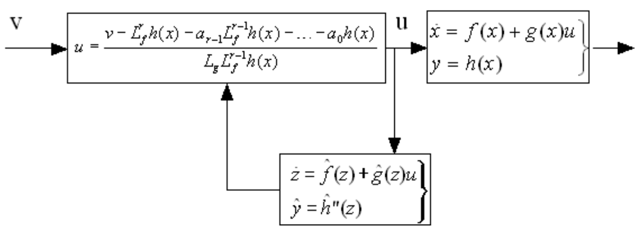

Fig. 4 GLC using a model as a state estimator

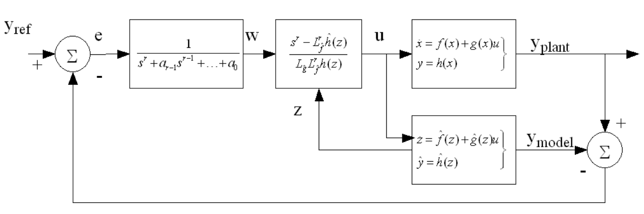

Fig. 5 IMC structure with a linear filter

Comparing the above result with that of when the model equals the plant, we see that the linearisation is the same but there will be an error term added. This is because there is no external feedback like in the case of the IMC to counter possible plant / model mismatches. Nevertheless, there are no zero-pole cancellations at the origin like in the case of the IMC, a fact which shows that GLC is more stable than IMC when there exist model / plant mismatches.

Let us consider the linearising feedback of the GLC described in (10). These can be re-arranged to:

$$\begin{aligned} u &= \frac{w - L_f^r h(x)}{L_g L_f^{r-1} h(x)} \\ w &= v - a_{r-1} L_f^{r-1} h(x) - \ldots - a_0 h(x) \end{aligned} \tag{12}$$

The control law described in (12) looks similar to that of the IMC. Comparing (12) with (6) we notice that the difference is in the way that the reference signal is manipulated. In IMC the reference signal is differentiated *r* times, while in GLC, the reference signal is the output of a linear system whose dynamics the non-linear plant will be linearised to. This is similar to the effect of the filter in the IMC that Economou describes in his paper [2].

III. IMC – GLC Equivalence

The main difference between IMC and GLC, is that the later is a control strategy that is based on an input state-approach, in contrary with IMC which is based on an input – output approach. This is because in GLC the control law is formulated by feeding the states and Lie derivatives from either the plant directly, or from a model of the plant. This is not necessary the case with IMC. In the later case, an inverted model of the plant or a model of the plant is used. This controller does not have to be based on the state representation of the plant, and neither it requires any states or Lie derivatives. Using the Hirschorn inverse in IMC, the procedure is expanded to input–state–output approach, because the Hirschorn inverse requires the states and Lie derivatives of the plant. This gives an edge to the design of the IMC described above, and makes it similar to the GLC approach.

Let us consider the case of Fig. 5. Here, we have the IMC structure with a Hirschorn inverse controller and a linear filter. Consider the case where the model matches the plant exactly and therefore the feedback will be zero at all time. Also because of the linear dynamics,

$$\begin{aligned} e = v = w^{(r)} + a_{r-1} w^{(r-1)} + \ldots + a_0 w \Leftrightarrow \\ w^{(r)} = v - a_{r-1} w^{(r-1)} - \ldots - a_0 w \end{aligned} \tag{13}$$

Since there will be exact cancellation of dynamics in plant and inverse, *w*=*y*. Then because of (3) and (13), the input to the plant will be:

$$u=\frac{w^{(r)}-L_f^r h(x)}{L_g L_f^{r-1}h(x)}=\frac{v-a_{r-1}w^{(r-1)}-\ldots-a_0 w-L_f^r h(x)}{L_g L_f^{r-1}h(x)}=$$
$$=\frac{v-L_f^r h(x)-a_{r-1}L_f^{(r-1)}h(x)-\ldots-a_0 h(x)}{L_g L_f^{r-1}h(x)}$$

But this is the same as in the linearisation procedure with GLC. This is a very important result. It signifies that under certain conditions the IMC strategy is equivalent to the GLC. Therefore we can say that:

Proposition 1: The IMC strategy is equivalent to the linearisation procedure of the GLC strategy under the following conditions:

1) The IMC controller is a Hirschorn inverse controller.

2) The dynamics and states of the plant or a model of the plant compose the controller.

3) In case a model is used the model matches the plant exactly.

4) A linear filter is used with relative order equal to that of the plant / model.

Under these conditions the plant is linearised to the linear dynamics of the filter.

As we have seen, it is not always possible to have an exact match between plant and model. Assuming that this is the case, then $x \simeq z$, $f(x) \simeq \hat{f}(z)$, $g(x) \simeq \hat{g}(z)$ and $h(x) \simeq \hat{h}(z)$. In this case, the Hirschorn inverse will completely cancel out the dynamics of the model, and therefore we can safely assume that the input to the inverse is the output of the model i.e. referring to, $w=\hat{y}$.

Then using (3) and (13) the input to the plant will be:

$$\left.\begin{array}{l} u=\dfrac{w^{(r)}-L_{\hat{f}}^r \hat{h}(x)}{L_{\hat{g}} L_{\hat{f}}^{r-1}\hat{h}(x)}=\dfrac{e-a_{r-1}w^{(r-1)}-\ldots-a_0 w-L_{\hat{f}}^r \hat{h}(x)}{L_{\hat{g}} L_{\hat{f}}^{r-1}\hat{h}(x)} \\ \qquad\qquad\qquad\qquad e=v-\varepsilon_o \end{array}\right\}\Leftrightarrow$$
$$u=\frac{v-\varepsilon_o-L_{\hat{f}}^r \hat{h}(x)-a_{r-1}L_{\hat{f}}^{r-1}\hat{h}(x)-\ldots-a_0\hat{h}(x)}{L_{\hat{g}} L_{\hat{f}}^{r-1}\hat{h}(x)}\Rightarrow$$
$$u=\frac{v-L_{\hat{f}}^r \hat{h}(x)-a_{r-1}L_{\hat{f}}^{r-1}\hat{h}(x)-\ldots-a_0\hat{h}(x)}{L_{\hat{g}} L_{\hat{f}}^{r-1}\hat{h}(x)}-\frac{\varepsilon_o}{L_{\hat{g}} L_{\hat{f}}^{r-1}\hat{h}(x)}$$

Therefore even in the case where small model / plant mismatches exist, the input to the plant is still similar Between the two strategies, with a small discrepancy proportional to the model-plant mismatch. Therefore **proposition 1** can be extended to:

Proposition 2: The IMC strategy is equivalent to the linearisation procedure of the GLC strategy within small discrepancies, under the following conditions:

1) The IMC controller is a Hirschorn inverse controller.

2) The dynamics and states of the plant or a model of the plant compose the controller.

3) A linear filter is used with relative order equal to that of the plant / model.

Under these conditions the plant is linearised to the linear dynamics of the filter. The discrepancies are proportional to the model-plant mismatch.

IV. SIMULATIONS

In this section we will test the two control strategies for a non-linear plant, using a variety of models. The plant selected for the simulations is a well-known, well-used non-linear system [8-10, 12-18].

The Single Link Manipulator (SLM) is essentially a pendulum, and the control problem is to control at any point in time, its position and velocity. The state space equations for this system are [8-10, 12-18]:

$$\left.\begin{array}{l} \dot{x}_1=x_2 \\ \dot{x}_2=-9.8\sin x_1-3x_2+0.5u \\ y=x_1 \end{array}\right\} \qquad (14)$$

We will test the two strategies first using the exact same system as a model. This will provide us with a benchmark case that will be used to compare with the other to cases. In the second simulation errors are introduced to the model dynamics in order to test simple plant – model mismatches. Finally the third simulation uses a trained Dynamic Recurrent Neural Network (DRNN) as a model for the plant. In all simulations the IMC linear filter and the target linear dynamics of the GLC were the same.

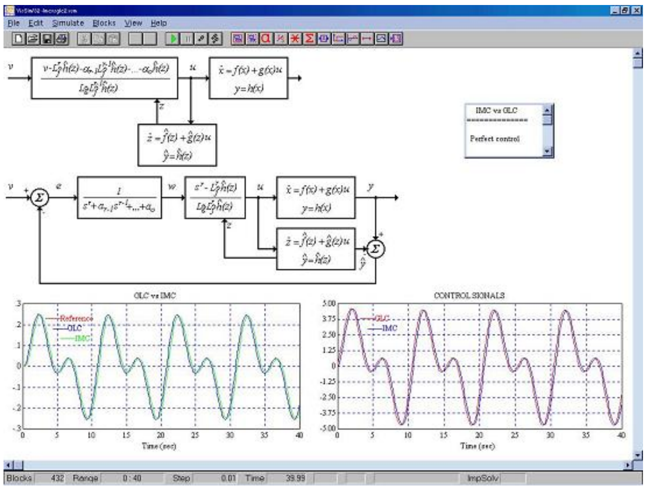

Fig. 6 IMC vs GLC using an exact model

In Fig. 6, the two strategies are implemented when the model used to produce the control strategies is equal to the plant. In this case, the plant output for both GLC and IMC strategies follows the reference exactly. More important, the control signals for both strategies can be seen to be identical.

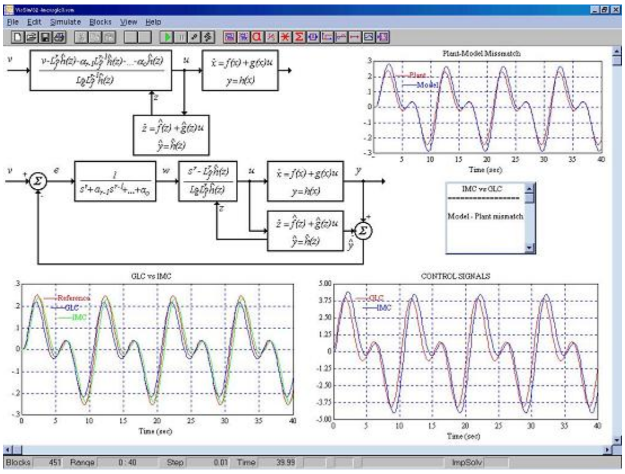

Fig. 7 IMC vs GLC using a non-exact model

In order to generalise the above observation an almost identical system is employed as a model for the two strategies. The state equations of this system are:

$$\left.\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -15\sin x_1 - 8x_2 + u \\ y &= x_1 \end{aligned}\right\}$$

As it can be seen in Fig. 7 even with some model inaccuracies, the control inputs to the plant in both the IMC and the GLC are almost identical. Notice that even though the output of the SLM controlled with IMC is still good, it is not as good as in the previous simulation. Since the model is now distinctly different from the plant, the *dual stability criterion*, no longer holds.

Finally a two-neuron Hopfield network was trained as a model of the SLM using a simple Genetic Algorithm. The state equations of the Hopfield network are given by:

$$\left.\begin{aligned} \dot{x}_1 &= -\beta_1 x_1 + w_{11}\tanh(x_1) + w_{12}\tanh(x_2) + \gamma_1 u \\ \dot{x}_2 &= -\beta_2 x_2 + w_{21}\tanh(x_1) + w_{22}\tanh(x_2) + \gamma_2 u \\ y &= \dot{x}_1 \end{aligned}\right\}$$

The values of the weights where trained to:

$$\begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} -22.9495 & 1.12802 \\ -8.67581 & 89.1383 \end{bmatrix},$$

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} -22.9495 \\ 89.6383 \end{bmatrix}, \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} = \begin{bmatrix} -0.016585 \\ 0.43926 \end{bmatrix}$$

Notice that $\gamma_1 \simeq 0$. According to [19] the neural network has a relative order equal to that of the plant, indicating that the network has learned the dynamics of the plant correctly. In addition the eigenvalues of the network can easily be calculated and are almost identical to those in (14). The controlled plant can be seen in Fig. 8.

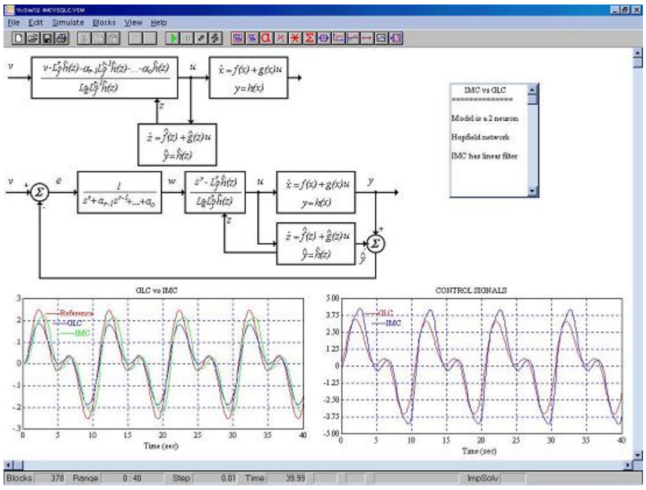

Fig. 8: IMC vs GLC using a Hopfield network as a model

In all simulations the IMC is performing slightly better. This is because the IMC utilises feedback that cancels out exactly any model / plant mismatches as we have seen in the previous sections. However GLC does not employ feedback and thus there is always an error as described in (11). However, in real systems it is not very easy to calculate the derivative of a signal. The effect of having errors in the derivative calculation when IMC strategy is applied can be seen in Fig. 9. In this simulation, errors in the derivative where introduced by the addition of a random number with a Gaussian distribution. As predicted, the error in the derivative is integrated at the plant output, and therefore the linearised plant output is drifting with time.

V. Conclusions

We have investigated two control strategies that are widely used in non-linear control. Both control strategies require models that correctly identify the plant's dynamics. Internal Model Control, as the name suggests, utilizes the inverse of the model of the plant. IMC can be viewed as an input-output approach of dynamic output feedback control. For the IMC strategy to be stable, the plant, its model and the inverse have to be stable. If the model is only modelling the input-output behaviour of the nonlinear plant, then the use of the Hirschorn inverse fails since there are pole-zero cancellations at the origin, giving rise to internal stability problems. However, if the model is correctly identifying the dynamics of the plant, IMC becomes a state-space approach of dynamic output feedback control, since the inverse is based on a state-space framework. In addition the model can be used as an observer providing the approximations of the states of the plant. IMC can compensate for any model / plant mismatches, but the calculation of the derivative of the reference signal that is required can be difficult, and any errors will lead to an integrated error.

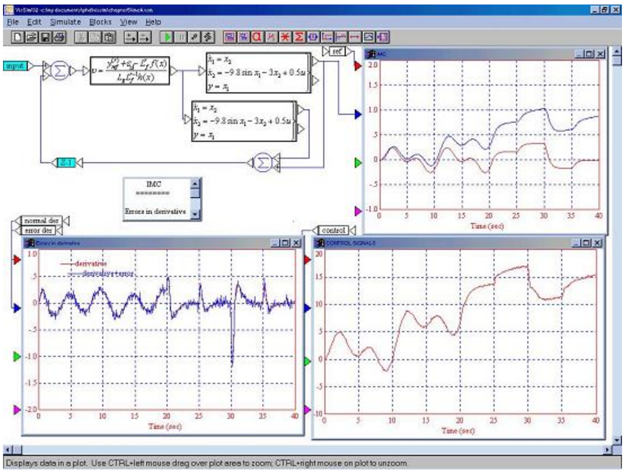

Fig. 9 IMC strategy with an exact model by errors in the derivative calculation

Globally Linearising Control (GLC) is a state-space approach of dynamic output feedback control, where a model of the process is used to create a linearising feedback, making it possible to control the resultant linearised plant with conventional linear controllers. It has been proposed that the two strategies are equivalent under certain conditions. These conditions require that the inverse of the model used in IMC is a Hirschorn inverse, that the model in addition correctly identifies the non-linear dynamics of the plant, and that the linear filter is of equal relative order as the plant. In this case, the linearised system will exhibit the linear dynamics of the filter, in the same manner as in the case that GLC is used.

REFERENCES

[1] C. E. Garcia and M. Morari, "Internal Model Control. 1. A unifying review and some new results," *Ind. Eng. Chem. Process Des. Dev.*, vol. 21, pp. 308-323., 1982.

[2] C. G. Economou, M. Morari, and B. O. Palsson, "Internal Model Control. 5. Extension to Nonlinear Systems," *Ind. Eng. Chem. Process Des. Dev.*, vol. 25, pp. 403-411., 1986.

[3] K. J. Hunt and D. Sbarbaro, "Neural Networks For Nonlinear Internal Model Control," *IEE Proceedings-D Control Theory and Applications*, vol. 138, pp. 431-438, 1991.

[4] C. Kravaris and J. C. Kantor, "Geometric Methods For Nonlinear Process-Control .2. Controller Synthesis," *Industrial & Engineering Chemistry Research*, vol. 29, pp. 2310-2323, 1990.

[5] P. Daoutidis and C. Kravaris, "Dynamic ouput-feedback control of minimum-phase nonlinear processes," Chemical Engineering Science, vol. 47, pp. 837-849, 1992.

[6] A. Isidori, Nonlinear Control Systems. An Introduction, 2nd ed: Springer-Verlag, 1989.

[7] R. M. Hirschorn, "Invertibility of Nonlinear Control Systems," SIAM. J. Cont. Opt., vol. 17, pp. 289-297., 1979.

[8] A. Delgado and C. Kambhampati, "Global linearising control using recurrent networks," presented at European Robotics and Intelligent systems Conference (EURICON), Malaga, Spain, 1994.

[9] A. Delgado, "Input/output linearisation of control affine systems using neural networks," in Department of Cybernetics. Reading: University of Reading, 1996.

[10] A. Delgado, C. Kambhampati, and K. Warwick, "Input/output linearization using dynamic recurrent-neural networks," Mathematics and Computers in Simulation, vol. 41, pp. 451-460, 1996.

[11] A. Delgado, C. Kambhampati, and K. Warwick, "Stable linearization using multilayer neural networks," presented at UKACC interantional conference on control, Exeter, UK, 1996.

[12] A. Delgado and C. Kambhampati, "Differential Geometric Control of Non-linear System with Recurrent Neural Networks," Department of Cybernetics, University of Reading, Reading 1994 1994.

[13] A. Delgado, C. Kambhampati, and K. Warwick, "Approximation of non-linear systems using dynamic recurrent neural networks," presented at International conference on identifications in engineering systems, Univ. of Wales, Swansea, U.K., 1996.

[14] K. P. Dimopoulos and C. Kambhampati, "Multiple Interconnected Hopfield Networks for Intelligent Global Linearising Control," presented at IJCNN, Washington D.C. USA, 1999.

[15] K. P. Dimopoulos, C. Kambhampati, and R. J. Craddock, "Efficient recurrent neural network training incorporating a priori knowledge," Mathematics and Computers in Simulation, vol. 52, pp. 137-162, 2000.

[16] C. Kambhampati, R. J. Craddock, M. Tham, and K. Warwick, "Inverse Model Control Using Recurrent Networks," To appear in Mathematics and Computers in Simulation., 1998.

[17] C. Kambhampati, R. J. Craddock, M. Tham, and K. Warwick, "Internal model control of nonlinear systems through the inversion of recurrent neural networks," World Congress on Computational Intelligence, Alaska., 1998.

[18] A. Delgado, C. Kambhampati, and K. Warwick, "Dynamic recurrent neural networks for system identification and control," IEE Proc. Control Theory Appl., vol. 142, pp. 307 - 314., 1995.

[19] K. P. Dimopoulos and C. Kambhampati, "Apriori Information in Network Design," in Dealing with Complexity: A Neural Network Approach., M. Karny, K. Warwick, and V. Kurkova., Eds., 1998.