Polynomial embedding algorithms for controllers in a behavioral framework

Trentelman, Hendrikus; Zavala Yoe, R.; Praagman, Cornelis; Zavala Yoé, 27772

Published in:
IEEE Transactions on Automatic Control

DOI:
10.1109/TAC.2007.906455

Publication date:
2007

much less. Similar to algorithms for standard MDPs, value iterations generally requires more number of iterations to converge as compared with policy iteration.

For FVI, the replacement problem meets the convergence condition: a state (three components fail at the same time) is reachable from any state under any action with a positive probability ranging between 10^{-6} and 10^{-5}. However, this probability is so small that directly using it to set the parameter "μ" for the algorithm as recommended in [4] leads unacceptable slow convergence. Therefore, a larger value "0.1" is optionally chosen by us. Such adjusted parameter setting violates the convergence condition (11) in [4] but still leads to an optimal policy after 103 iterations as shown in Table I. This reveals that, on one hand, the computing load of FVI is affected by the parameter "μ" which depends on both transition probabilities and B_u (see (11) in [4]). On the other hand, the strong convergence condition needs to be weakened as mentioned in [4]. In contrast, our algorithm does not have such difficulties on parameter selections or convergence requirements. However, our algorithm solves a set of standard MDPs which may bring extra computational requirements. Fortunately, δ_n converges fast especially when B_u and B_l are close as proved in Theorem 2 and 3. This means that we generally only solve a few numbers of standard MDPs to obtain a near optimal policy. The results summarized in Table I show that only four standard MDPs are solved with 19 total number of iterations, i.e., $n = 4$, $m = 19$ for IVI. The fast converged trajectories of δ_n are illustrated in Fig. 1.

VI. Concluding Remarks

Standard algorithms may be intractable to solve an MDP with a large state space. If the problem possesses structural features such as having a large number of uncontrollable states, our algorithm performs standard value iteration on those controllable states based on time aggregation. Compared to existing algorithms for time aggregated MDPs, our algorithm requires less storage and computation during iterations than policy iteration in [1] and converges under a much weaker assumption than that required by the value iteration algorithm in [4].

Existing algorithms for standard MDPs can be used in step 2) of the incremental optimization approach. For example, employing the R-learning algorithm (see, e.g., [5]) will result in a new R-learning algorithm for time aggregated MDPs. All the algorithms developed in this note for time aggregated MDPs are directly applicable to MDPs with fractional costs.

References

[1] X. R. Cao, Z. Y. Ren, S. Bhatnagar, M. Fu, and S. Marcus, "A time aggregation approach to Markov decision process," *Automatica*, vol. 38, pp. 929–943, 2002.

[2] R. Dekker, R. E. Wildeman, and R. Egmond, "Joint replacement in an operational planning phase," *European J. Oper. Res.*, vol. 91, pp. 74–88, 1996.

[3] M. L. Puterman, *Markov Decision Process: Discrete Stochastic Dynamic Programming*. New York: Wiley, 1994.

[4] Z. Ren and B. H. Krogh, "Markov decision processes with fractional costs," *IEEE Trans. Autom. Control*, vol. 50, pp. 646–650, 2005.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

Polynomial Embedding Algorithms for Controllers in a Behavioral Framework

H. L. Trentelman, *Member, IEEE*, R. Zavala Yoe, and C. Praagman

***Abstract*—In this correspondence, we will establish polynomial algorithms for computation of controllers in the behavioral approach to control, in particular for the computation of controllers that regularly implement a given desired behavior and for controllers that achieve pole placement and stabilization by behavioral full interconnection and partial interconnection. These synthesis problems were studied before in articles by Belur and Trentelman, Rocha and Wood, and Willems in the reference section. In the algorithms, we will apply ideas around the unimodular and stable embedding problems. The algorithms that are presented in this correspondence can be implemented by means of the Polynomial Toolbox of Matlab.**

***Index Terms*—Behavioral systems, controller design, regular implementation, stabilization and pole placement, unimodular embedding problem.**

I. Introduction

In the behavioral approach, a system is defined as a triple $\Sigma = (\mathbb{R}, \mathbb{R}^q, \mathfrak{B})$, where $\mathbb{R}$ is the time axis, $\mathbb{R}^q$ is the signal space, and the behavior $\mathfrak{B}$ is the subspace of $\mathcal{L}_1^{\rm loc}(\mathbb{R}, \mathbb{R}^q)$ (the space of all locally integrable functions from $\mathbb{R}$ to $\mathbb{R}^q$) of all solutions of a set of higher order, linear, constant coefficient differential equations. In particular, $\mathfrak{B} = \{w \in \mathcal{L}_1^{\rm loc}(\mathbb{R}, \mathbb{R}^q) | R(\mathrm{d}/\mathrm{d}t)w = 0\}$. Here, R is a real polynomial matrix with q columns, and $R(\mathrm{d}/\mathrm{d}t)w = 0$ is understood to hold in the distributional sense. Σ is called a *linear differential system*. The set of all linear differential systems with q variables is denoted by $\mathcal{L}^q$. Often, we speak about the system $\mathfrak{B} \in \mathcal{L}^q$ (instead of $\Sigma \in \mathcal{L}^q$). The representation $R(\mathrm{d}/\mathrm{d}t) = 0$ of $\mathfrak{B}$ is called a kernel representation of $\mathfrak{B}$, and we often write $\mathfrak{B} = \ker(R)$. The kernel representation is called *minimal* if R has the minimal number of rows. This holds if and only if the polynomial matrix R has full-row rank. This minimal number of rows is denoted by $\mathrm{p}(\mathfrak{B})$, and is called the *output cardinality* of $\mathfrak{B}$. It corresponds to the number of outputs in any input/output representation of $\mathfrak{B}$. For a given $\mathfrak{B} \in \mathcal{L}^q$ we denote by $\mathfrak{B}_{\rm cont}$ the largest controllable subbehavior of $\mathfrak{B}$, (see [6]). This subbehavior of $\mathfrak{B}$ is called *the controllable part* of $\mathfrak{B}$. If $\mathfrak{B} = \ker(R)$ is a minimal representation, then any factorization of R as $R = DR_1$ with D square and nonsingular and $R_1(\lambda)$ full-row rank for all λ, yields $\mathfrak{B}_{\rm cont} = \ker(R_1)$.

A polynomial p is called is called Hurwitz if its zeroes are contained in the open left half complex plane $\mathbb{C}^- := \{\lambda \in \mathbb{C} | \mathrm{Re}(\lambda) < 0\}$. A square polynomial matrix P is called Hurwitz if $\det(P)$ is Hurwitz.

Manuscript received December 13, 2005; revised January 13, 2007. Recommended by Associate Editor M. Fujita.

H. L. Trentelman is with the Institute for Mathematics and Computing Science, University of Groningen, 9700 AV Groningen, The Netherlands (e-mail: h.l.trentelman@math.rug.nl).

R. Z. Yoe is with the Instituto Tecnologico y de Estudios Superiores de Monterrey, Departamento de Ingenieria, Col. Ejidos de Tlalpan, CP. 14380, Mexico DF, Mexico (e-mail: zavalay@itesm.mx).

C. Praagman is with the Institute of Economics and Econometrics, University of Groningen, 9700 AV Groningen, The Netherlands (e-mail: c.praagman@eco.rug.nl).

Digital Object Identifier 10.1109/TAC.2007.906455

II. The Unimodular and Stable Embedding Problems

In our algorithms, at several points we need to *embed* a given polynomial matrix into a unimodular one, or one that is Hurwitz. The unimodular embedding problem was studied before, e.g., in [2], using matrix pencils. Numerical experiments have shown that the matrix pencil method in [2] is not numerically reliable, and therefore in this section we propose a new algorithm.

We will first briefly recall the unimodular and stable embedding problems, and next discuss new, simple, algorithms for unimodular and stable embedding. First we discuss the unimodular embedding problem. Let P be a polynomial matrix with k rows and q columns, where $k < q$. The unimodular embedding problem is to find a polynomial matrix Q with $q - k$ rows and q columns such that the stacked matrix $(P; Q)$ is unimodular. It is well-known that such Q exists if and only if $P(\lambda)$ has full-row rank k for every complex number λ, equivalently, the Smith form of P is equal to $[I_k\ 0]$, where I_k denotes the $k \times k$ identity matrix. Below we will describe an algorithm to compute a required Q.

Algorithm 2.1

Input: a $k \times q$ polynomial matrix $P, k < q$.

1. (column compression). Find a $q \times q$ unimodular polynomial matrix U such that $PU = [P_1\ 0]$ with P_1 full column rank.
2. Check whether P_1 is square and unimodular, if not then a required Q does not exist, otherwise go to step 3.
3. Compute Q as the solution of the polynomial equation $QU = [0\ I_{q-k}]$. Then

$$\begin{bmatrix} P \\ Q \end{bmatrix} U = \begin{bmatrix} P_1 & 0 \\ 0 & I_{q-k} \end{bmatrix}.$$

Since the matrix on the right in this equation is unimodular, the same holds for the matrix on the left. Thus Q does the job.

In a similar way an algorithm for stable embedding can be established. Again, let P be a polynomial matrix with k rows and q columns, where $k < q$. The stable embedding problem is to find a polynomial matrix Q with $q - k$ rows and q columns such that the stacked matrix $(P; Q)$ is Hurwitz. Obviously such Q exists if and only if $P(\lambda)$ has full-row rank k for every complex number λ with $\mathrm{Re}(\lambda) \geq 0$. The algorithm mimics the previous one:

Algorithm 2.2

Input: a $k \times q$ polynomial matrix $P, k < q$.

1. (column compression). Find a $q \times q$ unimodular polynomial matrix U such that $PU = [P_1\ 0]$ with P_1 full column rank.
2. Check whether P_1 is square and Hurwitz, if not then a required Q does not exist, otherwise go to step 3.
3. Compute Q as the solution of the polynomial equation $QU = [0\ I_{q-k}]$. Then

$$\begin{bmatrix} P \\ Q \end{bmatrix} U = \begin{bmatrix} P_1 & 0 \\ 0 & I_{q-k} \end{bmatrix}.$$

Since the matrix on the right in this equation is Hurwitz, the same holds for the matrix on the left. Thus Q does the job.

These algorithms can be implemented using standard tools from the Polynomial Toolbox in Matlab. The column compression uses `colred`, which actually computes the full column rank matrix P_1 in column reduced form. In fact, `colred` also computes the inverse of the unimodular matrix U, which can be used to obtain Q in step 3.

The reader might wonder why we did not use the Smith form in order to compute unimodular and stable embeddings. In fact, computation of the Smith form is also a standard command in the Polynomial Toolbox. The reason why we did not use the Smith form is that, in our numerical experiments, computations based on the Smith form did not lead to correct answers. Indeed, it was argued in [11] that in general the reduction to Smith canonical form is numerically unstable. In contrast to this, the method of column compression using `colred` performs very satisfactory.

III. Algorithms for Regular Implementability

In this section, we will establish algorithms to check whether a given desired subbehavior is regularly implementable. We will also give algorithms to compute controllers that regularly implement a given behavior.

We will first briefly review the notions of implementability and regular implementability. We first consider the full interconnection case, the case that all system variables are available for control. Let $\mathcal{P} \in \mathcal{L}^q$ be a plant behavior. A controller for $\mathcal{P}$ is a system behavior $\mathcal{C} \in \mathcal{L}^q$. The *full interconnection* of $\mathcal{P}$ and $\mathcal{C}$ is defined as the system which has the intersection $\mathcal{P} \cap \mathcal{C}$ as its behavior. This *controlled behavior* is again an element of $\mathcal{L}^q$. The full interconnection is called *regular* if $\mathrm{p}(\mathcal{P} \cap \mathcal{C}) = \mathrm{p}(\mathcal{P}) + \mathrm{p}(\mathcal{C})$, see [13]. Let $\mathcal{K} \in \mathcal{L}^q$ be a given behavior, to be interpreted as a 'desired' behavior. If $\mathcal{K}$ can be achieved as controlled behavior, i.e. if there exists $\mathcal{C} \in \mathcal{L}^q$ such that $\mathcal{K} = \mathcal{P} \cap \mathcal{C}$, then we call $\mathcal{K}$ *implementable by full interconnection* (with respect to $\mathcal{P}$). If $\mathcal{K}$ can be achieved by regular interconnection, i.e., if there exists $\mathcal{C}$ such that $\mathcal{K} = \mathcal{P} \cap \mathcal{C}$ and $\mathrm{p}(\mathcal{P} \cap \mathcal{C}) = \mathrm{p}(\mathcal{P}) + \mathrm{p}(\mathcal{C})$, then we call $\mathcal{K}$ *regularly implementable by full interconnection*. The following result from [7] gives conditions for regular implementability by full interconnection (see also [1], [4], [8]).

Theorem 3.1: Let $\mathcal{P}, \mathcal{K} \in \mathcal{L}^q$. Let $R(\mathrm{d}/\mathrm{d}t)w = 0$ and $K(\mathrm{d}/\mathrm{d}t)w = 0$ be minimal kernel representations of $\mathcal{P}$ and $\mathcal{K}$, respectively. Then the following statements are equivalent:

1) $\mathcal{K}$ is regularly implementable w.r.t. $\mathcal{P}$ by full interconnection;
2) there exists a polynomial matrix F with $F(\lambda)$ full-row rank for all $\lambda \in \mathbb{C}$, such that $R = FK$;
3) $\mathcal{K} + \mathcal{P}_{\mathrm{cont}} = \mathcal{P}$.

Here, $\mathcal{P}_{\mathrm{cont}}$ denotes the controllable part of $\mathcal{P}$.

We propose now an algorithmic implementation of this result to check regular implementability of a given behavior, and to compute a controller that regularly implements it. The algorithm is based on the equivalence of steps 1. and 2. Note that if K has full-row rank and the polynomial equation $R = FK$ has a solution F, then it is unique.

Algorithm 3.2

Input: full-row rank polynomial matrices R and K.

1. Solve the equation $R = FK$. If no solution exists, $\mathcal{K}$ is not regularly implementable. If a solution F exists go to step 2.
2. (Column compression.) Compute a unimodular U such that $FU = [F_1\ 0]$, with F_1 full column rank.
3. Check if F_1 is square and unimodular. If not, then $\mathcal{K}$ is not regularly implementable, otherwise it is. In that case go to step 4 to compute a controller.
4. Compute W such that $(F; W)$ is unimodular.
5. Put $C = WK$. Then the controller $\ker(C)$ regularly implements $\mathcal{K}$.

An implementation of this algorithm uses the command `axb` for step 1, and `colred` for step 2. The command `colred` computes also the inverse of U, which can be used in step 4 to compute an embedding matrix W as in step 3 of algorithm 1: define $W = [0\ I]U^{-1}$, where I denotes the identity matrix of appropriate size.

In the above we have restricted ourselves to *full* interconnection. We will now deal with the general case of *partial* interconnection. Again, let a plant behavior be given. If we are only allowed to interconnect the plant to a controller through a specific subset of the system variables, we use the term *partial interconnection*. In that case, our plant has two kinds of variables, w and c. The variables w are interpreted as the variables to be controlled, the variables c are those through which we can interconnect the plant to a controller (the *control variables*). This setup has been discussed before in [1], [4], [5], [10], [14]. More specific, assume we have a linear differential plant behavior $\mathcal{P}_{\rm full} \in \mathcal{L}^{q+k}$, with system variable (w, c), were w takes its values in $\mathbb{R}^q$ and c in $\mathbb{R}^k$. Let $\mathcal{C} \in \mathcal{L}^k$ be a controller behavior, with variable c. The interconnection of $\mathcal{P}_{\rm full}$ and $\mathcal{C}$ through c is defined as the system behavior $\mathcal{K}_{\rm full}(\mathcal{C}) \in \mathcal{L}^{q+k}$, defined as

$$\mathcal{K}_{\rm full}(\mathcal{C}) = \{(w,c) | (w,c) \in \mathcal{P}_{\rm full} \text{ and } c \in \mathcal{C}\},$$

which is called *the full controlled behavior*. We define the *manifest controlled behavior* as the behavior in $\mathcal{L}^q$ obtained by eliminating the control variable c from $\mathcal{K}_{\rm full}(\mathcal{C})$. This behavior is given by

$$(\mathcal{K}_{\rm full}(\mathcal{C}))_w = \{w | \exists\ c \in \mathcal{C} \text{ such that } (w,c) \in \mathcal{P}_{\rm full}\}^{\rm closure},$$

where we take the closure in the topology of $\mathcal{L}_1^{\rm loc}$. Let $\mathcal{K} \in \mathcal{L}^q$ be a given behavior, to be interpreted as a "desired" behavior. If there exists a controller $\mathcal{C} \in \mathcal{L}^q$ such that $\mathcal{K} = (\mathcal{K}_{\rm full}(\mathcal{C}))_w$ we call $\mathcal{K}$ *implementable by partial interconnection (through c, with respect to $\mathcal{P}_{\rm full}$)*. Necessary and sufficient conditions for a given $\mathcal{K} \in \mathcal{L}^q$ to be implementable by partial interconnection have been obtained in [14]. We review these conditions here. They are given in terms of the *manifest plant behavior* and *hidden behavior* associated with the full plant behavior $\mathcal{P}_{\rm full}$, which are defined as follows. The manifest plant behavior is the behavior $(\mathcal{P}_{\rm full})_w \in \mathcal{L}^q$ obtained from $\mathcal{P}_{\rm full}$ by eliminating c : $(\mathcal{P}_{\rm full})_w = \{w | \text{there exists } c \text{ such that } (w,c) \in \mathcal{P}_{\rm full}\}^{\rm closure}$. The hidden behavior consists of those w trajectories that appear in $\mathcal{P}_{\rm full}$ with c equal to zero: $\mathcal{N} = \{w | (w,0) \in \mathcal{P}_{\rm full}\}$. According to [14], a given $\mathcal{K} \in \mathcal{L}^q$ is implementable by partial interconnection through c if and only if $\mathcal{N} \subseteq \mathcal{K} \subseteq (\mathcal{P}_{\rm full})_w$. In [9] these conditions were generalized to more general classes of systems.

If $\mathcal{K}$ can be achieved by *regular* partial interconnection, i.e. if there exists $\mathcal{C}$ such that $\mathcal{K} = (\mathcal{K}_{\rm full}(\mathcal{C}))_w$ and $\mathrm{p}(\mathcal{K}_{\rm full}(\mathcal{C})) = \mathrm{p}(\mathcal{P}_{\rm full}) + \mathrm{p}(\mathcal{C})$, then we call $\mathcal{K}$ *regularly implementable by partial interconnection*. Conditions for regular implementability were given in [4] (see also [1], [12]):

Proposition 3.3: Let $\mathcal{P}_{\rm full} \in \mathcal{L}^{q+k}$. $\mathcal{K} \in \mathcal{L}^q$ is regularly implementable by partial interconnection through c if and only if $\mathcal{N} \subseteq \mathcal{K} \subseteq (\mathcal{P}_{\rm full})_w$ and $\mathcal{K}$ is regularly implementable w.r.t. $(\mathcal{P}_{\rm full})_w$ by full interconnection.

We now develop an algorithm to check regular implementability by partial interconnection, and to compute a required controller. Suppose that $\mathcal{P}_{\rm full} = \ker[R_1\ R_2]$ is a minimal kernel representation. Clearly, the hidden behavior $\mathcal{N}$ is equal to $\ker(R_1)$. Let U be a unimodular matrix such that $UR_2 = (R_{12}; 0)$ and such that R_{12} has full-row rank. Let R_{11} and R_{21} be obtained by partitioning in the same way $UR_1 = (R_{11}; R_{21})$. It is well known (see, e.g., [6, Ch. 6]), that $(\mathcal{P}_{\rm full})_w = \ker(R_{21})$ is a minimal kernel representation of the manifest plant behavior. This leads to the following algorithm to check regular implementability of $\mathcal{K}$ with respect to $\mathcal{P}_{\rm full}$.

Algorithm 3.4

Input: $[R_1\ R_2]$ and K of full-row rank.

1. Compute a solution L of $K = LR_1$. If no solution exists, $\mathcal{K}$ is not regularly implementable. Otherwise go to step 2.
2. (Row compression.) Compute a unimodular matrix U and a full-row rank polynomial matrix R_{12} such that $UR_2 = (R_{12}; 0)$.
3. Partition $UR_1 = (R_{11}; R_{21})$ compatible with step 2.
4. Solve the equation $R_{21} = FK$. If no solution F exists, $\mathcal{K}$ is not regularly implementable. If a solution F exists, go to step 5.
5. (Column compression.) Compute a unimodular U such that $FU = [F_1\ 0]$, with F_1 full column rank.
6. Check if F_1 is square and unimodular. If not, then $\mathcal{K}$ is not regularly implementable, otherwise it is. In that case go to step 7 to compute a controller.
7. Compute W such that $(F; W)$ is unimodular.
8. Compute $C = WLR_2$. Then $\ker(C)$ regularly implements $\mathcal{K}$.

An implementation of the algorithm again only uses the standard commands from the Polynomial Toolbox that we already mentioned.

IV. Algorithms for Stabilization and Pole Placement

This section deals with the synthesis problems of stabilization and pole placement by regular interconnection. We will give algorithms to compute, for a given plant behavior, controllers that achieve pole placement and stabilization. These algorithms require the computation of unimodular and stable embeddings.

For completeness, we first discuss the problems of pole placement and stabilization by regular *full* interconnection. Before reviewing these problems, we first recall some facts on autonomous systems. If a behavior $\mathfrak{B} \in \mathcal{L}^{\rm q}$ has the property that $\mathrm{p}(\mathfrak{B}) = q$ (so all variables are output), then we call $\mathfrak{B}$ *autonomous*. An autonomous system is called *stable* if $\lim_{t\to\infty} w(t) = 0$ for all $w \in \mathfrak{B}$. For autonomous behaviors we have the notion of characteristic polynomial. If $\mathfrak{B}$ is autonomous then there exists a $q \times q$ polynomial matrix R with $\det(R) \neq 0$ such that $\mathfrak{B} = \ker(R)$. We can choose R such that $\det(R)$ has leading coefficient equal to 1. This monic polynomial is denoted by $\chi_{\mathfrak{B}}$ and is called *the characteristic polynomial of* $\mathfrak{B}$. Of course $\mathfrak{B}$ is stable if and only if $\chi_{\mathfrak{B}}$ is Hurwitz.

Let $\mathcal{P} \in \mathcal{L}^q$ be a given plant behavior. The problem of pole placement is defined as follows. Let r be a monic real polynomial. Find a controller behavior $\mathcal{C}$ such that the controlled behavior $\mathcal{K} = \mathcal{P} \cap \mathcal{C}$ is autonomous, has characteristic polynomial $\chi_{\mathcal{K}} = r$, and the interconnection is regular. It was proven in [13] that for every monic real polynomial r there exists such controller $\mathcal{C}$ if and only if the plant behavior $\mathcal{P}$ is controllable and $\mathrm{p}(\mathcal{P}) < q$. The stabilization problem is the following: find a controller behavior $\mathcal{C}$ such that the controlled behavior $\mathcal{K} = \mathcal{P} \cap \mathcal{C}$ is autonomous, its characteristic polynomial $\chi_{\mathcal{K}}$ is Hurwitz, and the interconnection is regular. It was proven in [13] that there exists such controller $\mathcal{C}$ if and only if the plant behavior $\mathcal{P}$ is stabilizable.

Assume now that $\mathcal{P} = \ker(R)$ is a minimal representation. Then $\mathcal{P}$ is controllable if and only if $R(\lambda)$ has full-row rank for all $\lambda \in \mathbb{C}$, see [6]. Obviously, $\mathrm{p}(\mathcal{P}) < q$ if and only if the number of rows of R is less

than q. Assume this to be the case. In the following we first describe an algorithm that checks controllability.

Algorithm 4.1

Input: a full-row rank polynomial matrix R.

1. (Column compression.) Find a unimodular polynomial matrix U such that $RU = [R_1\ 0]$ with R_1 full column rank.
2. Check if R_1 is square and unimodular. If it is, then $\mathcal{P}$ is controllable, otherwise it is not.

If $\mathcal{P}$ is controllable, the following simple algorithm computes a controller that assigns the characteristic polynomial r.

Algorithm 4.2

Input: a full-row rank polynomial matrix R and a real monic polynomial r.

1. Compute a real polynomial matrix C_1 such that $(R; C_1)$ is unimodular.
2. Take C to be any polynomial matrix obtained by multiplying one of the rows of C_1 by r.

Define the controller $\mathcal{C} = \ker(C)$. Then the controlled system behavior $\mathcal{K} = \mathcal{P} \cap \mathcal{C}$ is represented by

$$\begin{bmatrix} R\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \\ C\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \end{bmatrix} w = 0$$

Clearly, $\det(R; C) = r$, so the controlled system $\mathcal{K}$ is autonomous and $\chi_{\mathcal{K}} = r$. As before, in the above, the column compression in algorithm 4.1, step 1 can be implemented using `colred`. This command also computes U^{-1}, which can be used to compute an embedding $C_1 = [0\ I]U^{-1}$ in algorithm 4.2.

Next, we consider the stabilization problem. Again, assume that $\mathcal{P}$ is represented by the minimal kernel representation $R(\mathrm{d}/\mathrm{d}t)w = 0$, with R a real polynomial matrix. $\mathcal{P}$ is stabilizable if and only if $R(\lambda)$ has full-row rank for all $\lambda \in \mathbb{C}^+$. An algorithm to check stabilizability and to compute a stabilizing controller has similar steps as algorithms 4.1 and 4.2: in algorithm 4.1, step 2 is replaced by checking whether R_1 is square and Hurwitz, while algorithm 4.2 is replaced by the computation of a real polynomial matrix C such that $(R; C)$ is Hurwitz. The controller behavior $\mathcal{C} := \ker(C)$ then stabilizes the system: $\det(R; C)$ is a Hurwitz polynomial so the controlled system $\mathcal{K}$ is autonomous and stable.

We now consider the partial interconnection case. The problem of pole placement by regular *partial* interconnection through c is formulated as follows: given a real monic polynomial r, find a controller $\mathcal{C} \in \mathcal{L}^k$ such that the manifest controlled behavior $(\mathcal{K}_{\mathrm{full}}(\mathcal{C}))_w$ is autonomous, has characteristic polynomial r, and the interconnection is regular. It was shown in [4] that for every real monic polynomial r there exists a required controller $\mathcal{C}$ if and only if $(\mathcal{P}_{\mathrm{full}})_w$ is controllable, $\mathrm{p}((\mathcal{P}_{\mathrm{full}})_w) < q$ and in $\mathcal{P}_{\mathrm{full}}$, w is observable from c. This observability condition is equivalent with $\mathcal{N} = 0$.

Suppose now that $\mathcal{P}_{\mathrm{full}} = \ker[R_1\ R_2]$ is a minimal kernel representation. We will now first establish an algorithm to check whether pole placement is possible, i.e., whether the controllablity and observability conditions are satisfied, and $\mathrm{p}((\mathcal{P}_{\mathrm{full}})_w) < q$. Clearly, $\mathcal{N} = \ker(R_1)$. Let U be a unimodular matrix such that $UR_2 = (R_{12}; 0)$ and such that R_{12} has full-row rank. Let R_{11} and R_{21} be obtained by partitioning in the same way $UR_1 = (R_{11}; R_{21})$. Then $\mathcal{P}_{\mathrm{full}}$ has a minimal kernel representation

$$\begin{bmatrix} R_{11}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) & R_{12}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \\ R_{21}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) & 0 \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0.$$

From this representation it is clear that $(\mathcal{P}_{\mathrm{full}})_w = \ker(R_{21})$ is a minimal kernel representation of the manifest plant behavior. Furthermore, the hidden behavior $\mathcal{N}$ is also represented by

$$\begin{bmatrix} R_{11}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \\ R_{21}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \end{bmatrix} w = 0. \tag{1}$$

This yields the following algorithm.

Algorithm 4.3

Input: $[R_1\ R_2]$ of full-row rank.

1. (Row compression of R_1.) Compute a unimodular V such that $VR_1 = (R_{11}; 0)$ with R_{11} full-row rank.
2. Check whether R_{11} is square and unimodular. If it is not, then we do not have observability, otherwise go to step 3.
3. (Row compression of R_2.) Compute a unimodular matrix U such that $UR_2 = (R_{12}; 0)$ with R_{12} full-row rank.
4. Partition $UR_1 = (R_{11}; R_{21})$ compatible with step 3.
5. Check whether the number of rows of R_{21} is less than its number of columns. If it is not, then pole placement is not possible. If it is, then go to step 6.
6. Apply algorithm 4.1 to R_{21} (to check controllability of $\ker(R_{21})$).

After checking the conditions, the following algorithm computes, for a given r, a required polynomial matrix C (representing a controller $\mathcal{C}$) that assigns the characteristic polynomial r. In the algorithm we need to solve a unimodular embedding problem twice. We assume that $(\mathcal{P}_{\mathrm{full}})_w$ is controllable, equivalently $R_{21}(\lambda)$ has full-row rank for all λ, and that $\mathrm{p}((\mathcal{P}_{\mathrm{full}})_w) < q$, equivalently the number of rows of R_{21} is less than its number of columns. Then R_{21} can be embedded into a unimodular polynomial matrix. We also assume that $\mathcal{N} = 0$, equivalently $(R_{11}(\lambda); R_{21}(\lambda))$ has full column rank for all λ. Thus $(R_{11}; R_{21})$ can be embedded into a unimodular matrix as well. Choose polynomial matrices U_{12} and U_{22} such that

$$\begin{bmatrix} R_{11} & U_{12} \\ R_{21} & U_{22} \end{bmatrix} \tag{2}$$

is unimodular. Next, solve the polynomial equation (in the unknowns X and Y)

$$\begin{bmatrix} R_{11} & U_{12} \\ R_{21} & U_{22} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} R_{12} \\ 0 \end{bmatrix}. \tag{3}$$

Then we have

$$\begin{bmatrix} R_{11} & U_{12} \\ R_{21} & U_{22} \end{bmatrix} \begin{bmatrix} I & X \\ 0 & Y \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & 0 \end{bmatrix}$$

so $\mathcal{P}_{\mathrm{full}}$ also has the minimal kernel representation

$$\begin{bmatrix} I & X\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \\ 0 & Y\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0.$$

From this, note that $(w,c) \in \mathcal{P}_{\rm full}$ implies $w = -X(\mathrm{d}/\mathrm{d}t)c$. Next, let C_0 be such that $(R_{21}; C_0)$ is unimodular, with determinant, say, $a \neq 0$, and let C_1 be any polynomial matrix obtained by multiplying one of the rows of C_0 by the desired polynomial r. Then of course $\det(R_{21}; C_1) = ar$. Define now the controller behavior $\mathcal{C} \in \mathcal{L}^k$ as the behavior represented by $C(\mathrm{d}/\mathrm{d}t)c = 0$, with C defined by $C := C_1X$. We claim that the corresponding manifest controlled behavior $(\mathcal{K}_{\rm full}(\mathcal{C}))_w$ is then represented by

$$\begin{bmatrix} R_{21}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \\ C_1\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \end{bmatrix} w = 0,$$

and that the interconnection of $\mathcal{P}_{\rm full}$ with $\mathcal{C}$ is regular.

Indeed, let $w \in (\mathcal{K}_{\rm full}(\mathcal{C}))_w$. Then there exists c such that $(w,c) \in \mathcal{P}_{\rm full}$ and $c \in \mathcal{C}$. Hence, $C(\mathrm{d}/\mathrm{d}t)c = 0$, so $C_1(\mathrm{d}/\mathrm{d}t)X(\mathrm{d}/\mathrm{d}t)c = 0$. Also, $(w,c) \in \mathcal{P}_{\rm full}$, so $w = -X(\mathrm{d}/\mathrm{d}t)c$. This yields $C_1(\mathrm{d}/\mathrm{d}t)w = 0$. Also, $R_{21}(\mathrm{d}/\mathrm{d}t)w = 0$. Conversely, assume that $R_{21}(\mathrm{d}/\mathrm{d}t)w = 0$ and $C_1(\mathrm{d}/\mathrm{d}t)w = 0$. There exists c such that $(w,c) \in \mathcal{P}_{\rm full}$. Hence $w = -X(\mathrm{d}/\mathrm{d}t)c$ so $C_1(\mathrm{d}/\mathrm{d}t)X(\mathrm{d}/\mathrm{d}t)c = 0$, equivalently, $c \in \mathcal{C}$. Thus we obtain $(w,c) \in \mathcal{K}_{\rm full}$, so $w \in (\mathcal{K}_{\rm full})_w$. We conclude that $(\mathcal{K}_{\rm full}(\mathcal{C}))_w$ has characteristic polynomial r. It can also be shown that the interconnection is regular. This leads to the following algorithm that takes the polynomial matrices representing the full plant behavior and the desired polynomial as input, and produces as output a polynomial matrix representing a desired controller.

Algorithm 4.4

Input: a polynomial matrix $[R_1\ R_2]$ of full-row rank and a monic real polynomial r.

1. (Row compression.) Compute a unimodular matrix U and a full-row rank polynomial matrix R_{12} such that $UR_2 = (R_{12}; 0)$
2. Partition $UR_1 = (R_{11}; R_{21})$ compatible with step 1.
3. Compute polynomial matrices U_{12} and U_{22} such that (2) is unimodular
4. Compute polynomial matrices X and Y that solve the polynomial equation (3).
5. Compute a polynomial matrix C_0 such that $(R_{21}; C_0)$ is unimodular. Compute C_1 as any polynomial matrix obtained by multiplying one of the rows of C_0 by r.
6. Compute $C = C_1X$.

A required controller is then given by $\ker(C)$.

Next we consider the problem of stabilization by regular partial interconnection. This problem is formulated as follows. Given $\mathcal{P}_{\rm full} \in \mathcal{L}^{q+k}$, find a controller $\mathcal{C} \in \mathcal{L}^k$ such that the corresponding manifest controlled behavior $(\mathcal{K}_{\rm full}(\mathcal{C}))_w$ is stable, and the interconnection is regular. This problem was studied extensively in [4], and it was shown that for the full plant behavior $\mathcal{P}_{\rm full}$ such controller exists if and only if the manifest plant behavior $(\mathcal{P}_{\rm full})_w$ is stabilizable and in $\mathcal{P}_{\rm full}w$ is detectable from c. This detectability condition is equivalent with the condition that the hidden behavior $\mathcal{N}$ is stable, see [4]. Again, let $\mathcal{P}_{\rm full} = \ker[R_1\ R_2]$ be a minimal kernel representation. An algorithm to check stabilizability and detectability is analogous to algorithm 4.3: step 2 is replaced by checking whether R_{11} is square and Hurwitz, step 5 is omitted, and in step 6 one should check whether $\ker(R_{21})$ is stabilizable.

We establish now an algorithm to compute a stabilizing controller $\mathcal{C} = \ker(C)$. Let U be a unimodular matrix that leads to the polynomial matrices R_{11}, R_{12} and R_{21} as in the above. Again, $(\mathcal{P}_{\rm full})_w$ is represented by $R_{21}(\mathrm{d}/\mathrm{d}t)w = 0$ and $\mathcal{N}$ by (1). We assume that $(\mathcal{P}_{\rm full})_w$ is stabilizable and $\mathcal{N}$ is stable. Then $R_{21}(\lambda)$ has full-row rank for all $\lambda \in \mathbb{C}^+$ and $(R_{11}(\lambda); R_{21}(\lambda))$ has full column rank for all $\lambda \in \mathbb{C}^+$. Hence there exists a Hurwitz polynomial matrix G and polynomial matrices R'_{11} and R'_{21} such that

$$\begin{bmatrix} R_{11} \\ R_{21} \end{bmatrix} = \begin{bmatrix} R'_{11} \\ R'_{21} \end{bmatrix} G \tag{4}$$

and such that $(R'_{11}(\lambda); R'_{21}(\lambda))$ has full column rank for all $\lambda \in \mathbb{C}$. Choose polynomial matrices U_{12} and U_{22} such that

$$\begin{bmatrix} R'_{11} & U_{12} \\ R'_{21} & U_{22} \end{bmatrix} \tag{5}$$

is unimodular. Next, solve the polynomial equation (in the unknowns X and Y)

$$\begin{bmatrix} R'_{11} & U_{12} \\ R'_{21} & U_{22} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} R_{12} \\ 0 \end{bmatrix}. \tag{6}$$

Then we have

$$\begin{bmatrix} R'_{11} & U_{12} \\ R'_{21} & U_{22} \end{bmatrix} \begin{bmatrix} G & X \\ 0 & Y \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & 0 \end{bmatrix}$$

so $\mathcal{P}_{\rm full}$ also has a minimal kernel representation

$$\begin{bmatrix} G & X\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \\ 0 & Y\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0.$$

From this, note that $(w,c) \in \mathcal{P}_{\rm full}$ implies $G(\mathrm{d}/\mathrm{d}t)w = -X(\mathrm{d}/\mathrm{d}t)c$. Next, let C_0 be such that $(R'_{21}; C_0)$ is Hurwitz (such C_0 exists since $R'_{21}(\lambda)$ has full-row rank for all $\lambda \in \mathbb{C}^+$. Define now the controller behavior $\mathcal{C} \in \mathcal{L}^k$ as the behavior represented by $C(\mathrm{d}/\mathrm{d}t)c = 0$, with C defined by $C := C_0X$. Similar as in the above, it can be shown that the corresponding manifest controlled behavior $(\mathcal{K}_{\rm full}(\mathcal{C}))_w$ is then represented by

$$\begin{bmatrix} R_{21}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \\ (C_0G)\left(\frac{\mathrm{d}}{\mathrm{d}t}\right) \end{bmatrix} w = 0,$$

$$R = \begin{bmatrix} -0.9 + 2s + s^2 + 0.1s^3 + 2s^4 & -1.5 + 13s + 2.5s^2 + s^3 + 4s^4 + s^5 & 11 - 0.3s^2 + 2s^3 + 1.3s^4 \\ 1.6 - 8s + 1.1s^3 + 2s^4 & 3 - 9s - s^2 + 1.5s^3 + 7s^4 + s^5 & 1 + s + 0.6s^2 + 3s^3 + 2s^4 + 0.3s^5 \end{bmatrix}$$

and

$$K = \begin{bmatrix} 1 & 1.5 + s & s + 0.3s^2 \\ 0.1 + 2s & 3s + s^2 & s \\ 0 & s & 1 \end{bmatrix}.$$

$$U = \begin{bmatrix} -s & 1 & s^2 \\ 1.1s - 0.05s^2 & 0.05s & 1 + 0.05s^3 \\ 1 + 0.058s - 0.0053s^2 & 0.053 + 0.0053s & 0.11 + 0.053s^2 + 0.0053s^3 \end{bmatrix}.$$

and that the interconnection of $\mathcal{P}_{\text{full}}$ with $\mathcal{C}$ is regular. Since $\det(R_{21}; C_0 G) = \det(R'_{21}; C_0)\det(G)$, $(\mathcal{K}_{\text{full}}(\mathcal{C}))_w$ is stable. Summarizing, this leads to the following algorithm:

Algorithm 4.5

Input: a full-row rank polynomial matrix $[R_1\ R_2]$.

1. (Row compression.) Compute a unimodular matrix U and a full-row rank polynomial matrix R_{12} such that $UR_2 = (R_{12}; 0)$.
2. Partition $UR_1 = (R_{11}; R_{21})$ compatible with step 1.
3. (Row compression.) Compute a unimodular V such that $V(R_{11}; R_{21}) = (G; 0)$ and G square and nonsingular. (Note: G is Hurwitz!).
4. Compute $(R'_{11}; R'_{21})$ from the equation $V(R'_{11}; R'_{21}) = (I; 0)$. (Note: we then have (4) and $(R'_{11}(\lambda); R'_{21}(\lambda))$ has full column rank for all $\lambda \in \mathbb{C}$.)
5. Compute polynomial matrices U_{12} and U_{22} such that (5) is unimodular.
6. Compute polynomial matrices X and Y that solve the polynomial equation (6).
7. Compute a polynomial matrix C_0 such that $(R'_{21}; C_0)$ is Hurwitz.
8. Compute $C = C_0 X$.

Note that implementations of all algorithms in this section only require the Matlab commands that we have already mentioned in Section II.

V. Numerical Examples

In this section, we apply some of the algorithms developed in this paper to concrete examples.

Example 5.1: In this example, we apply Algorithm 4 to check for a given plant $\mathcal{P} = \ker(R)$ whether $\mathcal{K} = \ker(K)$ is regularly implementable, and to compute a polynomial matrix C such that the controller $\mathcal{C} = \ker(C)$ regularly implements $\mathcal{K}$. Let (see the equations shown at the bottom of the previous page).

1. Use `xab` to compute

$$F = \begin{bmatrix} -1 + s^2 & 1 + s^3 & 11 + s^3 \\ 2 + s^3 & -4 + s^3 & 1 + 3s + 3s^3 \end{bmatrix}.$$

2, 3. Use `colred` to compute $F_1 = \begin{bmatrix} -2.4 & 11 \\ -4.3 & 1 \end{bmatrix}$. F_1 is unimodular, so $\mathcal{K}$ is regularly implementable w.r.t. $\mathcal{P}$. `colred` also computes a required unimodular U and its inverse U^{-1}.
4. Compute W as the last row of U^{-1}, which is given by

$$W = [1 - 0.04s - 0.085s^2 + 0.12s^3, \\ - 0.066s + 0.15s^2 - 0.092s^3 \\ 0.16s + 0.39s^2 + 0.14s^3].$$

Then $(F; W)$ is unimodular: $\det(F; W) = 45$.
5. Compute $C = WK = [1 - 0.046s - 0.2s^2 + 0.4s^3 - 0.18s^4, 1.5 + 0.94s - 0.21s^2 + 0.85s^3 + 0.13s^4 - 0.092s^5, 1.2s + 0.59s^2 + 0.19s^3 - 0.0016s^4 + 0.035s^5]$.

Example 5.2: In this example we apply algorithms 4.3 and 4.4 to compute a controller that assigns a desired characteristic polynomial to the manifest controlled behavior. Consider the full plant behavior $\mathcal{P}_{\text{full}} = \ker(R_1\ R_2)$ with R_1 and R_2 given by

$$R_1(s) = \begin{bmatrix} -1 & -1 \\ -s^3 - 3s^2 - s & -s^3 - 6s^2 - s \\ s + 3 & s + 6 \end{bmatrix}$$

$$R_2(s) = \begin{bmatrix} -1 & -3 \\ -s^3 - 9s^2 - s + 1 & -3.1s^3 - 28s^2 - 3s + 1 \\ s + 9 & 3.1s + 28 \end{bmatrix}.$$

We first apply Algorithm 4.3:

1, 2. Use `rowred` to compute $R_{11} = \begin{bmatrix} -1 & -1 \\ 3 & 6 \end{bmatrix}$. Obviously, R_{11} is unimodular, so w is observable from c.
3. Use `rowred` to compute $R_{12} = \begin{bmatrix} 1 & 1 \\ 9 & 28 \end{bmatrix}$, obtained using the unimodular matrix as shown in the equation at the top of the page.
4. Premultiplying R_1 by U yields $R_{11} = \begin{bmatrix} 0 & 0 \\ 3 - 0.05s & 6 - 0.05s \end{bmatrix}$ and $R_{21} = [-0.68 - 0.0053s, -0.37 - 0.0053s]$.
5, 6. The number of rows of R_{21} is less than its number of columns. Finally, use `colred` to columncompress R_{21} to [−0.37,0]. This shows that $\ker(R_{21})$ is controllable.

We continue then with Algorithm 4.4.

1, 2. R_{12}, R_{11} and R_{12} have already been computed in steps 3 and 4 above.
3. Compute U_{12} and U_{22} such that (2) is unimodular. These are computed as $U_{12} = [1, 0]'$ and $U_{22} = 0$.
4. Use `axb` to solve the (3). This yields $X = \begin{bmatrix} -1.1 - 0.016s & -3.4 - 0.049s \\ 2.1 + 0.016s & 6.4 + 0.049s \end{bmatrix}$
5, 6. With $C_0 := [1, 1]$, $(R_{21}; C_0)$ is unimodular. Choose as desired characteristic polynomial $r(s) = (s + 2)^2$. Define $C_1 = r(s)C_0$, and $C = C_1 X = [4 + 4s + s^2, 12 + 12s + 3s^2]$.

The controller $C(d/dt)c = 0$ yields the characteristic polynomial $r(s)$. Indeed, applying this controller, and eliminating c we obtain that the manifest controlled behavior is given by $\ker(R)$, with

$$R = \begin{bmatrix} -0.68 - 0.0053s & -0.37 - 0.0053s \\ -1.3 - 4s - s^2 & -2.5 - 4s - s^2 \end{bmatrix}.$$

We compute $\det(R) = 1.3 + 1.3s + 0.32s^2$, the characteristic polynomial of $\ker(R)$ is the monic polynomial with the same zeros, which is indeed equal to $r(s)$.

VI. Conclusion

In this correspondence, we have introduced algorithms to compute controllers in the context of control by interconnection. These algorithms act on the polynomial matrices that define kernel representations of the systems to be controlled. We have established algorithms for the computation of controllers that regularly implement a given desired controlled behavior, both for the full interconnection and the partial interconnection case. Also, algorithms were proposed for checking the properties of controllability, observability, stabilizability and detectability in the behavioral framework. Finally, we gave algorithms to compute controllers that assign the characteristic polynomial of the manifest controlled behavior, and to compute stabilizing controllers. A central role in our algorithms is played by the problems of unimodular and stable embedding. The algorithms in this paper can all be implemented using standard commands from the Matlab Polynomial Toolbox.

References

[1] M. N. Belur, "Control in a Behavioral Context," Doctoral Dissertation, University of Groningen, Groningen, The Netherlands, 2003.

[2] T. Beelen and P. Van Dooren, "A pencil approach for embedding a polynomial matrix into a unimodular matrix," *SIAM J. Matrix Anal. Appl.*, vol. 9, no. 1, pp. 77–89, 1988.

[3] M. Bisiacco and M. Valcher, "A note on the direct sum decompositions of two-dimensional behaviors," *IEEE Trans. Circuits Syst.*, vol. 48, no. 4, pp. 490–494, 2001.

[4] M. N. Belur and H. L. Trentelman, "Stabilization, pole placement and regular implementability," *IEEE Trans. Automat. Control*, vol. 47, no. 5, pp. 735–744, 2002.

[5] J. W. Polderman and I. Mareels, "A behavioral approach to adaptive control," in *The Mathematics of Systems and Control: From Intelligent Control to Behavioral Systems*, J. W. Polderman and H. L. Trentelman, Eds. Groningen, The Netherlands: Foundation Systems and Control Groningen, 1999, pp. 119–130.

[6] J. W. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Berlin, Germany: Springer-Verlag, 1997.

[7] C. Praagman, H. L. Trentelman, and R. Z. Yoe, "On the parametrization of all regularly implementing and stabilizing controllers," *SIAM J. Contr. Optim.*, to be published.

[8] P. Rocha and J. Wood, "Trajectory control and interconnection of 1D and nD systems," *SIAM J. Contr. Optim.*, vol. 40, pp. 107–134, 2001.

[9] A. J. van der Schaft, "Achievable behavior of general systems," *Syst. Contr. Lett.*, vol. 49, pp. 141–149, 2003.

[10] H. L. Trentelman and J. C. Willems, "Synthesis of dissipative systems using quadratic differential forms: Part II," *IEEE Trans. Automat. Control*, vol. 47, pp. 70–86, 2002.

[11] P. Van Dooren, "The basics of developing numerical algorithms," *IEEE Contr. Syst. Mag.*, pp. 18–27, Feb. 2004.

[12] A. A. Julius, J. C. Willems, M. N. Belur, and H. L. Trentelman, "The canonical controller and regular interconnection," *Syst. Contr. Lett.*, vol. 54, no. 8, pp. 787–797, 2005, 141–149, 2003.

[13] J. C. Willems, "On interconnections, control and feedback," *IEEE Trans. Automat. Control*, vol. 42, pp. 326–339, 1997.

[14] J. C. Willems and H. L. Trentelman, "Synthesis of dissipative systems using quadratic differential forms—Part I," *IEEE Trans. Automat. Control*, vol. 47, no. 1, pp. 53–69, Jan. 2002.

The Controlled Center Systems

Boumediene Hamzi and Arthur J. Krener

***Abstract*—In this correspondence, we propose a methodology to stabilize systems with control bifurcations by introducing "The Controlled Center Systems." A controlled center system is a reduced-order controlled dynamics consisting of the linearly uncontrollable dynamics with the first variable of the linearly controllable dynamics as input. The controller of the full order system is then constructed. We apply this methodology to systems with a transcontrollable, a Hopf, and a double-zero, control bifurcation.**

I. Introduction

Center manifold theory plays an important role in the study of the stability of nonlinear systems when the equilibrium point is not hyperbolic. The center manifold is an invariant manifold of the differential (difference) equation which is tangent at the equilibrium point to the eigenspace of the neutrally stable eigenvalues. After determining the reduced dynamics on the center manifold, we study its stability and then conclude about the stability of the full order system [6].

This theory can be viewed as a model reduction technique for nonlinear dynamical systems with nonhyperbolic equilibrium points. Indeed, the stability properties of a dynamical system around an equilibrium where one or more eigenvalues of its linear part are on the imaginary axis are characterized by the local asymptotic stability of the dynamics on the center manifold. Thus, this leads to a reduction of the dimension of the dynamics that needs to be analyzed to determine local asymptotic stability of the equilibrium.

For a nonlinear control system around an equilibrium, the local asymptotic stability of the linearly controllable directions can be easily achieved by linear feedback. Therefore the stabilizability of the whole system should depend on a reduced order model that corresponds to the stabilizability of the linearly uncontrollable directions. The Controlled Center Dynamics introduced in [9] formalizes this intuition. By assuming that the stabilizing feedback has a certain structure and is characterized by certain parameters, the controlled center dynamics is a reduced order dynamical system characterized by the parameters of the feedback. By finding the conditions under which this dynamical system is stable, we deduce conditions on the parameters of the feedback, and, thus, deduce a stabilizing controller for the full order system.

In this correspondence, we present a slightly different approach. Instead of assuming that the feedback has a certain structure and is characterized by certain parameters, we synthesize a controller on a reduced-order control system called the Controlled Center System. This system is a controlled dynamical system consisting of the linearly uncontrollable dynamics with the first variable of the linearly controllable dynamics playing the role of the input. By constructing a stabilizing controller, that satisfies certain conditions, for this reduced order control system, we are able to deduce a stabilizing controller for the full order system.

The paper is organized as follows. In Section II, we review the controlled center dynamics approach. Then, in Section III, we introduce the quadratic controlled center systems and propose a methodology to stabilize systems with control bifurcations. We apply this approach to systems with a trancontrollable and Hopf control bifurcation. Finally, in

Manuscript received March 1, 2006; revised November 10, 2006 and February 18, 2007. Recommended by Associate Editor S. Celikovsky.

The authors are with the Department of Mathematics, University of California, Davis, CA 95616 USA (e-mail: hamzi@math.ucdavis.edu; krener@math.ucdavis.edu).

Digital Object Identifier 10.1109/TAC.2007.902743