

# Estimation and control of mechatronic systems using sensitivity bond graphs <sup>1,2</sup>

Peter J Gawthrop <sup>a,3</sup> Eric Ronco <sup>b</sup>

<sup>a</sup>*Centre for Systems and Control and Department of Mechanical Engineering,  
University of Glasgow, GLASGOW. G12 8QQ Scotland.*

*email: P.Gawthrop@eng.gla.ac.uk*

*fax: +44 141 330 4343*

<sup>b</sup>*Institut d'Automatique, EPFL, 1015 Lausanne, Switzerland.*

*Eric.Ronco@epfl.ch*

---

## Abstract

A new bond graph framework for sensitivity theory is applied to model-based predictive control, state estimation, and parameter estimation in the context of physical systems. The approach is illustrated using a nonlinear mechatronic system.

*Key words:* Bond graphs; Sensitivity; Optimisation; Predictive control; Parameter estimation.

---

---

<sup>1</sup> Suggested running headline: "Estimation and Control of Mechatronic Systems"

<sup>2</sup> Extended version of paper submitted to 1st IFAC Conference on Mechatronic Systems, Darmstadt, September 2000

<sup>3</sup> Author for correspondence

## 1 Introduction

This paper combines a number of ideas to give a new adaptive model-based output feedback controller applicable to practical non-linear systems. The key ideas used are:

- bond graph modelling (Karnopp *et al.*, 1990; Thoma, 1990; Cellier, 1991; Gawthrop and Smith, 1996)
- sensitivity bond graphs (Cabanellas *et al.*, 1995; Gawthrop, 1999)
- optimisation (Fletcher, 1987; Press *et al.*, 1992)
- predictive pole-placement control (Gawthrop and Ronco, 1999)
- intermittent control (Ronco *et al.*, 1999)
- partially-known system identification (An *et al.*, 1988; Canudas de Wit, 1988; Dasgupta *et al.*, 1986; Gawthrop *et al.*, 1992; Gawthrop *et al.*, 1993)

Bond graphs provide a well-established technique for modelling dynamic systems; details may be found in the textbooks of Karnopp *et al.* (1990), Thoma (1990), Cellier (1991) and Gawthrop and Smith (1996). As a graphical approach, a number of computer-based graphical bond graph modelling tools have appeared including MTT (2000) – the one used to implement the ideas contained in this paper. The bond graph approach can be compared and contrasted with other methods under a number of different headings. Firstly, bond graphs are *equation* based (as opposed to *assignment statement* based). They share this property with approaches such as Modelica (2000) and Ascend (1999); but for this reason, the approach is superior to block diagram based approaches such as Simulink. Secondly they provide an *energy-based* approach which not only allows multi-domain modelling but ensures that the resultant model is energetically correct.

The bond graph approach has previously been suggested as a basis for control design by Karnopp (1979), Karnopp (1995) and by Gawthrop (1995*b*). This

paper develops the idea of *sensitivity* bond graphs (Cabanellas *et al.*, 1995; Gawthrop, 1999) to provide a basis for model-based optimisation for the control and estimation of nonlinear dynamic systems for which a bond graph model is available. In so doing, it leads on to a bond graph based computer environment which seamlessly combines bond graph tools ranging from modelling and model-based control design through to real-time identification and control.

The *sensitivity* theory of dynamic systems and its application is well established and summarised in the textbooks of Tomović and Vukobratović (1972) and Frank (1978). There are many applications of sensitivity methods to systems and control problems including: system optimisation (Cabanellas *et al.*, 1995), controller tuning (Van Amerongen and Udink ten Cate, 1975; Winning *et al.*, 1977; Oppen *et al.*, 1995) and parameter estimation (Eykhoff, 1974). The particular class of dynamic systems described by electrical networks has its own techniques (Calahan, 1972) based on the *adjoint circuit* approach. In contrast to standard sensitivity theory (Tomović and Vukobratović, 1972; Frank, 1978), which operates at the system ordinary differential equation level, bond graph based sensitivity models (Cabanellas *et al.*, 1995; Gawthrop, 1999) operate at the modelling level. In the special case of linearisation, Karnopp (1977) has stated: “Rather than treating linearized systems as abstract sets of equations, we here look for structural analogies between non-linear components relating total system variables and linearized models relating incremental system variables”; with “linearized” replaced by “sensitivity” this statement summarises the approach of Gawthrop (1999) used as a basis for this paper.

*Optimisation* has always been a fundamental technique in control and estimation of dynamic systems, and this has become even more so with the popularity of Model-based Predictive Control (MPC) (Clarke, 1994; Muske and Rawlings, 1993; Gawthrop *et al.*, 1998; Chen and Allgöwer, 1998; Kou-

varitakis *et al.*, 1999; Gawthrop and Ronco, 1999). Such methods rely on a *model* of the corresponding physical system. In some application areas (for example process control) physical, or first principles, models are hard to come by and so empirical models tend to be used. However in other application areas, in particular mechatronics, physical models are more readily available.

The predictive pole-placement (PPP) control introduced by Gawthrop and Ronco (1999) embeds the classical pole-placement state feedback design into a model-predictive formulation. This provides an alternative to model-predictive controllers which are based on linear-quadratic control. Although developed and analysed in a linear systems context, this paper shows that the method is applicable in the nonlinear case as well. As with any continuous-time method, there is a computational/real-time issue. This is solved using the *intermittent* approach of Ronco *et al.* (1999).

Whilst the emphasis in the paper is on systems that have a well-defined associated physical model, within this structure two forms of uncertainty are allowed: uncertain *states* and uncertain *physical parameters*. The identification of such *partially-known* systems has a long history (An *et al.*, 1988; Canudas de Wit, 1988; Dasgupta *et al.*, 1986; Gawthrop *et al.*, 1992; Gawthrop *et al.*, 1993) which has been given an sensitivity bond graph interpretation by Gawthrop (1999). The state estimation is used to convert the *state* feedback predictive pole-placement to an *output* feedback algorithm.

The outline of the paper follows. Section 2 reviews the sensitivity of systems described by bond graphs and derives some results for the particular bond graph components used in this paper. Section 4 considers nonlinear model-based predictive control based on the linear predictive pole-placement of Gawthrop and Ronco (1999) and model-based parameter and state estimation. Section 5 illustrates the approach using a detailed model of, and data from, a laboratory inverted pendulum experiment. Section 6 makes some con-

cluding remarks.

## 2 Sensitivity bond graphs

Sensitivity bond graphs are discussed by Cabanellas *et al.* (1995) and by Gawthrop (1999), and the closely related topic of linearised system bond graphs are discussed by Karnopp (1977). This section provides a brief introduction to the subject together with details relevant to this paper.

A bond graph component is associated with a *constitutive relationship* (or CR) which relates  $n_{v_i}$  time-varying *signals* within the component and  $n_\theta$  time-invariant *parameters* associated with the system within which the component lies. Thus the CR for the  $i$ th component of a system can be written as:

$$\Phi_i(v_i(t), \theta) = 0 \quad (1)$$

where  $v_i \in \mathfrak{R}^{n_{v_i}}$  contains the *component* signals and  $\theta \in \mathfrak{R}^{n_\theta}$  contains the *system* parameters.

In the special case that the CR is linear; it can be written as:

$$A_i(\theta)^T v_i(t) = 0 \quad (2)$$

where the time-invariant vector  $A_i(\theta) \in \mathfrak{R}^{n_{v_i}}$ .

In this paper, the *sensitivity*  ${}^j v_i \in \mathfrak{R}^{n_{v_i}}$  of the signal vector  $v_i$  with respect to any ( $j$ th) component  $\theta_j$  of the parameter vector  $\theta$  is of interest. In particular,  ${}^j v_i$  is defined as:

$${}^j v_i = \frac{\partial v_i}{\partial \theta_j} \quad (3)$$

It follows from Eqn. 1 that  $\frac{d\Phi_i(v_i(t), \theta)}{d\theta} = 0$  and so the *sensitivity* CR  $\phi_i({}^j v_i(t), v_i(t), \theta)$

becomes:

$$\begin{aligned} \phi_i({}^j v_i(t), v_i(t), \theta) &= \frac{\partial \Phi_i(v_i(t), \theta)^T}{\partial v_i} {}^j v_i(t) \\ &+ \frac{\partial \Phi_i(v_i(t), \theta)}{\partial \theta_j} = 0 \end{aligned} \quad (4)$$

Eqn. 4 is the sensitivity CR with respect to the  $j$ th parameter. It has the following important properties:

- (1) The first term of the right-hand side of Eqn. 4 represents the *linearised* (about  $v_i(t)$ ) CR relating the sensitivity functions  ${}^j v_i(t)$ . In other words, it is a *linear* CR modulated by the variables  $v_i(t)$  associated with the system itself. It can be written as the summation:

$$\frac{\partial \Phi_i(v_i(t), \theta)^T}{\partial v_i} {}^j v_i(t) = \sum_{k=1}^{n_{v_i}} \frac{\partial \Phi_i(v_i(t), \theta)^T}{\partial v_{ik}} v_{ik}^j(t) \quad (5)$$

where  $v_{ik}$  and  $v_{ik}^j$  are the  $k$ th components of the vectors  $v_i$  and  ${}^j v_i$  respectively.

- (2) The second term of the left-hand side of Eqn. 4 represents an additional input to the sensitivity CR dependent on the variables  $v_i(t)$  associated with the system itself.
- (3) The sensitivity CR of Eqn. 4 is *local* to the the component in the sense that the only variables appearing in Eqn. 4 are  $v_i(t)$  and  ${}^j v_i(t)$ .
- (4) The  $j$ th sensitivity CR of Eqn. 4 does *not* depend on  ${}^l v_i(t)$  for  $l \neq j$ .
- (5) If the  $i$ th CR  $\Phi_i(v_i(t), \theta)$  does not depend on  $\theta_j$  then the second term of Eqn. 4 is zero and there is no *explicit* coupling between the actual and sensitivity systems (though there will be *implicitly* if  $\Phi_i(v_i(t), \theta)$  is non-linear). The corresponding sensitivity component is then the *linearised* component (Karnopp, 1977).
- (6) If  $\Phi_i(v_i(t), \theta)$  is *linear* in  $v_i$  (Eqn. 2, then Eqn. 4 becomes:

$$A_i^T(\theta) {}^j v_i(t) + \frac{\partial A_i^T(\theta)}{\partial \theta_j} v_i(t) = 0 \quad (6)$$

- (7) If the conditions of both items 5 and 6 hold; then the CRs of the actual and sensitivity components are identical and uncoupled.
- (8) The bond graph sensitivity component can itself be represented by a bond graph comprising:
  - (a) the original component corresponding to Eqn. 1,
  - (b) the linearised component corresponding to the first term of the right-hand side of Eqn. 4 and
  - (c) a coupling component corresponding to the second term of the right-hand side of Eqn. 4.

An example of this appears in Figure 1.

For these reasons, it is possible to encapsulate two CRs: the system CR of Eqn. 1 and the  $j$ th sensitivity CR of Equation 4 within in a single component containing  $2n_{v_i}$  variables: those contained in  $v_i$  and  ${}^jv_i$ . If the original component had  $N$  ports, the new sensitivity component (s-component) therefore has  $2N$  ports. More conveniently, each port on the original component is replaced by a sensitivity port (or s-port) which carries not only the effort/flow pair  $e$  and  $f$  but also the corresponding sensitivity pair  ${}^je$  and  ${}^jf$ . Such ports may be considered to be connected by a *sensitivity bond* (s-bond) which encapsulates the energy bond carrying  $e$  and  $f$  with the pseudo-bond carrying  ${}^je$  and  ${}^jf$ .

It follows from the above equations that any bond graph component has an s-component equivalent. Moreover, if the corresponding component is *not* dependent on  $\theta$ , it is then the *linearised* component as discussed by Karnopp (1977).

The following two examples are used in the paper.

## 2.1 A linear one-port $\mathbf{R}$ component

The standard linear bond graph  $\mathbf{R}$  component has a single port with effort  $e$  and flow  $f$  covariables related by the CR

$$\Phi_i(v_i(t), \theta) = A_i(\theta)^T v_i(t) = e - r f = 0 \quad (7)$$

That is, in the notation of Eqn. 2

$$A_i(\theta) = \begin{pmatrix} 1 \\ -r \end{pmatrix}; v_i = \begin{pmatrix} e \\ f \end{pmatrix} \quad (8)$$

$\frac{\partial A_i^T(\theta)}{\partial \theta_j} = \begin{pmatrix} 0 & -1 \end{pmatrix}^T$  and so the sensitivity CR of Eqn. 6 can be written as:

$$\frac{\partial e}{\partial r} - r \frac{\partial f}{\partial r} - f = 0 \quad (9)$$

## 2.2 A linear amplifier $\mathbf{AE}$ component

The nonstandard linear bond graph  $\mathbf{AE}$  component has two ports with effort  $e_1$  and  $e_2$  variables related by the CR

$$\Phi_i(v_i(t), \theta) = A_i(\theta)^T v_i(t) = e_2 - K e_1 = 0 \quad (10)$$

As this represents an ideal effort amplifier, the input flow  $f_1 = 0$ .

That is, in the notation of Eqn. 2

$$A_i(\theta) = \begin{pmatrix} 1 \\ -K \end{pmatrix}; v_i = \begin{pmatrix} e_2 \\ e_1 \end{pmatrix} \quad (11)$$



$\frac{\partial A_i^T(\theta)}{\partial \theta_j} = \begin{pmatrix} 0 & -1 \end{pmatrix}^T$  and so the sensitivity CR of Eqn. 6 can be written as:

$$\frac{\partial e_2}{\partial K} - K \frac{\partial e_1}{\partial K} - e_1 = 0 \quad (12)$$

The first two terms of Eqn. 12 correspond to the linear system itself, and the third term corresponds to the sensitivity.

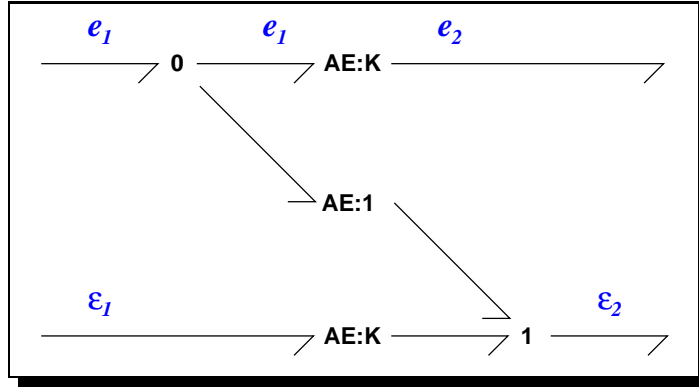


Fig. 1. The sensitivity component **sAE**

Figure 1 gives the bond graph corresponding to the sensitivity component **sAE**. The upper part of the diagram comprises the *actual* **AE** component with gain  $K$  and with CR given by Eqn. 10. The lower part of the diagram corresponds to the *sensitivity* CR where  $\epsilon_i \equiv \frac{\partial e_i}{\partial K}$ . The additional term ( $e_1$ ) of Eqn. 12 is represented by the middle unit gain **AE** component.

### 3 Optimisation

The algorithms for estimation and control considered in this paper give rise to optimisation problems of the form

$$\min_{\theta} J(\theta, t) \quad (13)$$

where  $\theta \in \mathfrak{R}^{n_\theta}$  is a the *parameter vector* and the *cost function*  $J(\theta, t)$  is

$$J(\theta, t) = \frac{1}{2} \int_0^T e^T(t, \tau) Q(\tau) e(t, \tau) d\tau \quad (14)$$

where  $Q(\tau) \in \mathfrak{R}^{n_y \times n_y}$  is a positive semi-definite weighting function and the error  $e(t, \tau) \in \mathfrak{R}^{n_y}$  is

$$e(t, \tau) = [y(t, \tau, \theta) - z(t, \tau)] \quad (15)$$

$y(t, \tau, \theta) \in \mathfrak{R}^{n_y}$  the system output and  $z(t, \tau) \in \mathfrak{R}^{n_y}$  a function of time. In general,  $y$  is *not* linear in  $\theta$  and so  $J(\theta, t)$  is *not* in general quadratic in  $\theta$ .

This paper is concerned with real-time control and estimation and so optimisation speed is of the essence. Therefore, rapid convergence combined with simplicity is desirable. As, using the sensitivity bond graph approach, gradient information is cheaply available, this suggests the use of methods which make use of gradient information. For these reasons, out of the plethora of methods available (see for example the book of Press *et al.* (1992)), the Quasi-Newton method was chosen. Further research may yield alternative choices, but the experience so far has been good. In particular, as shown by Ronco and Gawthrop (1999) this approach is much faster than nonlinear programming methods which do not use derivative information.

Differentiation of Eqns. 13 and 15 with respect to  $\theta$  relate the gradient  $J_\theta$  of  $J$  (with respect to  $\theta$ ) to the corresponding gradient  $y_\theta(t)$  and the output  $y(t)$  as:

$$J(\theta, t)_\theta = \int_0^T [y(t, \tau, \theta) - z(t, \tau)]^T Q(\tau) y_\theta(t, \tau) d\tau \quad (16)$$

The quasi-Newton approach approximates the second derivative  $J(\theta, t)_{\theta\theta}$  of the cost function by:

$$J(\theta, t)_{\theta\theta} \approx \hat{J}(\theta, t)_{\theta\theta} = \int_0^T y_\theta(t, \tau) Q(\tau) y_\theta^T(t, \tau) d\tau \quad (17)$$

As discussed by Gawthrop (1999) (and summarised in Section 2 The optimisation algorithm is then to repeatedly compute:

$$\theta := \theta - \Delta\theta \tag{18}$$

until some convergence criterion is satisfied where  $\Delta\theta$  is the solution of the set of linear equations:

$$\hat{J}_{\theta\theta}\Delta\theta = J_\theta \tag{19}$$

Two simple modifications of this method give additional robustness in the face of difficult optimisation problems (Press *et al.*, 1992):

- (1) Eqn. 19 is solved via a singular value decomposition based pseudo inverse and
- (2) A check is made that the cost function decreases at each step; if it doesn't, the step length is multiplied by the scalar  $0 < \beta < 1$  (whilst retaining the step direction) until it does.

Finally, it is worth emphasising that the optimisation algorithm attempts to minimise a the square of a *nonlinear* function with the help of the sensitivity (linearised) system – the sensitivity system is *not* itself optimised.

#### 4 Model-based estimation and control

There are many approaches to model-based predictive control including those described by Muske and Rawlings (1993),Gawthrop *et al.* (1998),Chen and Allgöwer (1998) and Kouvaritakis *et al.* (1999).

However, the recently developed method of Gawthrop and Ronco (1999) is particularly appropriate to the sensitivity bond graph theme of this paper; however, this does not exclude the possibility of bringing other methods within

the same framework. This method is briefly described in section 4.1.

In common with many model-based predictive control, the one described in Section 4.1 is a *state* feedback method. Therefore, to give *output* feedback, Section 4.2 develops a simple, but novel, non-linear observer which can be used for parameter, as well as state, estimation.

These two algorithms are brought together within the intermittent control context discussed by Ronco *et al.* (1999) to give the overall implementable algorithm in Section 4.3.

#### 4.1 Model-based predictive control

Much work on Model-based Predictive Control is in a discrete-time setting and therefore inappropriate to the context of this paper. However a number of continuous-time approaches are available including those of Demircioglu and Gawthrop (1991), Gawthrop *et al.* (1998) and Chen and Allgöwer (1998). More recently, a new approach *Predictive Pole Placement* (PPP) has been developed by Gawthrop and Ronco (1999) which, although for linear systems, readily extends to this non-linear context.

The nonlinear systems considered in this paper are represented by:

$$\begin{cases} E(x) \frac{dx}{dt} & = f(x, u) \\ y & = g(x) \end{cases} \quad (20)$$

Such *constrained-state* equations can be derived directly from the system bond graph (Gawthrop and Smith, 1996). In simple cases  $E(x) = I$ , the unit matrix, in which case Eqn. 20 is in standard ordinary differential equation form. In typical mechatronic systems,  $E(x) > 0 \forall x$ . Despite the fact that Eqn. 20 could

be then rewritten as:

$$\begin{cases} \frac{dx}{dt} & = f_e(x, u) \\ y & = g(x) \\ f_e(x, u) & = E^{-1}(x)f(x, u) \end{cases} \quad (21)$$

it is often better to work with Eqn. 20 directly as the inverse of  $E(x)$  can be a complicated algebraic expression.

As in the linear case discussed by Gawthrop and Ronco (1999) (and many other model-based predictive controllers), interest lies in the solutions of

$$\begin{cases} E(x^*(t, \tau)) \frac{d}{d\tau} x^*(t, \tau) & = f(x^*(t, \tau), u^*(t, \tau)) \\ y^*(t, \tau) & = g(x^*(t, \tau)) \end{cases} \quad (22)$$

The differential Eqns. 20 and 22 are related by having the *same* state space matrices and by imposing the *cross-coupling* conditions:

$$\begin{cases} x^*(t, 0) & = x(t) \\ u(t) & = u^*(t, 0) \end{cases} \quad (23)$$

As in the approach of Gawthrop and Ronco (1999), the *moving horizon* control signal  $u^*(t, \tau)$  is *linearly parameterised* by the  $n_U$  components of the column vector  $U(t)$  so that:

$$u^*(t, \tau) = U^*(\tau)U(t) \quad (24)$$

where  $U^*(\tau)$  is a  $n_u \times n_U$  matrix of functions of  $\tau$ . For the purposes of this paper, the particular  $U^*(\tau)$  given by:

$$U^{*T}(\tau) = e^{A_u \tau} U_0 \quad (25)$$

is chosen. That is  $U^{*T}(\tau)$  is the state of the autonomous system

$$\begin{cases} \frac{d}{d\tau}U^{*T}(\tau) &= A_u U^{*T}(\tau) \\ U^*(0) &= U_0^T \end{cases} \quad (26)$$

The components of  $U^*(\tau)$  can be regarded as a set of *basis functions* for the control signal  $u^*(t, \tau)$  and the components of  $U(t)$  the corresponding weights or *tuneable parameters*. This idea is equally applicable to the nonlinear case.

Similarly, the *moving horizon* setpoint  $w^*(t, \tau)$  is linearly parameterised by the  $n_W$  components of the column vector  $W(t)$  so that:

$$w^*(t, \tau) = W^*(\tau)W(t) \quad (27)$$

where  $W^*(\tau)$  is a  $n_y \times n_W$  matrix of functions of  $\tau$ . Typically the components  $W_i^*(\tau)$  of  $W^*(\tau)$  will be constant:

$$W_i^*(\tau) = \begin{cases} 1 & \text{for tracking} \\ 0 & \text{for regulation} \end{cases} \quad (28)$$

In the particular case of predictive control, the optimisation cost  $J(\theta)$  is of the form of Eqns. 13 and 14 where  $T = T_2$  is the upper time horizon,  $\theta = U(t)$  and  $z(t, \tau) = w^*(t, \tau)$ . The weighting function  $Q(t)$  is:

$$Q(t) = \begin{cases} 0_{n_y} & \text{if } t \leq T_1 \\ I_{n_y} & \text{if } t > T_1 \end{cases} \quad (29)$$

where  $0_{n_y}$  and  $I_{n_y}$  are the  $n_y \times n_y$  zero and unit matrices respectively. As discussed by Gawthrop and Ronco (1999), the aim of this cost is to make the system output as close to the setpoint as possible between times  $T_1$  and  $T_2$  whilst ignoring initial transient behaviour.

Unlike the linear case, the cost function  $J(\theta, t)$  will no longer be necessarily quadratic in  $U(t)$  and so explicit minimisation is no longer possible. However,

numerical optimisation is simplified if the derivative  $J_\theta(\theta, t)$  is available. Computation of  $J_\theta(\theta, t)$  requires, in turn, the computation of  $y_U^*(\tau)$ ; this is precisely the information available from the sensitivity bond graph.

### Bond Graph Interpretation

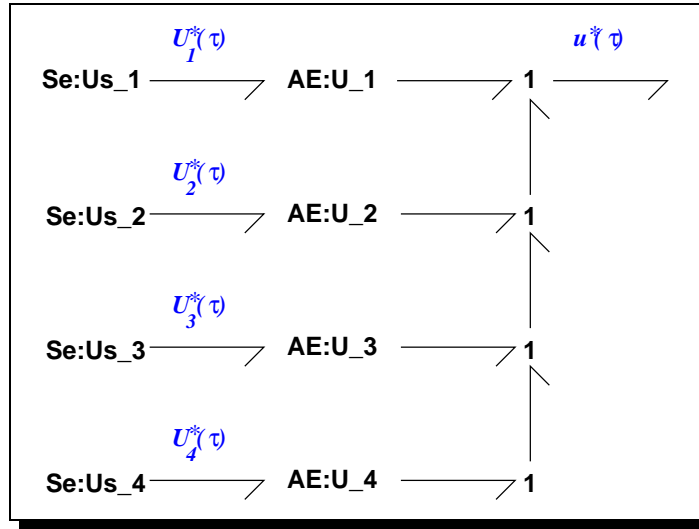


Fig. 2. Open-loop control

Here, a new sensitivity bond graph interpretation of the approach of Gawthrop and Ronco (1999) is given. The discussion is restricted to the single input case ( $n_u = 1$ ) assuming that the input is an effort, but can readily be extended to the more general case.

Eqn. 24 is the core of PPP and thus giving a bond graph representation of this equation is a key issue. This bond graph representation is given for the case when  $n_U = 4$  in Figure 2. The individual parts of Eqn. 24 are given bond graph interpretations as follows.

$U(t)$ : Each element of the (adjustable) weight vector  $U(t)$  ( $U_i(t)$ ) is interpreted as the *gain* of a bond graph amplifier (**AE**) component (see Section 2.2).

$U^*(\tau)$ : The corresponding element of the *basis function* vector  $U^*(\tau)$  ( $U_i^*(\tau)$ )

is interpreted as the *output* of a bond graph source **Se** component which acts as the *input* of the corresponding **AE** component. (The **Se**, as opposed to the *modulated MSe* source component, is appropriate here as it represents a *fixed* function of time as specified by, for example, Eqn. 25.)

Thus the output of each **AE** component represents the *product* of  $U_i(t)$  and  $U_i^*(\tau)$ ; these products are then *summed* at a **1** junction to implement the scalar product implied by Eqn. 24.

With this interpretation, the problem of determining the sensitivity  $y_U^*(\tau)$  of the system output  $y^*(t, \tau)$  of the dynamic system of Eqn. 22 with respect to the weighting vector  $U(t)$  is interpreted as finding the sensitivity bond graph corresponding to the the bond graph of the dynamic system augmented by the bond graph of Figure 2.

In particular, using the sensitivity bond graph approach of Section 2, the bond graph of Figure 2, together with the rest of the dynamic system, is converted into an sensitivity bond graph, thus giving the *sensitivity* of the system outputs with respect to each of the  $n_U$  amplifier *gains* –  $n_U$  components of  $U(t)$  required for Eqns. 16 and 17.

#### 4.2 Parameter and state estimation

Each bond graph component has a constitutive relationship containing a number of *parameters*. Thus, for example, the **R** component with CR of Eqn. 7 is parameterised by  $r$ . Because of the close mapping of the bond graph to the corresponding physical system, each such parameter has a precise physical meaning. Many of these parameters (such as lengths and masses) will be known a-priori from data sheets or physical inspection; others such as friction coefficients will not be precisely known.



Therefore, many systems will be *partially-known* in this sense. Such systems are typically not linear in the system parameters and various approaches to this issue are described elsewhere (Dasgupta *et al.*, 1986; Canudas de Wit, 1988; Gawthrop *et al.*, 1992). Here, an off-line optimisation approach based on Section 3 is adopted.

The signal  $z(t, \tau)$  of Eqn. 15 is defined as:

$$z(t) = y_m(t + \tau) \quad (30)$$

where  $y_m(t)$  is the *measured* actual system output. The parameter vector  $\theta$  contains the system parameters. The weighting function  $Q$  is

$$Q(t) = I_{n_y} \quad (31)$$

giving equal weighting to the parameter error.

As noted previously by Gawthrop (1995*a*), the initial state of a dynamic system represented by a bond graph may be explicitly represented by the addition of an **Se** component to the corresponding **C** components and an **Sf** component to the corresponding **I** components. Thus the initial state translates to a system parameter – the value of the source output – and so these parameter identification techniques can be applied equally well to initial state estimation.

### 4.3 *The intermittent approach*

Continuous-time predictive control algorithms have the apparently fatal drawback that optimisation must be completed within an infinitesimal time. However, this problem can be overcome using intermittent control see, for example (Ronco *et al.*, 1999) for a detailed discussion of the approach.

Briefly, the idea is to update the control weights *intermittently*, and, during this time  $T_{ol}$ , run the controller in open loop using the previously calculated

trajectory. Thus there are two processes running in parallel: generation of the open-loop control and computation of the control weights  $U(t)$  ready for the *next* open-loop control trajectory. Thus control is continuous but feedback intermittent: this has biological analogies.

As discussed by (Gawthrop and Ronco, 1999), the PPP optimisation problem can be solved explicitly or, if computed recursively, the Quasi newton algorithm converges in a single step. However, in the non-linear case, convergence can take many steps depending on the initial choice of  $\theta = U(t)$ . Therefore, the choice of the initial value is important to ensure rapid convergence. In the linear case, it is known that, in the absence of disturbances, the open and closed loop trajectories are the same and thus the open loop control trajectory in one interval is the *continuation* of that in the previous interval. This will not be true in the non-linear case, but nevertheless provides a good starting value for the optimisation. The following Lemma provides the appropriate information.

**Lemma 1 (Trajectory Continuation)** *If  $U^*(\tau)$  is given by Eqn. 25 and the control weighting function  $U(t)$  at time  $t = (k + 1)T_{ol}$  is related to that at time  $t = kT_{ol}$  by Eqn. 32*

$$U((k + 1)T_{ol}) = e^{A_u^T T_{ol}} U(kT_{ol}) \quad (32)$$

*then the control trajectory within the  $(k + 1)$ th open-loop interval is the continuation of the trajectory in the previous interval in the sense that*

$$u^*((k + 1)T_{ol}, \tau) = u^*(kT_{ol}, \tau + T_{ol}) \quad (33)$$

**PROOF.** Using Eqns. 24 and 25, it follows that

$$u^*(kT_{ol}, \tau + T_{ol}) = U_0^T e^{A_u^T (k+1)T_{ol}} U(kT_{ol}) \quad (34)$$

$$= U_0^T e^{A_u^T kT_{ol}} e^{A_u^T T_{ol}} U(kT_{ol}) \quad (35)$$

- (1) Compute the open-loop control using the *predefined*  $U^*(\tau)$  together with  $U(kT_{ol})$  computed at iteration  $k - 1$ :

$$u(\tau + kT_{ol}) = U^*(\tau)U(kT_{ol}) \quad (36)$$

- (2) Compute  $U((k + 1)T_{ol})$ :

- (a) Using output data from the *previous* interval ( $(k-1)T_{ol} \leq t < kT_{ol}$ ), and the previously estimated “initial” state use the method of Section 4.2 to estimate the system “initial” state at  $t = (k-1)T_{ol}$  and, optionally, system parameters.
- (b) Use to model to compute the *current* state at  $t = kT_{ol}$ .
- (c) Use to model to compute the *predicted* state at  $t = (k + 1)T_{ol}$ .
- (d) Use the continuation trajectory condition (Equation 32) to compute a starting value for the PPP optimisation.
- (e) Using the *predicted* state, together with estimated system parameters and the model, use the method of Section 4.1 to compute the control weighing function  $U((k + 1)T_{ol})$  for the *next* algorithm iteration.

Table 1

The Algorithm

The result follows using Eqn. 32.  $\square$

To summarise: parts 1 and 2 of the algorithm of Table 1 are executed *in parallel* every  $T_{ol}$  seconds. Part 1 of the algorithm is the open-loop control; and part 2 of the algorithm updates the state and parameter estimates together with the control weights for the *next* iteration. The iteration is indexed by  $k$

Parameter	Units	Value	Description
$l_a$	H	0.18e-3	Motor armature inductance
$j_a$	kgm <sup>2</sup>	3.87e-7	Motor armature inertia
$r_a$	$\Omega$	2.6	Motor armature resistance
$f_a$	Nm <sup>-1</sup> s	<b>unknown</b>	Motor armature friction
$k_g$	none	1/3.7	Motor gear ratio
$r$	m	0.635e-2	Cart wheel radius
$m_c$	kg	0.7429	Cart mass
$f_c$	Nm <sup>-1</sup> s	<b>unknown</b>	Cart friction
$m_p$	kg	0.210	Pendulum mass
$l_p$	m	0.61m	Pendulum length
$j_p$	kgm <sup>2</sup>	$\frac{1}{12}m_p l_p^2$	Pendulum inertia
$f_p$	kgm <sup>2</sup>	<b>negligible</b>	Pendulum friction

Table 2

Physical parameters

## 5 Example

A commercial laboratory inverted pendulum system described by Apkarian (1995) is pictured in Figure 3(a). This example was chosen for a number of reasons: the system is nonlinear, the system is non-square (two output, one input) and its dynamics, though complex, can be readily captured by the bond graph approach.

As a typical mechatronic system, some parameters are known and some are not. Table 2 shows a list of the relevant physical parameters of this system.

Apart from the three friction parameters ( $f_a$ ,  $f_c$  and  $f_p$ ) these parameters are all listed in the system manual and, where appropriate were checked by direct measurements on the component parts of the system.

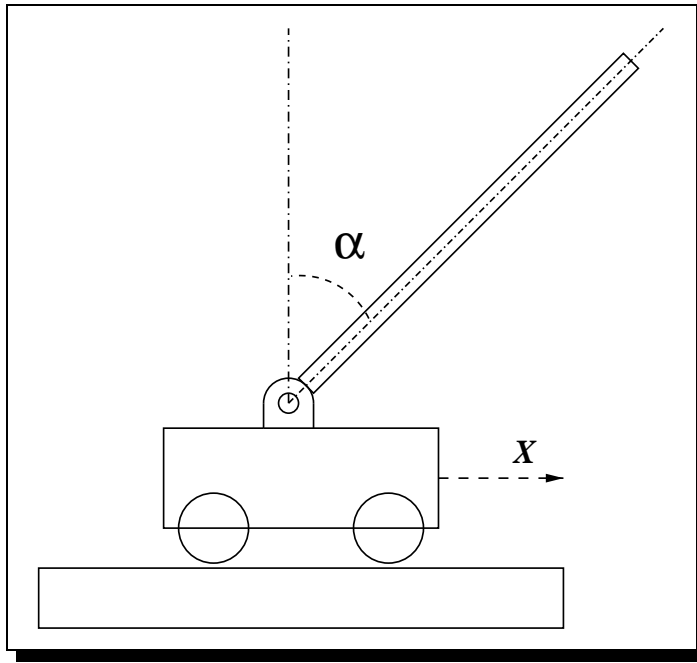
The construction of the system was such that the friction at the joint of the pendulum was negligible. The remaining friction coefficients were estimated from *measured data* using the methods of Section 4.2 as described in Section 5.2.

The system is designed for using simple *linear* control techniques and the pendulum angle is restricted to lie within a small distance from vertical and the system is therefore incapable of large movements. Hence the properties of the non-linear predictive pole-placement algorithm are demonstrated using a *simulated* version of the model of the system in Section 5.3.

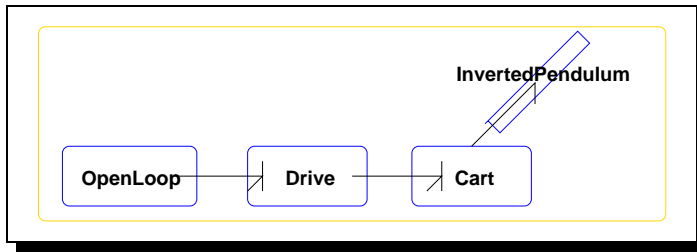
### 5.1 Bond graph model

Bond graph modelling is a well-established technique and a number of textbooks exist including those of Karnopp *et al.* (1990), Thoma (1990), Cellier (1991), and Gawthrop and Smith (1996). To avoid unnecessary detail, the *word* bond graph of the higher levels of the system is presented in Figure 3(b) and the *word* bond graph of two of the subsystems in 4. The open loop control bond graph (see Section 4.1) appears in Figure 2 The system nonlinearities arise from the angle-modulated transformers of Karnopp (1969) corresponding to the kinematic transformations inherent in the inverted pendulum problem. The details of the pendulum model are the same as presented by Gawthrop and Smith (1996).

The resulting system has five states (the pendulum angle and angular momentum, the cart position and momentum and the motor armature current). The pendulum model contains a number of  $I$  components in derivative causality



(a) Schematic

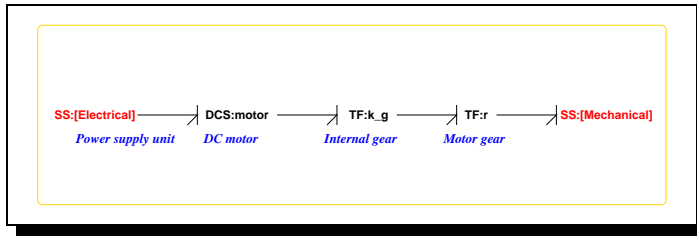


(b) Word bond graph

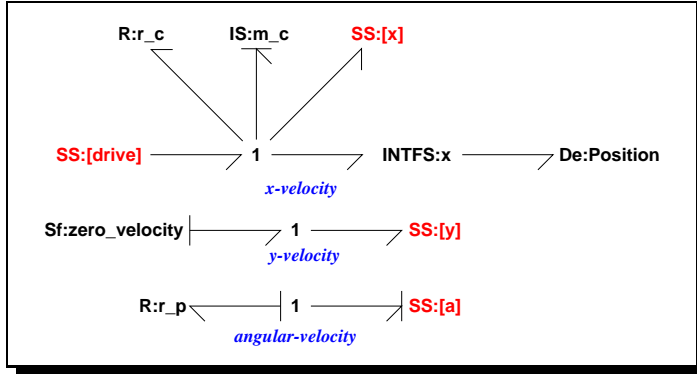
Fig. 3. The inverted pendulum on a cart

and, as discussed by Gawthrop and Smith (1996), this leads to constrained-state equations – in this case corresponding to the system *inertia matrix*.

The corresponding sensitivity system has 10 states as was automatically generated from the system bond graph using the software MTT (MTT, 2000) which also generated code in the C language to be compiled and executed ready for generating the sensitivity functions to be used in the optimisations and simulations.



(a) Drive subsystem



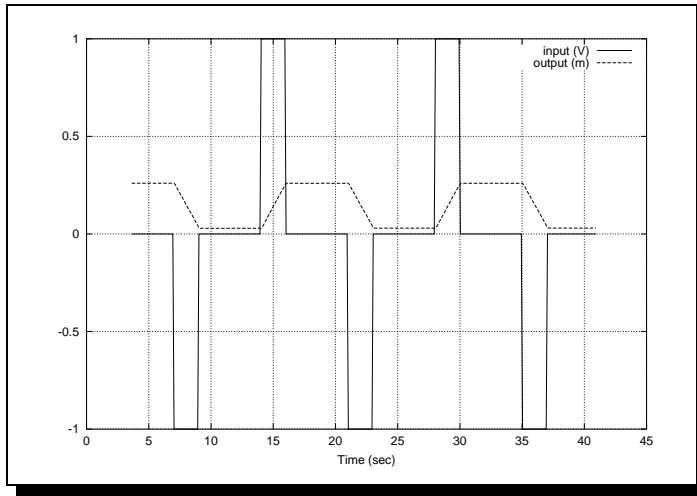
(b) Cart subsystem

Fig. 4. Subsystems: word bond graph

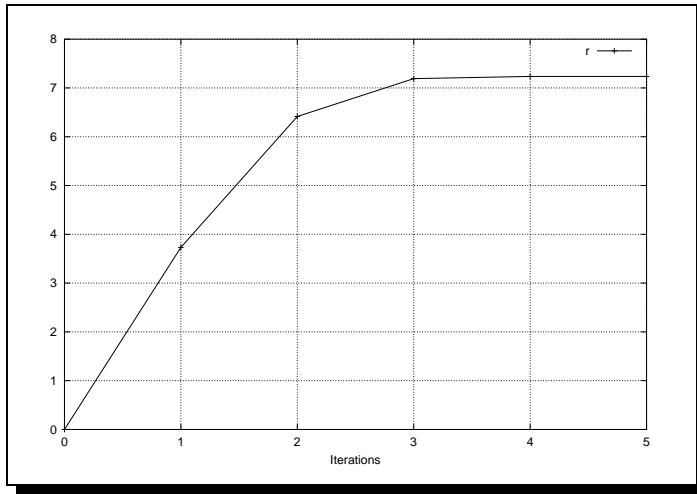
The algorithm of Table 1 was coded in Octave (Octave, 1999) and used to generate the figures using a Toshiba laptop running GNU/Linux (GNU, 1999).

## 5.2 Friction estimation

It appears from the experiments reported here that a linear friction model suffices. However, the method can be readily extended to more sophisticated friction models, for example those of Hirshorn and Miller (1999) and Canudas de Wit *et al.* (1995). Within this context, there are two unknown friction coefficients  $f_a$  and  $f_c$ . As far as the system output is concerned, these two friction coefficient may be replaced by 0 and  $r_c = f_c + \frac{1}{(rk_g)^2} f_a$ . Thus, for the purposes of control, only the single coefficient  $r_c$  need be measured.



(a) Data



(b) Estimated  $r_c$

Fig. 5. Estimating the cart friction

Input output data was gathered for the cart with the pendulum removed. Figure 5(a) shows the system input (V) and the system output (m) plotted against time (sec). This data was used in the algorithm of Section 4.2 and both  $r_c$  and the three initial system states were estimated. Figure 5(b) shows the estimated value of  $r_c$  plotted against the iteration number. Convergence takes about three iterations. Note that, in this context, that the value of the initial system states are not of any interest. However, it is necessary to estimate them



to fit the data correctly. The estimated value of  $r_c$  is

$$\hat{r}_c = 7.198 \text{Nm}^{-1}\text{s} \quad (37)$$

Two advantages of the physically based estimation approach are illustrated by this experiment. Estimation can take place on *part* of the overall system, and the known physical data is fully used.

### 5.3 Predictive control

Name	Symbol	Value
Input time constant	$T_U$	0.02 sec
Number of basis functions	$n_U$	4
Open-loop interval	$T_{ol}$	0.1 sec
Sample interval	$T_s$	0.005 sec
Lower optimisation horizon	$\tau_1$	0.8 sec
Upper optimisation horizon	$\tau_2$	1.0 sec
Angle setpoint	$w_\alpha$	0.0
Position setpoint	$w_X$	0.0

Table 3

Controller parameters

A number of simulations of the model of Section 5.1, using the parameters of Table 2 together with Equation 37, were made to verify the properties of the algorithm. The basic controller parameters are (unless otherwise stated) given in Table 3. The corresponding basis functions  $U^*(\tau)$  were the first four

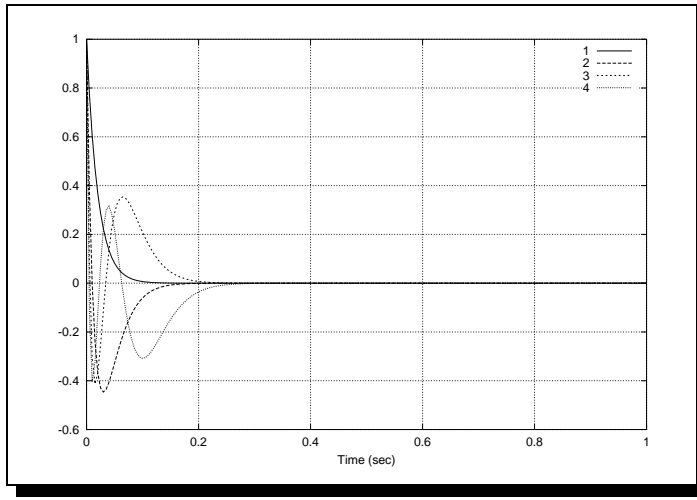


Fig. 6. The basis functions  $U^*$

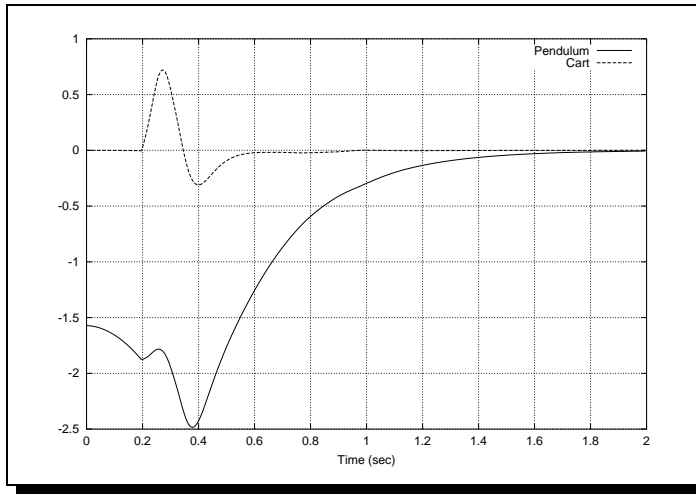
(unnormalised) Laguerre functions of Figure 6 corresponding to

$$A_u = -\frac{1}{T_U} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 \\ 2 & 2 & 2 & 1 \end{pmatrix} \quad (38)$$

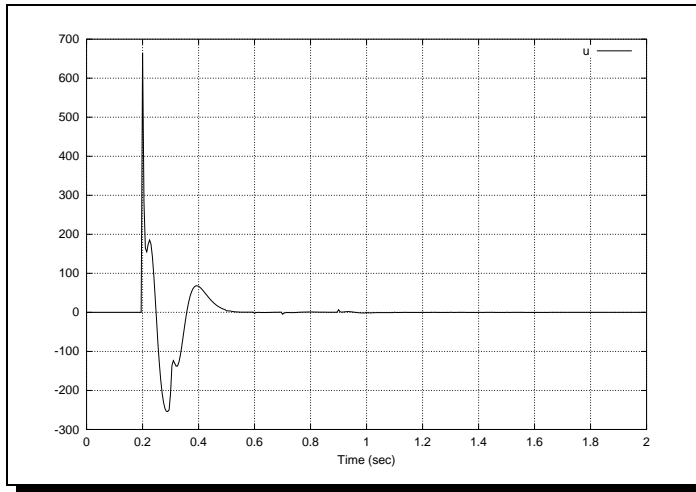
where  $T_U$  is given in Table 3. In all cases, regulation from an initial horizontal pendulum position to a vertical pendulum position with an incorrect initial state estimate is used; this is a demanding control problem.

The results are summarised as follows

**Figure 7** Part (a) of the Figure shows the two system outputs ( $X$  and  $\alpha$ ), and part (b) the corresponding control signal (motor voltage). The initial angle of the inverted pendulum is  $\alpha = -\frac{\pi}{2}$  (pendulum horizontal); but the model initially assumes  $\alpha = 0$ . For the latter reason, no control action is taken during the initial control interval  $t < T_{ol} = 0.2$  and the pendulum drops further. Based on this data, and in the absence of measurement noise, the state estimator obtains an accurate estimate of the state which is used as



(a) Pendulum and Cart positions

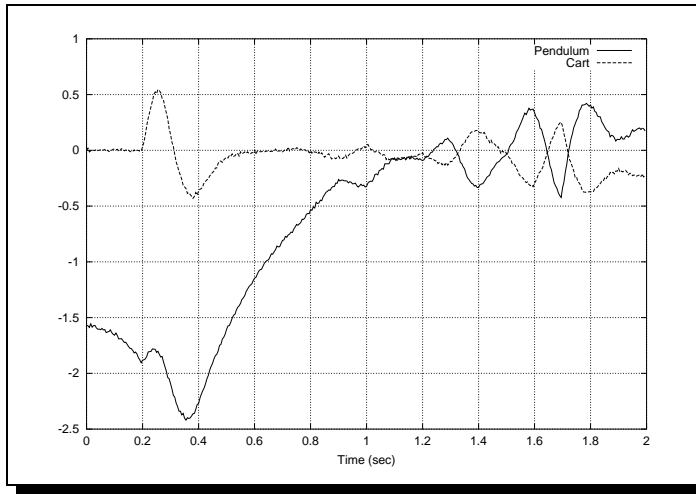


(b) Control

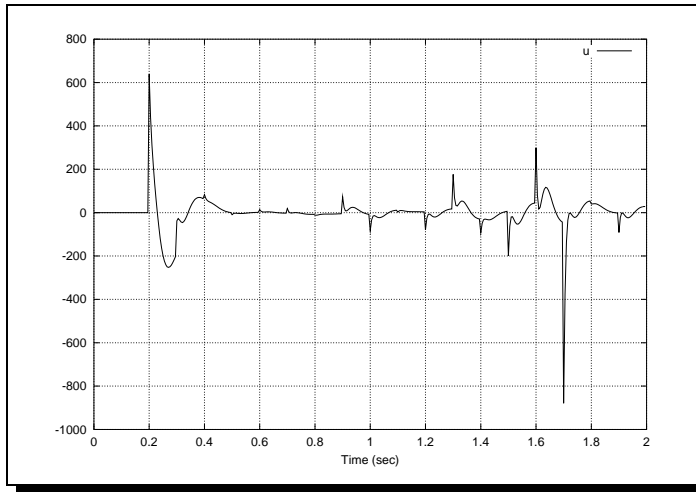
Fig. 7. Regulation from  $\theta = \frac{\pi}{2}$

the basis for the subsequent control actions. At the end of the simulation, both outputs are close to the setpoints of zero (pendulum vertical, centre track). The control signal is considerably outside the specified maximum of  $\pm 10V$ ; this is indicative of the fact noted above that the system design is unsuitable for this large motion.

**Figure 8** is the same as Figure 7 except that Gaussian white measurement



(a) Pendulum and Cart positions

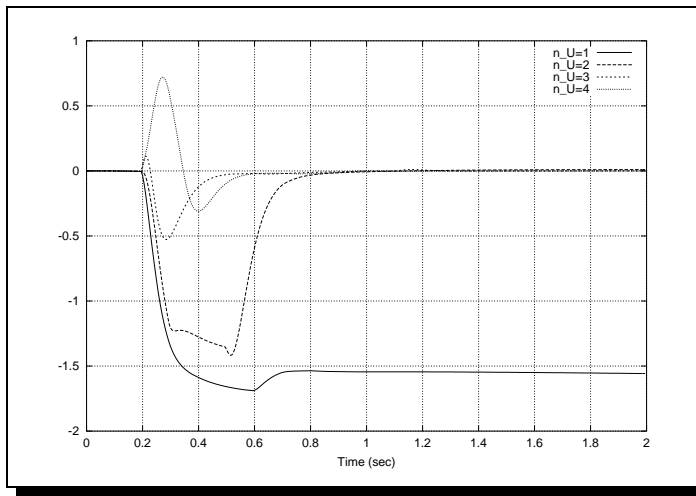


(b) Control

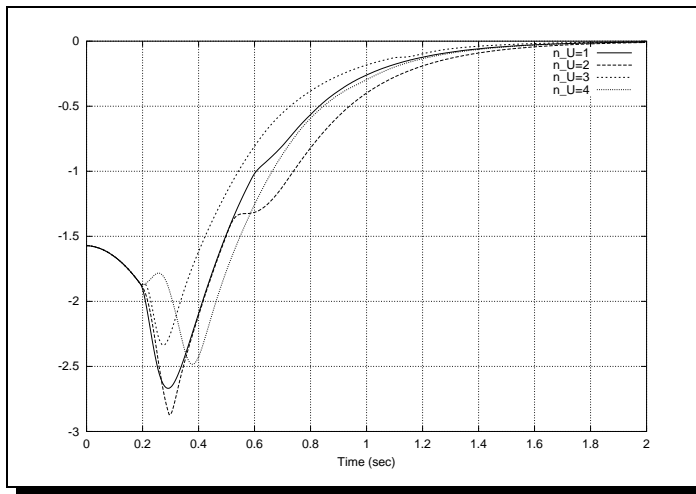
Fig. 8. Regulation from  $\theta = \frac{\pi}{2}$  with noise

noise of standard deviation 0.01 is added to each output. This causes some deviation from the ideal situation, but a stable result is still achieved.

**Figure 9** illustrates the same simulation as Figure 7 but with four different values of  $n_U = 1..4$ . For clarity, the Cart position is displayed separately (part (a)) from the Pendulum position (part (b)).  $n_U = 4$  corresponds exactly to the situation of Figure 7, the other three plots display the effect



(a) Cart position



(b) Pendulum position

Fig. 9. Effect of number of basis functions  $n_U$  of reducing the number of basis functions. Roughly speaking,  $n_U = 3$  gives little loss in performance, whilst smaller values of  $n_U$  are unsatisfactory.

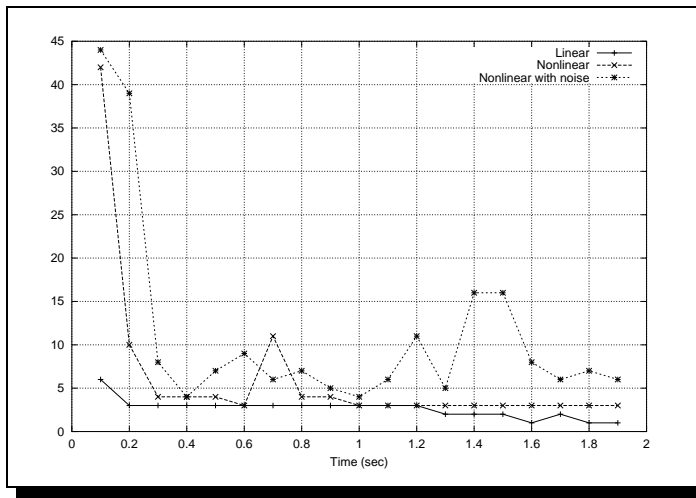


Fig. 10. Optimisation iterations

**Figure 10** shows the number of optimisation iterations required to evaluate the control weights  $U(t)$  at each step. Three graphs are shown. Those marked “Nonlinear” and “Nonlinear with noise” correspond to Figures 7 and 8 respectively, that marked “Linear” corresponds to a simulation identical to that of Figure 7 except that the initial angle is  $\alpha = -\frac{\pi}{100}$  and thus the system is essentially linear. Except for the first iteration, the linear case requires the minimum of 2 iterations indicating a linear/quadratic optimisation. The nonlinear case requires over forty iterations for the first non-trivial optimisation ( $t = 0.2$ ) but thereafter requires less than 10 indicating that the trajectory continuation method of Lemma 1 is effective. The noisy case requires more iterations as the estimated state and predicted states are different.

## 6 Conclusion

It has been shown that bond graph models of dynamic systems in general, and mechatronic systems in particular, can be used to generate sensitivity information in a form appropriate for optimisation, and that this optimisation can be used for real-time identification, state estimation and control of nonlinear

systems which have a bond graph representation.

Although the results of Section 5.2 are based on real data, because of the limitations of the equipment it was not possible to perform the swing-up experiment. However, the simulations of Section 5.3 were based on the verified model and are therefore believed to provide evidence of the applicability of this approach.

Making use of appropriate software tools, this approach provides a powerful way of facilitating the design of mechatronic and other control systems. It is planned to extend the software using the real-time Linux kernel to provide the seamless environment mentioned in the Introduction.

Future work will include extension of the analysis of the linear PPP algorithm to the nonlinear case; performing experiments on real mechatronic systems and experimental comparison with other approaches.

## **7 Acknowledgements**

This work was accomplished whilst the first author was a visitor at the Centre for Integrated Dynamics and Control, University of Newcastle, New South Wales; he would like to thank Prof. Graham Goodwin for providing an excellent work environment. Dr Will Heath of the University of Newcastle and Dr Tomas McKelvey of the University of Linköping provided insights into optimisation methods. He would also like to thank Prof. David Murray-Smith of Glasgow University for introducing me to sensitivity methods and Prof. Job Van Amerongen of the University of Twente, for helpful discussions.

Both authors would like to thank Prof. David Hill of the University of Sydney for providing the experimental facilities.

EPSRC support this work through the grant Physically-orientated nonlinear control.

## References

- An, C. H., C. G. Atkeson and J. M. Hollerbach (1988). *Model-based Control of Robot Manipulators*. The MIT Press.
- Apkarian, Jacob (1995). *A Comprehensive and Modular Laboratory for Control Systems Design and Implementation*. Quanser Consulting.
- Ascend (1999). Ascend:  
A flexible modelling environment for hard engineering and science problems.  
Online WWW Home Page. URL: <http://www.cs.cmu.edu/~ascend/>.
- Cabanellas, J. M., J. Felez and C. Vera (1995). A formulation of the sensitivity analysis for dynamic systems optimisation based on pseudo bond graphs. In: Cellier and Granda (1995). pp. 135–144.
- Calahan, D.A. (1972). *Computer-aided Network Design*. McGraw-Hill. New York.
- Canudas de Wit, C. (1988). *Adaptive control for partially known systems*. Elsevier. Amsterdam.
- Canudas de Wit, C., H. Olsson, K. J. Astrom and P. Lischinsky (1995). A new model for control of systems with friction. *IEEE Trans. on Automatic Control* **40**(3), 419–425.
- Cellier, F. E. (1991). *Continuous system modelling*. Springer-Verlag.
- Cellier, F. E. and Granda, J. J., Eds.) (1995). *Proceedings of the 1995 International Conference On Bond Graph Modeling and Simulation (ICBGM'95)*. Vol. 27 of *Simulation Series*. Society for Computer Simulation. Las Vegas, U.S.A.
- Chen, H. and F. Allgöwer (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica* **34**(10), 1205–1217.



- Clarke, D. W. (1994). *Advances in Model-based Predictive Control*. Oxford University Press.
- Dasgupta, S., B. D. O. Anderson and R. J. Kaye (1986). Output error identification methods for partially known systems. *Int. J. Control* **43**(1), 177–191.
- Demircioglu, H. and P. J. Gawthrop (1991). Continuous-time generalised predictive control. *Automatica* **27**(1), 55–74.
- Eykhoff, P. (1974). *System Identification*. Wiley.
- Fletcher, R. (1987). *Practical Methods of Optimization. 2nd Edition*. Wiley. Chichester.
- Frank, Paul M. (1978). *Introduction to System Sensitivity Theory*. Academic Press. New York.
- Gawthrop, P. J. (1995a). Bicausal bond graphs. In: Cellier and Granda (1995). pp. 83–88.
- Gawthrop, P. J. (1995b). Physical model-based control: A bond graph approach. *Journal of the Franklin Institute*.
- Gawthrop, P. J. and L. P. S. Smith (1996). *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall. Hemel Hempstead, Herts, England.
- Gawthrop, P. J., H. Demircioglu and I. Siller-Alcala (1998). Multivariable continuous-time generalised predictive control: A state-space approach to linear and nonlinear systems. *Proc. IEE Pt. D: Control Theory and Applications* **145**(3), 241–250.
- Gawthrop, P. J., J. Ježek, R. W. Jones and I. Sroka (1993). Grey-box model identification. *Control-Theory and Advanced Technology*.
- Gawthrop, P. J., R. W. Jones and S. A. MacKenzie (1992). Identification of partially-known systems. *Automatica* **28**(4), 831–836.
- Gawthrop, Peter J (1999). Sensitivity bond graphs. *Journal of the Franklin Institute*. (Submitted).

- Gawthrop, Peter J. and Eric Ronco (1999). Predictive pole-placement control with linear models. *Automatica*. (Submitted).
- GNU (1999). Gnu home page. Online WWW. URL: <http://www.gnu.org>.
- Hirshorn, R.M. and G. Miller (1999). Control of nonlinear systems with friction. *CST* **7**(5), 588–595.
- Karnopp, D. C. (1969). Power-conserving transformations: Physical interpretations and applications using bond graphs. *J. Franklin Institute* **288**(3), 175–201.
- Karnopp, D. C. (1979). Bond graphs in control: Physical state variables and observers. *J. Franklin Institute* **308**(3), 221–234.
- Karnopp, D. C. (1995). Actively controlled systems: – an ideal application area for bond graph modelling (plenary address). In: Cellier and Granda (1995). p. 3.
- Karnopp, D. C., D. L. Margolis and R. C. Rosenberg (1990). *System Dynamics: A Unified Approach*. John Wiley.
- Karnopp, Dean (1977). Power and energy in linearised systems. *Journal of the Franklin Institute* **303**(1), 85–98.
- Kouvaritakis, B., M. Cannon and J.A. Rossiter (1999). Non-linear model-based predictive control. *Int. J. Control* **72**(10), 919–928.
- Modelica (2000). Modelica: Language design for multi-domain modeling. Online WWW Home Page. URL: <http://www.modelica.org/>.
- MTT (2000). MTT: Model transformation tools. Online WWW Home Page. URL: <http://www.eng.gla.ac.uk/~peterg/software/MTT/>.
- Muske, K. R. and J. B. Rawlings (1993). Model predictive control with linear models. *Process Systems Engineering* **39**(2), 262–287.
- Octave (1999). Octave home page. Online WWW. URL: <http://www.che.wisc.edu/octave/>.
- Oppen, I., Mingrui Gong and D. J. Murray-Smith (1995). The development of tuning techniques for single-input single-output and multivariable control systems

- using controller sensitivity measures. In: *UKSS'95 - Second Conference of the UK Simulation Society* (R.C.H. Cheng and R.J. Pooley, Eds.). UKSS. Edinburgh, U.K.. pp. 73–79.
- Press, W.H., S.A. Teukolsky, W.T. Vetterling and B.P. Flannery (1992). *Numerical Recipes in C*. 2nd ed.. Cambridge University Press.
- Ronco, E., T. Arsan and P. J. Gawthrop (1999). Open-loop intermittent feedback control: Practical continuous-time GPC. *IEE Proceedings Part D: Control Theory and Applications* **146**(5), 426–434.
- Ronco, Eric and Peter J. Gawthrop (1999). Symbolic quasi-Newton optimisation for system identification. In: *Proceedings of the UKACC conference "Control 2000"*. Cambridge, U.K.
- Thoma, J. U. (1990). *Simulation by bond graphs*. Springer-Verlag. Berlin.
- Tomović, R. and M. Vukobratović (1972). *General Sensitivity Theory*. number 35 In: *Modern analytic and computational methods in science and mathematics*. Elsevier. New York.
- Van Amerongen, J. and A. J. Udink ten Cate (1975). Model reference autopilots for ships. *Automatica* **11**, 441–449.
- Winning, D.J., E.H.T. El-Shirbeeney, E.C. Thompson and D.J. Murray-Smith (1977). Sensitivity method for online optimisation of a synchronous generator excitation controller. *Proceedings IEE Part D: Control Theory and Applications* **124**(7), 631–638.