CONTROLO 2008
8th Portuguese Conference on Automatic Control
University of Trás-os-Montes and Alto Douro, Vila Real, Portugal
July 21-23, 2008

DETERMINING FLAT OUTPUTS OF MIMO NONLINEAR SYSTEMS USING DIRECTED GRAPHS

Dávid Csercsik [*] **Gábor Szederkényi** [*,1] **Katalin M. Hangos** [*,2]

[*] *Process Control Research Group, Computer and Automation Research Institute HAS H-1518 Budapest, P.O. Box 63, Hungary, csercsik@scl.sztaki.hu*

Abstract: Graph theoretic methods for flatness analysis of MIMO systems are proposed in this paper. Necessary and sufficient conditions for flatness are formulated, and an algorithm is described for the explicit expression of the state variables and inputs from the flat outputs. For this purpose, some new concepts and constructions are defined. In all cases, structural flatness properties are examined (i.e. only the appearance and expressibility of the state variables are critical for the analysis).
The method is demonstrated on a simplified nonlinear model of a the primary circuit dynamics of a pressurized water nuclear reactor (PWR).

Keywords: Linear and nonlinear control, control theory, process control, control applications, hybrid systems

1. INTRODUCTION

Finding flat outputs for a nonlinear system gives us valuable support in state estimation and controller design, therefore this is an extensively studied area of modern systems and control theory. A good introductory theory about flatness-based control applications can be found in (Fliess *et al.*, 1994) and in (Rathinam, 1997). Further computation techniques of state and input trajectories for flat systems using automatic differentiation is discussed in (Robenback and Vogel, 2004). Necessary and sufficient algebraic conditions for the flatness of four-dimensional systems are given in (Pomet, 1996). Trajectory generation for differentially flat systems with inequality constrains is detailed in (Faiz *et al.*, 2001). Applications of the flatness property to mechanical systems are described in (Kiss, 1998).

Previously, bond graphs extended with the bicausality concept proved to be useful in finding flat outputs in nonlinear MIMO systems (Richard *et al.*, 2002) related to dynamic inversion (Gawthrop, 2000). However, bond graph description of dynamical systems is heavily based on the identification of important physical processes (particularly energy exchange) of the modeled system (Broenink, 1999). Our proposed approach is substantially different from this, because it relies only on the algebraic structure and properties of the system equations. Hence, our algorithm is usable with such models where the physical meaning is not transparent (or completely lost e.g. because of coordinates transformations, embeddings etc.).

We will consider the class of general nonlinear systems in the following form:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t)) \end{aligned}, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input and $y \in \mathbb{R}^m$ is the output. A system (1) is called (differentially) flat if the state and input variables can be expressed as functions of the outputs

[1] Gábor Szederkényi is a grantee of the Bolyai János Research Scholarship of the Hungarian Academy of Sciences
[2] Partially supported by the Hungarian Research Fund OTKA through grant K67625 and by the Control Engineering Research Group of BUTE

and their derivatives, i.e.:
$x = A(y, \dot{y}, \ddot{y}, ..., y^{\alpha}), \;\; u = B(y, \dot{y}, \ddot{y}, ..., y^{\alpha+1})$.

In this paper our aim is to identify a special subclass of flat systems, with the so called *global explicit expressibility property*. This property means, that we can get closed and globally valid explicit formulas for the functions A and B.

For this aim, we will define some graphs to formulate necessary and sufficient conditions for flatness using the theory of directed graphs, and define a method for automatic check of explicit expressible flatness in the case of suggested flat outputs.

2. THE GRAPH THEORETICAL ALGORITHM

In this part, we define the basic concepts and notations related to the proposed graph-theoretical algorithm for finding flat outputs to a MIMO nonlinear system.

Throughout the paper **general MIMO square systems are only considered, where** $r = dim(u) = dim(y) < n$ **with** n **being the number of state variables. Furthermore, we will discuss only the simplest case, when the flat outputs are considered as state variables.**

2.1 *Representation graphs*

The algebraic structure of the system is described by the **structure graph** which is defined as follows:

- The vertices of the graph are the state-variables, the input variables, and the output variables.
- A directed path leads from vertex V_1 to vertex V_2 if and only if the variable V_2 depends on V_1 (If V_2 is a state-variable, this means that V_1 can be found in the state-equation describing the time derivative of V_2. An output variable depends on a state variable if and only if the state variable can be found in it's output equation).

The structure graph shows which variables are influenced by a state or input. If we take the same graph with reversed path directions, we will get information about which outputs or states depend on other states and inputs. We'll call this graph in the following the **dependency graph**.

If we suppose that some state variables and derivatives are known, and we want to express some other state variables from a state equation, the algebraic form of the state equation determines whether we can get an explicit expression of the desired variable or not. To characterize such properties of a state space model, we define the **explicit expressibility graph** in the following way:

- The vertices of the graph are the state-variables, the input variables, and the output variables.
- A directed path leads form vertex V_1 to vertex V_2 if and only if V_2 can be expressed explicitly from the differential equation describing V_1.

Remarks. (i) Loop edges will be discarded during the algorithm, because our aim here will be always to express a next state as a function of already expressed *other* states. But the loop edges influence neither the states' dependency on each other nor the properties related to explicit expressibility. (ii) It is important to note that *causality is implicitly present* in the structure graphs, and thus in the dependency and explicit expressibility graphs, too. As a directed path from vertex V_1 to vertex V_2 in a structure graph is interpreted as the variable V_2 depends on V_1, then the change in variable V_1 causes a change in V_2.

Problem instance. The proposed method can be applied to check whether a given set of candidate outputs is flat for a given nonlinear state space model described by its explicit expressibility graph. A problem instance is thus composed of an *explicit expressibility graph and a set of candidate flat outputs.*

2.2 *Necessary conditions for flatness*

The necessary conditions below are checked first to sort out problem instances for which either the representation graphs or the set of candidate flat outputs are not suitable.

(*Lemma 1*) In the case of explicit expression of states, the maximal number of state-variables which have no ingoing edge in the explicit expressibility graph is $(r - 1)$.

(*Proof*) Such states can only be expressed if an output is defined for them. If there were $r = dim(u)$=$dim(y)$ number of such states, either one of them or the remaining states (which surely present because $r < n$) couldn't be expressed.

(*Lemma 2*) For a possible set of flat outputs $x_{i..j}$ all states and inputs have to be reachable from $x_{i..j}$ in the dependency graph.

(*Proof*) If a state or input is not reachable from the possible flat outputs' vertices, it will not appear after any number of derivation of the outputs.

Theorem 1 In the case of m inputs, for a possible set of flat output-input pairs, m pairwise disjoint paths must exist in the dependency graph, the union of which covers each vertex of the dependency graph.

(*Proof*) If no paths exist at all that cover the entire dependency graph, there exists at least a state which will not appear in any derivatives of the outputs. If they are not pairwise disjoint, the sub-graph depicted in figure 1 can be found in the graph defined by the paths.

In this sub-graph, the paths of the explicit expressibility graph also have to appear (we can also suppose that

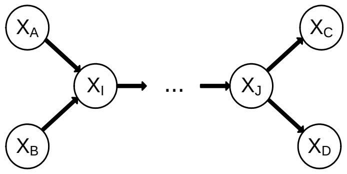

Fig. 1. Sub-graph in the case of non pairwise disjoint paths

the explicit expressibility graph and the dependency graph are equivalent in this case). In this case, x_I can be expressed in two ways, this means we get redundant information for it. In other words this implies a correlation between the derivatives of the flat outputs. In fact at x_J the following problem appears for both output derivatives:

Consider that we have chosen a state as a possible flat output. As we take the derivative of the output, according to the dependency graph, we can immediately identify which state space variables will surely appear in the derivative's equation. If only one currently unexpressed state variable appears, and we can express it, we can go on to the further derivatives of the output.

At x_J, (at least) two unknown (previously unexpressed) variables appear in the equation. In the derivation thereafter, at least one unknown state variable or input appears in the equations in each step, and so we will have always less equations than unknown variables to express.

2.3 *The step-by-step construction algorithm*

Given a problem instance, the initial step of the flat model construction algorithm is to check if the necessary conditions, i.e. the conditions of *Lemma 1 and 2*, and *Theorem 1* are fulfilled. The latter implies to have m pairwise disjoint paths that together cover the entire expressibility graph.

The next steps of the algorithm explore step by step the expressions for the state and input variables that constitute the flat model by walking systematically through the path system. Let us denote the set of all state-variables and inputs by X, and the set of already expressed variables at any step of the algorithm by X_E. For the sake of simplicity, the inputs are treated like states, that have to be expressed.

One step of the algorithm At each step, we differentiate $k \leq m$ unexplored outputs. Suppose that before the differentiation we have $X_E \subseteq X$ variables which have already been expressed in the previous steps. Because of the pairwise disjoint paths in the explicit expressibility graph, at least k new explicitly expressible variables ($X_N \doteq \{x_{N1}...x_{Nk}\}$) appear in the new equations. If all of the other variables on the right side of the new equations belong to X_E, then it is possible that $x_{N1}...x_{Nk}$ can be expressed from the equations. In the expressible case $X_E^{new} = X_E \bigcup \{x_{N1}...x_{Nk}\}$, otherwise the step is unsuccessful.

If not all of the other variables on the right side of the new equation belong to X_E then we have less equations than unknown variables to express, so the step is unsuccessful.

If an input's derivative appears in the new equation, then either the step is unsuccessful or we must define new states (pre-compensators) for the corresponding input.

A next step can follow after a successful step, otherwise one should repeat the unsuccessful step by choosing another k outputs to explore.

The above described step-by-step algorithm gives rise to the sufficient condition for a system to be flat formulated in a theorem below.

Theorem 2 If there exists a set of pairwise disjoint output-input pathways in the explicit expressibility graph, which cover the entire dependency graph's vertices, and the pathways can be 'walked through' with the step-by-step method described above, then the system is flat, and the state-variables can be expressed in an explicit way as functions of the flat outputs and their time derivatives.

3. CASE STUDY: PRIMARY CIRCUIT OF A NUCLEAR PLANT

Consider the nonlinear hybrid model of a PWR (VVER-type) nuclear power plant's primary circuit described in (Fazekas *et al.*, 2006). The state space model, derived from energy and mass balances, and engineering principles is the following:

$$
\begin{aligned}
\frac{dN}{dt} &= \frac{\beta}{\Lambda}(p_1 v^2 + p_2 v + p_3)N + S \\
\frac{dM_{PC}}{dt} &= m_{in} - m_{out} \\
\frac{dT_{PC}}{dt} &= \frac{1}{c_{P,PC}M_{PC}}[c_{P,PC}m_{in}(T_{PC,I} - T_{PC}) + c_{\Psi 1}N \\
&\quad + c_{P,PC}m_{out}15 - 6K_{T,SG}(T_{PC} - T_{SG}) \\
&\quad - W_{loss,PC}(T_{PC} - T_{env})] \\
\frac{dT_{PR}}{dt} &= \frac{1}{c_{P,PR}M_{PR}}(\chi_{m_{PR}>0}c_{P,PC}m_{PR}T_{PC,HL} + \\
&\quad + \chi_{m_{PR}<0}c_{P,PR}M_{PR}T_{PR} - c_{P,PR}M_{PR}T_{PR} - \\
&\quad - W_{loss,PR} + W_{heat,PR}) \\
\frac{dT_{SG}}{dt} &= \frac{1}{c^L_{P,SG}M_{SG}}(c^L_{P,SG}m_{SG}T_{SG,SW} - W_{loss,SG} - \\
&\quad - c^V_{P,SG}m_{SG}T_{SG} - m_{SG}E_{evap,SG} + \\
&\quad + K_{T,SG}(T_{PC} - TSG))
\end{aligned} \tag{2}
$$

where N denotes the neutron flux, M_{PC} is the mass of the primary circuit water, T_{PC} is the primary circuit water temperature, T_{PR} is the pressurizer temperature and T_{SG} is the temperature in the steam generator. The hybrid behaviour of the pressurizer dynamics is represented using a characteristic function in the followng way. The value of the expression $\chi_{m_{PR}<0}$ is 1 if $m_{PR} < 0$, and 0 if $m_{PR} \geq 0$. The operational principles and basic mathematical models of nuclear reactors can be found e.g. in (Kessler, 1983).

To shorten and simplify the above equations, the notations in the following tables will be used

State space variables	
N	x_1
M_{PC}	x_2
T_{PC}	x_3
T_{PR}	x_4
T_{SG}	x_5

Inputs	
v	u_1
m_{in}	u_2
$W_{heat,PR}$	u_3

Disturbances	
m_{out}	d_1
m_{SG}	d_2
M_{SG}	d_3
$T_{SG,SW}$	d_4
$T_{PC,I}$	d_5

Constants	
$c_{P,PC}$	c_1
$c_{P,PR}$	c_2
$c^L_{P,SG}$	c_3
$c^V_{P,SG}$	c_4
$K_{T,SG}$	K_1
$E_{evap,SG}$	E_1
$c_{\Psi 1}$	$c_{\Psi 1}$
S	S
$W_{loss,PR}$	W_2
$W_{loss,SG}$	W_3

Constants	
$V_{PC,0}$	V_1
β	β
Λ	Λ
	a_2
	a_3
	a_4
	a_5
p_1	p_1
p_2	p_2
p_3	p_3

Functions	
$M_{PR}(M_{PC},T_{PC}) = M_{PC}$ $-\rho(T_{PC})V_{PC,0}$	$M_1(x_2,x_3)$
$\rho(T_{PC}) = a_1T_{PC}^2 + a_2T_{PC} + a_3$	$\rho(x_3)$
$W_{loss,PC}(T_{PC}) = a_4T_{PC} + a_5$	$W_1(x_3)$
$T_{PC,HL}(T_{PC}) = T_{PC} + 15$	$T_1(x_3)$
$m_{PR} = \frac{dM_{PR}}{dt} = \frac{dM_{PC}}{dt}$ $-\frac{d\rho(T_{PC})}{dT_{PC}}\frac{dT_{PC}}{dt}V_{PC,0}$	$m_1(x_1,x_2,x_3,x_5,)$ $u_2,d_1,d_5)$

3.1 *Simplified dynamics*

We can derive the equations of the simplified dynamics, if we substitute $\chi_{m_{PR}>0} = 0$ and $\chi_{m_{PR}<0} = 1$ to the original state space equations (2), and use the notations defined above.

$$
\begin{aligned}
\frac{dx_1}{dt} &= \frac{\beta}{\Lambda}(p_1u_1^2 + p_2u_1 + p_3)x_1 + S \\
\frac{dx_2}{dt} &= u_2 - d_1 \\
\frac{dx3}{dt} &= \frac{1}{c_1x_2}[c_1u_2(d_5 - x_3) + c_1d_1 15 + c_{\Psi 1}x_1 \\
&\quad -6K_1(x_3 - x_5) - W_1(x_3)] \\
\frac{dx_4}{dt} &= \frac{1}{c_2M_1(x_2,x_3)}[-W_2 + u_3] \\
\frac{dx_5}{dt} &= \frac{c_3d_2d_4 - c_4d_2x_5 - d_2E_1 + K_1(x_3 - x_5) - W_3}{c_3d_3}
\end{aligned}
$$

We assume that we only utilize inputs u_1 and u_2 in this setup. For this case, the dependency graph can be seen in Fig 2. The explicit expressibility graph is depicted in Fig. 3 (loop-edges and disturbances are neglected).
Outputs are chosen as follows: $y_1 = x_4$, $y_2 = x_5$. The pairwise disjoint paths are denoted by bold arrows.

Expression of variables. In this case, u_1 can not be directly explicitly expressed from x_1, but this problem can be solved as follows.

- In the beginning $X_E^1 = \{x_4\}$ $X_E^2 = \{x_5\}$ $(X_E = \{X_E^1 \bigcup X_E^2\})$.

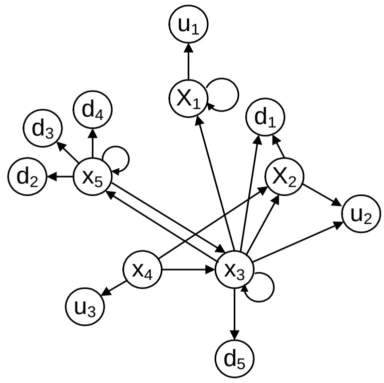

Fig. 2. The dependency graph in the simple case

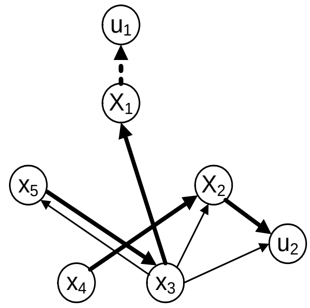

Fig. 3. The explicit expressibility graph in the simple case

- $\dot{y_2} = \dot{x_5} = f(x_3, x_5) \rightarrow x_3$ can be expressed $\rightarrow$ $X_E^1 = \{x_4\}$ $X_E^2 = \{x_3, x_5\}$
- $\dot{y_1} = \dot{x_4} = f(x_2, x_3) \rightarrow x_3$ is already expressed $\rightarrow x_2$ can be expressed $\rightarrow X_E^1 = \{x_2, x_3, x_4\}$ $X_E^2 = \{x_3, x_5\}$
- $\ddot{y_2} = f(\dot{x_3}, x_5) = f(x_1, x_2, x_3, x_5, u_2)$ as it can be seen from the dependency graph. x_1 and u_2 are the new variables.
- $\ddot{y_1} = f(\dot{x_2}, x_3) = f(x_1, x_2, x_3, x_5, u_2)$ as it can be seen from the dependency graph. x_1 and u_2 are the new variables.

At this point, we have 2 *independent* equations for two unknown variables that can be solved in this case.

The detailed derivation of the above steps are as follows.

$$y_1 = x_4, \quad y_2 = x_5$$

$$\dot{y}_2 = \frac{1}{c_3d_3}(c_3d_2d_4 - c_4d_2x_5 - d_2E_1 + K_1(x_3 - x_5) - W_3)$$

$$x_3 = \frac{c_4d_2x_5 - c_3d_2d_4 + d_2E_1 + K_1x_5 + W_3 + \dot{y_2}c_3d_3}{K_1}$$

$$\dot{y}_1 = \frac{-W_2}{c_2M_1(x_2,x_3)} = \frac{-W_2}{c_2(x_2 - (a_1x_3^2 + a_2x_3 + a_3)V_1)}$$

$$
\begin{aligned}
x_2 &= \frac{-W_2}{\dot{y}_1c_2} + (a_1x_3^2 + a_2x_3 + a_3)V_1 \\
\ddot{y}_1 &= \ddot{x_4} = -\frac{W_2\ (\dot{x}_2 - (2\,a_1\,x_3\,\dot{x}_3 + a_2\,\dot{x}_3 + a_3)\,V_1)}{c_2\ (x_2 - (a_1\,{x_3}^2 + a_2\,x_3 + a_3)\,V_1)^2} \\
\dot{x}_2 &= u_2 - d_1 \\
u_2 &= \frac{\ddot{y}_1\,c_2\ \left(x_2 - \left(a_1\,{x_3}^2 + a_2\,x_3 + a_3\right)\,V_1\right)^2}{W_2} + \\
&\quad + (2\,a_1\,x_3\,\dot{x}_3 + a_2\,\dot{x}_3)\,V_1 + d_1
\end{aligned}
\tag{3}
$$

$$\ddot{y}_2 = \frac{1}{c_2 d_3^2}\left(-\dot{d}_3[c_3 d_2 d_4 - c_4 d_2 x_5 - d_2 E_1 + K_1(x_3 - x_5)]\right) + \frac{1}{c_2 d_3^2}\left(-\dot{d}_3 W_3 + d_3[c_3(\dot{d}_2 d_4 + \dot{d}_4 d_2) - c_4(\dot{d}_2 x_5)]\right) + \frac{1}{c_2 d_3^2}\left(+d_3[d_2 \dot{x}_5 - E_1 \dot{d}_2 + K_1(\dot{x}_3 - \dot{x}_5)]\right) \tag{4}$$

It is easy to see, that from this equation $\dot{x}_3$ can be easily expressed. We denote the expressed form of $\dot{x}_3$ by $F_1(.)$. The notation $(.)$ refers to the fact, that no new (unexpressed) variables appear in the expression. In this case u_2 can be expressed from equation (3). On the other hand:

$$\dot{x}_3 = F_1(.) = \frac{1}{c_1 x_2}[c_1 u_2(d_5 - x_3) + c_1 d_1 15] + \frac{1}{c_1 x_2}[c_{\Psi 1} x_1 - 6K_1(x_3 - x_5) - W_1(x_3)]$$

In this new equation x_1 is yet unexpressed with the derivatives of y_1 and y_2.

$$x_1 = \frac{1}{c_{\Psi 1}}(F_1(.)c_1 x_2 - c_1 u_2(d_5 - x_3) - c_1 d_1 15 + 6K_1(x_3 - x_5) + W_1(x_3) \tag{5}$$

The only one vertex of the dependency graph, which is not expressed yet is the input u_1. u_1 can not be explicitly expressed from the differential equation of x_1, but the function can be locally inverted. In fact two solutions can be expected :

- If we take the third derivative of any output, $\ddot{x}_3$ will appear, that depends on u_1 and $\dot{u}_2$ as currently unexpressed variables. If we use a pre-compensator for the input u_2, we can get a flat system.
- Because x_1 does not depend on the other state-space variables, and it appears linearly in the differential equation describing x_3, it can be taken into account as an input to the system, and use the backstepping method described in (van der Schaft, 2000) to determine u_1.

3.2 *A more complex dynamics*

We can derive the equations of the more complex part of the hybrid dynamics. For this, we substitute $\chi_{m_{PR}>0} = 1$ and $\chi_{m_{PR}<0} = 0$ in the original state space equations (2).

$$\frac{dx_1}{dt} = \frac{\beta}{\Lambda}(p_1 u_1^2 + p_2 u_1 + p_3)x_1 + S$$

$$\frac{dx_2}{dt} = u_2 - d_1$$

$$\frac{dx_3}{dt} = \frac{1}{c_1 x_2}[c_1 u_2(d_5 - x_3) + c_1 d_1 15 + c_{\Psi 1} x_1 - 6K_1(x_3 - x_5) - W_1(x_3)]$$

$$\frac{dx_4}{dt} = \frac{1}{c_2 M_1(x_2, x_3)}[c_1 m_1(x_1, x_2, x_3, x_5, u_2, d_1, d_5)T_1(x_3) - W_2 + u_3] - x_4$$

$$\frac{dx_5}{dt} = \frac{c_3 d_2 d_4 - c_4 d_2 x_5 - d_2 E_1 + K_1(x_3 - x_5) - W_3}{c_3 d_3}$$

The equations define the dependency graph that is shown in Fig. 4 if we suppose the utilization of all 3 inputs. Accordingly, 3 pairwise disjoint paths appear.

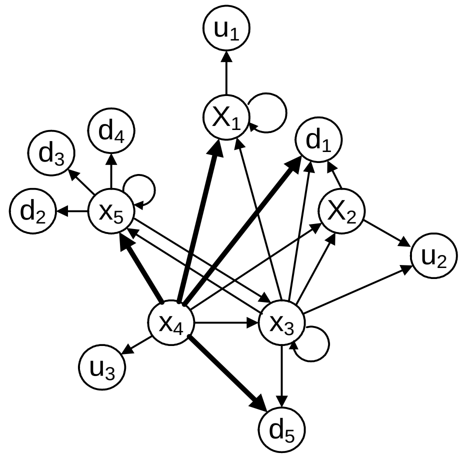

Fig. 4. The dependency graph in the complex case - the bold edges do not appear in figure2.

The explicit expressibility graph is depicted in Fig. 5 (loop-edges and external disturbances are omitted). The outputs are chosen as:

$$y_1 = x_2,\ y_2 = x_4,\ y_3 = x_5.$$

The pairwise disjoint paths are denoted by bold arrows again.

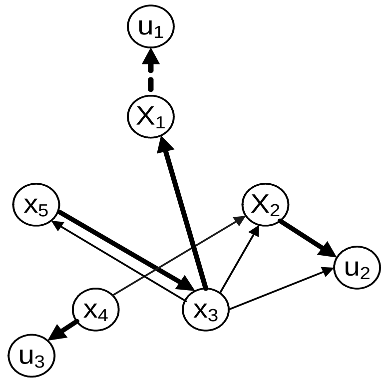

Fig. 5. The explicit expressibility graph in the complex case

Expression of variables As it can be seen from the graphs, in this case our strategy will be the following:

- From the first derivative of $y_1 = x_2$ we can express $u_2 \rightarrow X_E = \{x_2, x_4, x_5, u_2\}$.
- From the first derivative of $y_3 = x_5$ we can express $x_3 \rightarrow X_E = \{x_2, x_3, x_4, x_5, u_2\}$.
- The second derivative of $y_3 = x_5$ will be $\ddot{y_3} = f(x_1, x_2, x_3, x_5, u_2)$ as it can be seen in the dependency graph. We can express $x_1 \rightarrow X_E = \{x_1, x_2, x_3, x_4, x_5, u_2\}$.
- The first derivative of $y_2 = x_4$ will be $\dot{y_2} = f(x_1, x_2, x_3, x_5, u_2, u_3)$ as it can be seen in the dependency graph. We can express $u_3 \rightarrow X_E = \{x_1, x_2, x_3, x_4, x_5, u_2, u_3\} = X \setminus u_1$

The detailed derivation is as follows.

$$
\begin{aligned}
\dot{y}_1 &= u_2 - d_2 \;\; \rightarrow \;\; u_2 = d_2 - \dot{y}_1 \\
X_E &= \{x_2, x_4, x_5, u_2\} \\
\dot{y}_3 &= \frac{c_3 d_2 d_4 - c_4 d_2 x_5 - d_2 E_1 + K_1(x_3 - x_5) - W_3}{c_3 d_3} \\
x_3 &= \frac{(c_4 d_2 x_5 - c_3 d_2 d_4 + d_2 E_1 + K_1 x_5 + W_3 + \dot{y}_3 c_3 d_3)}{K_1} \\
X_E &= \{x_2, x_3, x_4, x_5, u_2\}
\end{aligned}
$$

It is easy to see, that from the equation

$$
\begin{aligned}
\ddot{y}_3 = \frac{1}{c_3 d_3}^2 & (((c_3 d_3)((c_3 \dot{d}_2 d_4) + (c_3 d_2 \dot{d}_4) - \\
& - (c_4 d p_2 x_5 + c_4 d_2 \dot{x}_2) - \dot{p}_2 E_1 + (K_1(\dot{x}_3 - \dot{x}_5)) - W_3) \\
& -(c_3 \dot{d}_3)(c_3 d_2 d_4 - c_4 d_2 x_5 - d_2 E_1 + K_1(x_3 - x_5) - W_3))
\end{aligned}
$$

x_1 can be easily expressed (x_1 appears linearly in $\dot{x}_2$ which appears linearly in this equation). Thus, finally we obtain

$$X_E = \{x_1, x_2, x_3, x_4, x_5, u_2\}$$

Furthermore, from the equation

$$
\begin{aligned}
\dot{y}_2 = & \frac{1}{c_2 M_1(x_2, x_3)} [c_1 m_1(x_1, x_2, x_3, x_5, u_2, d_1, d_5) T_1(x_3) \\
& - W_2 + u_3] - x_4
\end{aligned}
$$

the expression of u_3 is trivial, and all other state and input variables in the equation are in X_E

$$X_E = \{x_1, x_2, x_3, x_4, x_5, u_2, u_3\} = X \setminus u_1$$

The situation with u_1 is the same as detailed in subsection 3.1, and similar solutions can be found.

4. CONCLUSIONS

It has been shown that the flatness property with respect to the state variables that serve as simple outputs can be analyzed using a graph theoretical method for a class of MIMO nonlinear systems. For this purpose, a special directed graph, the so-called explicit expressibility graph has been introduced and some useful necessary conditions for flatness have been worked out.

An algorithm has been constructed for the expression of state variables and inputs using the properties of the explicit expressibility graph. The method is capable of proving flatness only in a subclass of nonlinear systems because of the required structural and algebraic properties.

The operation of the method has been illustrated on the simplified (hybrid) model of the primary circuit dynamics of a pressurized water nuclear power plant. Further work will be directed towards the extension of the method to the case when certain states and inputs are non-globally expressible from the equations.

ACKNOWLEDGEMENT

The first author would like to thank Bálint Kiss at the Dept. of Control Engineering and Information Technology of the Budapest University of Technology and Economics for sharing his knowledge on flat systems.

5. REFERENCES

Broenink, J.F. (1999). Introduction to physical systems modelling with bond graphs. Technical report. University of Twente, Dept. of Electrical Engineering.

Faiz, N., S. K. Agrawal and R. M. Murray (2001). Trajectory planning of differentially flat systems with dynamics and inequalities. *Journal of Guidance, Control, and Dynamics* **24**, 219–227.

Fazekas, Cs., G. Szederkényi and K.M. Hangos (2006). A simple dynamic model of the primary circuit in vver plants for controller design purposes. *Nuclear Engineering and Design* **237**, 1071–1087.

Fliess, M., J. Levine, P. Martin and P.Rouchon (1994). Flatness and defect of nonlinear systems: Introductory theory and examples. *CAS Internal report A-284*.

Gawthrop, P.J. (2000). Physical interpretation of inverse dynamics using bicausal bond graphs. *Journal of the Franklin Institute* **337**, 743–769.

Kessler, G. (1983). *Nuclear Fission Reactors*. Springer-Verlag. Wien, New York.

Kiss, B. (1998). Using flatness to control pendulum mechanical systems. In: *Proceedings of the Conference of the Latest Results in Information Technology. Technical University of Budapest*. Budapest, Hungary. pp. 36–42.

Pomet, J. B. (1996). On dynamic feedback linearization of four-dimensional affine control systems with two inputs. *Research Report of the Traitment du signal, automatique and produtcique*.

Rathinam, M. (1997). Differentialy flat nonlinear control systems. *Technical Report of the Control and Dynamical Systems Option, California Institute of Technology, Pasadena, California*.

Richard, P.Y., J. Buisson and H. Cormerais (2002). Analysis of flatness using bond graphs and bicausality. In: *IFAC 15th Triennial World Congress*. Barcelona, Spain.

Robenback, K. and O. Vogel (2004). Computation of state and input trajectories for flat systems using automatic differentiation. *Automatica* **32**, 459–464.

van der Schaft, Arjan (2000). *L2-Gain and Passivity Techniques in Nonlinear Control*. Springer.