REFERENCES

[1] H. Akcay and B. Ninness, "Orthonormal basis functions for continuous-time systems and Lp convergence," *Math. Contr., Signals Syst.*, vol. 12, no. 3, pp. 295–305, 1999.

[2] J.-M. Biannic, S. Tarbouriech, and D. Farret, "A practical approach to performance analysis of saturated systems with application to fighter aircraft flight controllers," in *5th 1FAC Symp. ROCOND*, Toulouse, France, 2006.

[3] G. Ferreres, *A Practical Approach to Robustness Analysis with Aeronautical Applications*. New York: Springer-Verlag, 1999.

[4] G. Ferreres and G. Puyou, "Flight control law design for a flexible aircraft: Limits of performance," *J. Guid., Control Dyn.*, vol. 29, no. 4, pp. 870–878, 2006.

[5] G. Ferreres and C. Roos, "Robust feedforward design in the presence of LTI/LTV uncertainties," *Int. J. Robust Nonlinear Control*, vol. 17, no. 14, pp. 1278–1293, Sep. 2007.

[6] A. Giusto and F. Paganini, "Robust synthesis of feedforward compensators," *IEEE Trans. Autom. Control*, vol. 44, no. 8, pp. 1578–1582, 1999.

[7] J. M. Gomes da Silva Jr. and S. Tarbouriecb, "Anti-windup design with guaranteed regions of stability: An LMI-based approach," *IEEE Trans. Autom. Control*, vol. 50, no. 1, pp. 106–111, Jan. 2005.

[8] G. Grimm, J. Hatfield, I. Postlethwaite, A. R. Teel, M. C. Turner, and L. Zaccarian, "Anti-windup for stable linear systems with input saturation: An LMI-based synthesis," *IEEE Trans. Autom. Control*, vol. 48, no. 9, pp. 1509–1525, Sep. 2003.

[9] T. Hu, A. R. Teel, and L. Zaccarian, "Nonlinear $\mathcal{L}_2$ gain and regional analysis for linear systems with anti-windup compensation," *Proc. ACC*, pp. 3391–3395, Jun. 2005, Portland, OR, USA.

[10] M. Kothare and M. Morari, "Multiplier theory for stability analysis of anti-windup control systems," *Automatica*, vol. 35, pp. 917–928, 1999.

[11] M. Kothare., P. Campo, M. Morari, and C. Nett, "A unified framework for the study of anti-windup designs," *Automatica*, vol. 30, no. 12, pp. 1869–1883, 1994.

[12] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Trans. Autom. Control*, vol. 42, no. 6, pp. 819–830, Jun. 1997.

[13] E. F. Mulder, M. V. Kothare, and M. Morari, "Multivariable anti-windup controller synthesis using linear matrix inequalities," *Automatica*, vol. 37, no. 9, pp. 1407–1416, 2001.

[14] R. T. Reichert, "Dynamic scheduling of modern robust control autopilot design for missiles," *IEEE Control Syst. Mag.*, vol. 12, no. 5, pp. 35–42, Oct. 1992.

[15] K. Sun and A. Packard, "Robust H_2 and H_∞ filters for uncertain LFT systems," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 715–720, 2005.

[16] F. Wu and M. Soto, "Extended anti-windup control schemes for LTI and LFT systems with actuator saturations," *Int. J. Robust Nonlin. Control*, vol. 14, pp. 1255–1281, 2004.

Incremental Value Iteration for Time-Aggregated Markov-Decision Processes

Tao Sun, Qianchuan Zhao, and Peter B. Luh, *Fellow, IEEE*

***Abstract*—A value iteration algorithm for time-aggregated Markov-decision processes (MDPs) is developed to solve problems with large state spaces. The algorithm is based on a novel approach which solves a time aggregated MDP by incrementally solving a set of standard MDPs. Therefore, the algorithm converges under the same assumption as standard value iteration. Such assumption is much weaker than that required by the existing time aggregated value iteration algorithm. The algorithms developed in this paper are also applicable to MDPs with fractional costs.**

***Index Terms*—Fractional cost, Markov-decision processes (MDPs), policy iteration, time aggregation, value iteration.**

I. INTRODUCTION

Markov-decision processes (MDPs) with the average cost criterion play important roles in the fields such as control, operations research and artificial intelligence (see, e.g., [1]–[3], and [5]). The major obstacle of applying MDPs to practical problems is the large state spaces which may cause MDPs intractable by standard approaches, i.e., policy iteration and value iteration. Nevertheless, value iteration generally solves much larger problems than policy iteration which solves linear equations at each iteration [3]. The recently developed time aggregation approach [1] results in reduced state spaces. This may substantially reduce the storage and computational requirements especially for problems with certain structures, e.g., a large number of uncontrollable states. For these large problems, a value iteration algorithm with time aggregation is powerful. However, such an algorithm has not been well developed yet.

The idea of time aggregation is to divide the original process into segments by certain states (e.g., those controllable states) to form an embedded (time aggregated) MDP. The performance function is converted accordingly and a policy iteration algorithm was presented in [1]. However, since the performance function is not explicitly known, it is generally difficult to perform value iteration. Due to the close similarities, the value iteration algorithm presented in [4] for MDPs with fractional cost is applicable to time aggregated MDPs. However, the algorithm converges under a strong assumption, i.e., there exists a state that is admissible with a positive probability from any state under any action.

This note develops a value iteration algorithm for time aggregated MDPs to solve problems with large state spaces. In Section II, time-aggregated MDPs are briefly reviewed and their optimal policies are investigated. A novel approach is developed in Section III to solve an aggregated MDP by incrementally solving a set of standard MDPs. This approach directly leads to a new policy iteration algorithm and helps to develop a value iteration algorithm which is proved to converge

Manuscript received March 9, 2007; revised June 15, 2007. Recommended by Associate Editor I. Paschalidis. This work was supported by the NSFC under Grants 60274011 and 60574067, the NCET program (No. NCET-04-0094) of China, and the National 111 International Collaboration Project.

T. Sun and Q. Zhao are with the Center for Intelligent and Networked Systems (CFINS), Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: suntao99@mails.tsinghua.edu.cn; zhaoqc@tsinghua.edu.cn).

P. B. Luh is with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-2157 USA and also with CFINS, Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: Peter.Luh@uconn.edu).

Digital Object Identifier 10.1109/TAC.2007.908359

in Section IV under ergodicity, an assumption much weaker than that required by the convergence of the existing algorithm [4]. Numerical testing is provided in Section V to illustrate the efficiency and effectiveness of our algorithm.

II. The Time Aggregation Approach for MDPs

A. Standard MDPs and Time-Aggregated MDPs

We first define a standard MDP, then briefly describe the time aggregated MDP developed by [1]. Most of the following notations are borrowed from [1]. We consider a discrete-time Markov chain, $\mathbf{X} = \{X_t, t = 0, 1, \ldots\}$, with a finite state–space $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$, where $|\cdot|$ denotes the set cardinality. Let $\mathcal{A}$ be a finite set of actions and $\mathcal{A}(i)$ stand for all feasible actions for state i. We consider the set of stationary policies denoted by $\mathcal{E}$. A policy $\mathcal{L} \in \mathcal{E}$ is a mapping $\mathcal{L} : \mathcal{S} \to \mathcal{A}$. Under policy $\mathcal{L}$, the action $\mathcal{L}(i) \in \mathcal{A}(i)$ taken for state i leads to state transition probability from i to j described by $p^{\mathcal{L}(i)}(i, j), j = 1, 2, \ldots, |\mathcal{S}|$. In addition, the Markov chain evolves following the transition probability matrix $P^{\mathcal{L}}$, with $[P^{\mathcal{L}}]_{i,j} = p^{\mathcal{L}(i)}(i, j)$.

Ergodicity Assumption: The Markov chain is ergodic under any policy in $\mathcal{E}$.

By ergodicity assumption, $\pi^{\mathcal{L}} P^{\mathcal{L}} = \pi^{\mathcal{L}}$ has a unique solution $\pi^{\mathcal{L}} = (\pi^{\mathcal{L}}(1), \ldots, \pi^{\mathcal{L}}(|\mathcal{S}|))$, with $\pi^{\mathcal{L}}(i)$ being the steady-state probability of state i under policy $\mathcal{L}$. Let $f^{\mathcal{L}} = (f^{\mathcal{L}}(1), \ldots, f^{\mathcal{L}}(|\mathcal{S}|))^T$ be a column vector of performance functions, where "T" denotes transpose. The performance of $\mathcal{L}$, represented by the average cost, is well defined and does not depend on the initial state

$$\eta^{\mathcal{L}} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} f^{\mathcal{L}}(X_t) = \pi^{\mathcal{L}} f^{\mathcal{L}}. \tag{1}$$

The problem is to obtain an optimal policy achieving the minimum average cost

$$\mathcal{L}^* = \arg\min_{\mathcal{L} \in \mathcal{E}} \{\eta^{\mathcal{L}}\}. \tag{2}$$

Let $\mathcal{S}_1$ and $\mathcal{S}_2 = \mathcal{S} - \mathcal{S}_1$ be two complementary subsets of $\mathcal{S}$. As in [1], we focus on a subset of MDPs, for which actions can only be taken for states in $\mathcal{S}_1$ and the transition probabilities and performance functions for states in $\mathcal{S}_2$ do not depend on actions. Therefore, $f^{\mathcal{L}}$ and $P^{\mathcal{L}}$ can be partitioned according to $\mathcal{S}_1$ and $\mathcal{S}_2$ as

$$P^{\mathcal{L}} = \begin{bmatrix} P^{\mathcal{L}}_{\mathcal{S}_1 \mathcal{S}_1} & P^{\mathcal{L}}_{\mathcal{S}_1 \mathcal{S}_2} \\ P_{\mathcal{S}_2 \mathcal{S}_1} & P_{\mathcal{S}_2 \mathcal{S}_2} \end{bmatrix} \quad \text{and} \quad f^{\mathcal{L}} = \begin{bmatrix} f^{\mathcal{L}}_{\mathcal{S}_1} \\ f_{\mathcal{S}_2} \end{bmatrix}. \tag{3}$$

For a sample path $\{X_0, X_1, \ldots\}$ generated under $\mathcal{L}$ with $X_0 \in \mathcal{S}_1$, let $t_0 = 0$ and $t_i = \min\{t | t > t_{i-1}, X_t \in \mathcal{S}_1\}, i = 1, 2, \ldots$. Then $\{X_{t_i}, i = 0, 1, \ldots\}$ forms an embedded Markov chain which is also ergodic. Let $\tilde{P}^{\mathcal{L}}$ and $\tilde{\pi}^{\mathcal{L}}$ be the transition matrix and the steady-state probability row vector of the embedded chain under $\mathcal{L}$. We have [1]

$$\tilde{P}^{\mathcal{L}} = P^{\mathcal{L}}_{\mathcal{S}_1} + P^{\mathcal{L}}_{\mathcal{S}_1 \mathcal{S}_2} (I - P_{\mathcal{S}_2 \mathcal{S}_2})^{-1} P_{\mathcal{S}_2 \mathcal{S}_1}. \tag{4}$$

Those states in the embedded chain divide the original Markov chain into segments. For example, a segment generated based on $X_{t_i} \in \mathcal{S}_1$ is $(X_{t_i}, X_{t_i+1}, \ldots, X_{t_{i+1}-1})$. The expected total cost of a segment starts from $i \in \mathcal{S}_1$ is defined as

$$H^{\mathcal{L}}_f(i) = E\left[\sum_{j=1}^{t_{i+1}-t_i} f^{\mathcal{L}}(X_{t_i+j-1}) | X_{t_i} = i \right]. \tag{5}$$

Let $H^{\mathcal{L}}_f = [H^{\mathcal{L}}_f(1), \ldots, H^{\mathcal{L}}_f(|\mathcal{S}_1|)]^T$, which can be computed as [1]

$$H^{\mathcal{L}}_f = f^{\mathcal{L}}_{\mathcal{S}_1} + P^{\mathcal{L}}_{\mathcal{S}_1 \mathcal{S}_2} (I - P_{\mathcal{S}_2 \mathcal{S}_2})^{-1} f_{\mathcal{S}_2}. \tag{6}$$

For well-structured problems (such as the replacement problem in Section V), $\tilde{P}^{\mathcal{L}}$ and $H^{\mathcal{L}}_f$ may be directly calculated by their definitions without performing burdensome matrix computations as (4) and (6). Use $H^{\mathcal{L}}_1$ to denote the case where $f^{\mathcal{L}}(i) \equiv 1$ for any $i \in \mathcal{S}$ and $\mathcal{L} \in \mathcal{E}$. The definition (5) shows that $H^{\mathcal{L}}_1(i)$ is the expected length of the segment starting from $i \in \mathcal{S}_1$. Let B_u and B_l be the upper and lower bounds of $H^{\mathcal{L}}_1(i)$. Then for any $\mathcal{L} \in \mathcal{E}$

$$1 \le B_l \le \bar{n}^{\mathcal{L}} \le B_u, \tag{7}$$

with

$$\bar{n}^{\mathcal{L}} \equiv \tilde{\pi}^{\mathcal{L}} H^{\mathcal{L}}_1. \tag{8}$$

The performance of the original Markov chain computed by (1) can also be obtained as

$$\eta^{\mathcal{L}} = \frac{\tilde{\pi}^{\mathcal{L}} H^{\mathcal{L}}_f}{\tilde{\pi}^{\mathcal{L}} H^{\mathcal{L}}_1}. \tag{9}$$

Therefore, the performance of the original MDP can be considered as the ratio of two average costs, i.e., $\tilde{\pi}^{\mathcal{L}} H^{\mathcal{L}}_f$ and $\tilde{\pi}^{\mathcal{L}} H^{\mathcal{L}}_1$ (MDPs with fractional costs [4]). If the performance function of the embedded chain is set as

$$\tilde{f}^{\mathcal{L}} = \frac{1}{\bar{n}^{\mathcal{L}}} H^{\mathcal{L}}_f \tag{10}$$

then the embedded chain has the same average cost as the original chain, i.e., $\tilde{\pi}^{\mathcal{L}} \tilde{f}^{\mathcal{L}} = \eta^{\mathcal{L}}$. It looks as if the entire segment is aggregated onto the embedded point $i \in \mathcal{S}_1$. Thus the embedded chain defined with performance function is called "time aggregated MDP."

The time aggregated MDP with performance function (10) cannot be directly solved. The reason is $\tilde{f}^{\mathcal{L}}(i)$ depends on actions taken for states other than i, as can be seen from $\bar{n}^{\mathcal{L}}$. To perform policy improvement over the embedded chain, a performance function is defined in [1] as:

$$r_\delta(i, a) \equiv H_f(i, a) - \delta H_1(i, a). \tag{11}$$

In the above, δ is a real parameter. When δ is set as $\eta^{\mathcal{L}}$, i.e., the average cost of the original MDP under policy $\mathcal{L}$, the potential vector $\tilde{g}^{\mathcal{L}}$ is computed through solving the Poisson equation [1]

$$(I - \tilde{P}^{\mathcal{L}} + e\tilde{\pi}^{\mathcal{L}})\tilde{g}^{\mathcal{L}} = r^{\mathcal{L}}_{\eta^{\mathcal{L}}} = H^{\mathcal{L}}_f - \eta^{\mathcal{L}} H^{\mathcal{L}}_1. \tag{12}$$

Then a new policy $\mathcal{L}'$ can be obtained through the following policy improvement process:

$$\mathcal{L}' = \arg\min_{\phi \in \mathcal{E}} \left\{ r^{\phi}_{\eta^{\mathcal{L}}} + \tilde{P}^{\phi} \tilde{g}^{\mathcal{L}} \right\}. \tag{13}$$

The policy iteration algorithm (Algorithm 1 in [1]) results in an optimal policy for the original MDP through iteratively carrying on the steps (12) and (13). During iterations, δ is updated to be the best-so-far performance of the policies obtained. Therefore, the performance function (11) is not explicitly known but changes during policy iterations. This makes it difficult to perform value iteration.

B. Properties of the Policies for the Aggregated MDPs

To distinguish from the original MDP (2), the embedded chain employs the performance function (11) with a fixed δ is called "aggregated MDP." Such an aggregated MDP is a standard MDP with state space $\mathcal{S}_1$ and an optimal policy $\tilde{\phi}^*_\delta$ which is obtained as

$$\tilde{\phi}^*_\delta = \arg\min_{\mathcal{L}\in\mathcal{E}}\{\tilde{\eta}^{\mathcal{L}}\}, \quad \text{with} \quad \tilde{\eta}^{\mathcal{L}} = \tilde{\pi}^{\mathcal{L}}\left(H_f^{\mathcal{L}} - \delta H_1^{\mathcal{L}}\right). \tag{14}$$

Moreover, by (8) and (9)

$$\tilde{\eta}^{\mathcal{L}} = (\eta^{\mathcal{L}} - \delta)\bar{n}^{\mathcal{L}}. \tag{15}$$

Notice that $\tilde{\phi}^*_\delta$ depends on the value of δ and may not be optimal for the original MDP. The properties of $\tilde{\phi}^*_\delta$ are analyzed as follows.

Theorem 1: Let $\mathcal{L}^*$ be an optimal policy of the original MDP (2) and $\tilde{\phi}^*_\delta$ be an optimal policy for the aggregated MDP (14).

a) If δ is chosen in the range $\delta \le \eta^{\mathcal{L}^*}$, then $\bar{n}^{\tilde{\phi}^*_\delta} \le \bar{n}^{\mathcal{L}^*}$, $\tilde{\eta}^{\tilde{\phi}^*_\delta} \ge 0$, and $\delta \le \eta^{\tilde{\phi}^*_\delta} \le \eta^{\mathcal{L}^*} + \tilde{\eta}^{\tilde{\phi}^*_\delta}/B_l$.

b) If δ is chosen in the range $\delta \ge \eta^{\mathcal{L}^*}$, then $\bar{n}^{\tilde{\phi}^*_\delta} \ge \bar{n}^{\mathcal{L}^*}$, $\tilde{\eta}^{\tilde{\phi}^*_\delta} \le 0$, and $\eta^{\tilde{\phi}^*_\delta} \le \delta \le \eta^{\mathcal{L}^*} - \tilde{\eta}^{\tilde{\phi}^*_\delta}/B_l$.

Proof: Since $\mathcal{L}^*$ is optimal for the original MDP, $\eta^{\mathcal{L}^*} \le \eta^{\tilde{\phi}^*_\delta}$. By (15)

$$\tilde{\eta}^{\tilde{\phi}^*_\delta} = \left(\eta^{\tilde{\phi}^*_\delta} - \delta\right)\bar{n}^{\tilde{\phi}^*_\delta} \ge \left(\eta^{\mathcal{L}^*} - \delta\right)\bar{n}^{\tilde{\phi}^*_\delta}. \tag{16}$$

Since $\tilde{\phi}^*_\delta$ is optimal for the aggregated MDP, therefore $\tilde{\eta}^{\tilde{\phi}^*_\delta} \le \tilde{\eta}^{\mathcal{L}^*}$, i.e.

$$\tilde{\eta}^{\tilde{\phi}^*_\delta} \le \left(\eta^{\mathcal{L}^*} - \delta\right)\bar{n}^{\mathcal{L}^*}. \tag{17}$$

Based on (16) and (17)

$$\frac{\tilde{\eta}^{\tilde{\phi}^*_\delta}}{\bar{n}^{\mathcal{L}^*}} \le \eta^{\mathcal{L}^*} - \delta \le \frac{\tilde{\eta}^{\tilde{\phi}^*_\delta}}{\bar{n}^{\tilde{\phi}^*_\delta}}. \tag{18}$$

If $\delta \le \eta^{\mathcal{L}^*}$, then $\eta^{\tilde{\phi}^*_\delta} \ge \delta$, and $\tilde{\eta}^{\tilde{\phi}^*_\delta} \ge 0$ by (16). Thus from (18), $\bar{n}^{\tilde{\phi}^*_\delta} \le \bar{n}^{\mathcal{L}^*}$. Furthermore, by (15)

$$\eta^{\tilde{\phi}^*_\delta} = \delta + \frac{\tilde{\eta}^{\tilde{\phi}^*_\delta}}{\bar{n}^{\tilde{\phi}^*_\delta}} \le \eta^{\mathcal{L}^*} + \frac{\tilde{\eta}^{\tilde{\phi}^*_\delta}}{\bar{n}^{\tilde{\phi}^*_\delta}} \le \eta^{\mathcal{L}^*} + \frac{\tilde{\eta}^{\tilde{\phi}^*_\delta}}{B_l}. \tag{19}$$

If $\delta \ge \eta^{\mathcal{L}^*}$, then by (15) and (17), $\tilde{\eta}^{\tilde{\phi}^*_\delta} \le \tilde{\eta}^{\mathcal{L}^*} \le 0$ and $\eta^{\mathcal{L}^*} \le \eta^{\tilde{\phi}^*_\delta} \le \delta$. Thus from (18), $\bar{n}^{\tilde{\phi}^*_\delta} \ge \bar{n}^{\mathcal{L}^*}$, and

$$\delta = \eta^{\mathcal{L}^*} - \frac{\tilde{\eta}^{\mathcal{L}^*}}{\bar{n}^{\mathcal{L}^*}} \le \eta^{\mathcal{L}^*} - \frac{\tilde{\eta}^{\tilde{\phi}^*_\delta}}{\bar{n}^{\mathcal{L}^*}} \le \eta^{\mathcal{L}^*} - \frac{\tilde{\eta}^{\tilde{\phi}^*_\delta}}{B_l}.$$

□

From Theorem 1, the following corollary can be obtained.

Corollary 1: An optimal policy $\tilde{\phi}^*_\delta$ for the aggregated MDP (14) is also optimal for the original MDP (2) if either of the following conditions holds.

a) $\delta = \eta^{\mathcal{L}^*}$.

b) $\bar{n}^{\tilde{\phi}^*_\delta} = \bar{n}^{\mathcal{L}^*}$.

Proof:

a) By b) of Theorem 1, $\eta^{\tilde{\phi}^*_\delta} \le \delta = \eta^{\mathcal{L}^*}$. Thus $\tilde{\phi}^*_\delta$ is optimal for the original MDP.

b) When $\bar{n}^{\tilde{\phi}^*_\delta} = \bar{n}^{\mathcal{L}^*}$, by (17)$\eta^{\tilde{\phi}^*_\delta} - \delta \le (\eta^{\mathcal{L}^*} - \delta)\bar{n}^{\mathcal{L}^*}/\bar{n}^{\tilde{\phi}^*_\delta} = \eta^{\mathcal{L}^*} - \delta$. Therefore, $\eta^{\tilde{\phi}^*_\delta} = \eta^{\mathcal{L}^*}$. □

Corollary 1 reveals that the information on the optimal policies of the original MDP is valuable for the time aggregation approach. In particular, the aggregated MDP can be directly solved as a standard MDP, which results in an optimal policy for the original MDP if we have knowledge either as a) or b).

III. An Incremental Optimization Approach

The information required for Corollary 1 is generally not readily available. This section develops an incremental optimization approach to gradually reach the optimal performance and policy for the original MDP through solving a series of standard MDPs. This approach is flexible for developing new algorithms for time aggregated MDPs. In particular, a new policy iteration algorithm can be directly obtained by using standard policy iteration to solve those standard MDPs. More importantly, the approach motivates a new value iteration algorithm in Section IV through incorporating standard value iteration. The detailed steps of the approach are presented as follows.

Algorithm 1 (An Incremental Optimization Approach)

1) Initialize $\delta_0 \in R^1$. Set iteration index $n = 0$, and specify $\epsilon > 0$.
2) Construct an MDP (14) based on the embedded chain and the performance function (11) defined by δ_n. Obtain an optimal policy $\tilde{\phi}^*_{\delta_n}$ for the MDP.
3) Compute $\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}$ by (14). If $|\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}| \le \epsilon$, let $\phi^* = \tilde{\phi}^*_{\delta_n}$, and stop. Otherwise, let

$$\delta_{n+1} = \delta_n + \alpha\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}, \quad \text{with} \quad \alpha = 1/B_u.$$

Set $n = n + 1$ and return to step 2).

The following theorem shows that Algorithm 1 converges to a policy whose performance can be quantified.

Theorem 2: Suppose $\mathcal{L}^*$ is an optimal policy of the original MDP (2). For the incremental optimization approach, we have the following.

a) $|\delta_{n+1} - \eta^{\mathcal{L}^*}| \le \beta|\delta_n - \eta^{\mathcal{L}^*}|$, where $\beta = 1 - B_l/B_u$.

b) The algorithm terminates in a finite number of iterations.

c) The policy ϕ^* satisfies $\eta^{\phi^*} \le \eta^{\mathcal{L}^*} + \epsilon/B_l$.

Proof: If $\delta_0 = \eta^{\mathcal{L}^*}$, the algorithm terminates after one iteration with $n = 0$. By Corollary 1, ϕ^* is optimal for the original chain, i.e., $\eta^{\phi^*} = \eta^{\mathcal{L}^*}$. When $\delta_0 \ne \eta^{\mathcal{L}^*}$, the proofs are provided next.

a) By (16) and (17)

$$B_l \le \bar{n}^{\mathcal{L}^*} \le \frac{\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}}{\eta^{\mathcal{L}^*} - \delta_n} \le \bar{n}^{\tilde{\phi}^*_{\delta_n}} \le B_u, \quad \text{for} \quad \delta_n > \eta^{\mathcal{L}^*} \tag{20}$$

and

$$B_l \le \bar{n}^{\tilde{\phi}^*_{\delta_n}} \le \frac{\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}}{\eta^{\mathcal{L}^*} - \delta_n} \le \bar{n}^{\mathcal{L}^*} \le B_u, \quad \text{for} \quad \delta_n < \eta^{\mathcal{L}^*}. \tag{21}$$

Therefore, $|\delta_n - \eta^{\mathcal{L}^*}| \ge |\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}/B_u|$, which implies that $\delta_{n+1} - \eta^{\mathcal{L}^*}$ has the same sign as $\delta_n - \eta^{\mathcal{L}^*}$. Then

$$\left|\frac{\delta_{n+1} - \eta^{\mathcal{L}^*}}{\delta_n - \eta^{\mathcal{L}^*}}\right| = \frac{\delta_n - \eta^{\mathcal{L}^*} + \tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}/B_u}{\delta_n - \eta^{\mathcal{L}^*}} = 1 - \frac{\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}}{\eta^{\mathcal{L}^*} - \delta_n}\frac{1}{B_u}.$$

This, together with (20) and (21), leads to

$$\frac{B_l}{B_u} \le \frac{\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}}{\eta^{\mathcal{L}^*} - \delta_n}\frac{1}{B_u} \le 1$$

$$0 \le \left|\frac{\delta_{n+1} - \eta^{\mathcal{L}^*}}{\delta_n - \eta^{\mathcal{L}^*}}\right| \le 1 - \frac{B_l}{B_u}. \tag{22}$$

b) In step 2) of Algorithm 1, the MDP, as pointed out for (14), is a standard MDP that can be efficiently solved by using existing algorithms, e.g., policy iteration. When $\delta_n < \eta^{\mathcal{L}^*}$, by Theorem 1a) and (17)

$$0 \le \tilde{\eta}^{\tilde{\phi}^*_{\delta_n}} \le \left(\eta^{\mathcal{L}^*} - \delta_n\right) \bar{n}^{\mathcal{L}^*}.$$

When $\delta_n > \eta^{\mathcal{L}^*}$, by Theorem 1b) and (16)

$$\left|\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}\right| \le \left|\left(\eta^{\mathcal{L}^*} - \delta_n\right)\right| \bar{n}^{\tilde{\phi}^*_{\delta_n}}.$$

Thus,

$$\left|\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}\right| \le \left|\eta^{\mathcal{L}^*} - \delta_n\right| B_u. \tag{23}$$

By a), $|\delta_n - \eta^{\mathcal{L}^*}|$ decreases geometrically. Thus the stopping criterion $|\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}| \le \epsilon$ will be met after a finite number of iterations for any positive number ϵ.

c) When the algorithm terminates, $|\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}| \le \epsilon$. By Theorem 1, $\eta^{\phi^*} \le \eta^{\mathcal{L}^*} + |\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}|/B_l$. Thus, $\eta^{\phi^*} \le \eta^{\mathcal{L}^*} + \epsilon/B_l$. □

Algorithm 1 is in essence a two-level optimization approach. The low level is to obtain an optimal policy $\tilde{\phi}^*_{\delta_n}$ in step 2) for a standard MDP. At the high level, δ_n incrementally approaches the optimal performance of the original MDP. The geometric convergence of the algorithm provides an estimation on the number of iterations required for algorithm termination.

IV. Incremental Value Iteration

For time aggregated MDPs, policy iteration must store matrices with dimensions $|\mathcal{S}_1| \times |\mathcal{S}_1|$ to compute steady-state probabilities and solve Poisson equation at each iteration. In contrast, value iteration does not perform such computations and may only need to store a $|\mathcal{S}_1|$-dimensional vector $\hat{g}$. Therefore, value iteration is effective for large problems with structural features suitable for the time aggregation approach. Incorporating standard value iteration in step 2) of Algorithm 1, this section develops a new value iteration algorithm which converges under much weaker assumptions than that required by the existing algorithm in [4]. The detailed steps are presented below.

Algorithm 2 (An Incremental Value Iteration Algorithm)

1) Choose a $|\mathcal{S}_1|$-dimensional vector $\hat{g}_0$ and initialize $\delta_0 \in R^1$. Set $m = n = 0$ and specify $\epsilon > \sigma > 0$.
2) For each $i \in \mathcal{S}_1$, compute $\hat{g}_{m+1}(i)$ by

$$\hat{g}_{m+1}(i) = \min_{a \in \mathcal{A}(i)} \left\{ H_f(i,a) - \delta_n H_1(i,a) + \sum_{j \in \mathcal{S}_1} \tilde{p}^a(i,j) \hat{g}_m(j) \right\}. \tag{24}$$

3) If $sp(\hat{g}_{m+1} - \hat{g}_m) \le \sigma$, where $sp(\hat{g})$ is the span of $\hat{g}$ ([3]):

$$sp(\hat{g}) \equiv \max_{i \in \mathcal{S}_1} \hat{g}(i) - \min_{i \in \mathcal{S}_1} \hat{g}(i),$$

go to step 4). Otherwise, set $m = m + 1$ and return to step 2).
4) Compute $\hat{\eta}^{\delta_n}$ as

$$\hat{\eta}^{\delta_n} = \frac{1}{2}\left[\max_{i \in \mathcal{S}_1} (\hat{g}_{m+1}(i) - \hat{g}_m(i)) + \min_{i \in \mathcal{S}_1} (\hat{g}_{m+1}(i) - \hat{g}_m(i))\right].$$

If $|\hat{\eta}^{\delta_n}| \le \epsilon$, go to step 5). Otherwise, let

$$\delta_{n+1} = \delta_n + \alpha \hat{\eta}^{\delta_n}, \quad \text{with} \quad \alpha = 1/B_u$$

set $n = n + 1$, $m = m + 1$ and return to step 2).
5) Obtain a policy ϕ^*. For any $i \in \mathcal{S}_1$, choose

$$a^{\phi^*}(i) \in \arg\min_{a \in \mathcal{A}(i)} \left\{ H_f(i,a) - \delta_n H_1(i,a) + \sum_{j \in \mathcal{S}_1} \tilde{p}^a(i,j) \hat{g}_m(j) \right\}.$$

This algorithm is guaranteed to converge as summarized in the next theorem.

Theorem 3: Suppose $\mathcal{L}^*$ is an optimal policy of the original MDP (2). For the incremental value iteration algorithm, we have the following.

a) $|\delta_{n+1} - \eta^{\mathcal{L}^*}| \le \beta |\delta_n - \eta^{\mathcal{L}^*}|$, where $\beta = \max\{\sigma/(2\epsilon - \sigma), 1 - (B_l/B_u)(1 - (\sigma/(2\epsilon - \sigma)))\}$.
b) The algorithm terminates in a finite number of iterations.
c) The policy ϕ^* satisfies $\eta^{\phi^*} \le \eta^{\mathcal{L}^*} + (\epsilon + 3\sigma/2)/B_l$.

Proof:

a) Let $\tilde{\phi}^*_{\delta_n}$ be an optimal policy for the aggregated MDP with performance function (11) defined by δ_n. The estimation $\hat{\eta}^{\delta_n}$ calculated in step 4) satisfies (see [3, p. 370])

$$\left|\hat{\eta}^{\delta_n} - \tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}\right| \le \sigma/2, \quad \text{and} \quad \left|\hat{\eta}^{\delta_n} - \tilde{\eta}^{\phi^*}\right| \le \sigma/2. \tag{25}$$

Before the algorithm stops, the criterion in step 4) is not met, i.e., $|\hat{\eta}^{\delta_n}| > \epsilon > \sigma$. Therefore, $\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}} - \sigma/2 \le \hat{\eta}^{\delta_n} \le \tilde{\eta}^{\tilde{\phi}^*_{\delta_n}} + \sigma/2$, and

$$\left|\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}\right| \ge \epsilon - \frac{\sigma}{2}. \tag{26}$$

Thus, $\eta^{\tilde{\phi}^*_{\delta_n}}$ has a same sign with $\hat{\eta}^{\delta_n}$. In addition

$$\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}\left[1 - \frac{\sigma}{2\epsilon - \sigma}\right] \le \hat{\eta}^{\delta_n} \le \tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}\left[1 + \frac{\sigma}{2\epsilon - \sigma}\right] \quad \text{for } \hat{\eta}^{\delta_n} > 0,$$
$$\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}\left[1 + \frac{\sigma}{2\epsilon - \sigma}\right] \le \hat{\eta}^{\delta_n} \le \tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}\left[1 - \frac{\sigma}{2\epsilon - \sigma}\right] \quad \text{for } \hat{\eta}^{\delta_n} < 0.$$

It follows that

$$1 - \frac{\sigma}{2\epsilon - \sigma} \le \frac{\hat{\eta}^{\delta_n}}{\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}} \le 1 + \frac{\sigma}{2\epsilon - \sigma}.$$

In view of (22), we obtain

$$\frac{B_l}{B_u}\left(1 - \frac{\sigma}{2\epsilon - \sigma}\right) \le \frac{\hat{\eta}^{\delta_n}}{\eta^{\mathcal{L}^*} - \delta_n} \frac{1}{B_u} \le 1 + \frac{\sigma}{2\epsilon - \sigma}.$$

Since

$$\frac{\delta_{n+1} - \eta^{\mathcal{L}^*}}{\delta_n - \eta^{\mathcal{L}^*}} = \frac{\delta_n - \eta^{\mathcal{L}^*} + \hat{\eta}^{\delta_n}/B_u}{\delta_n - \eta^{\mathcal{L}^*}} = 1 - \frac{\hat{\eta}^{\delta_n}}{\eta^{\mathcal{L}^*} - \delta_n} \frac{1}{B_u}$$

TABLE I
ALGORITHM COMPARISONS BY THE REPLACEMENT PROBLEM

Algorithm	PI	FVI	IVI
Storage requirement	1.4 MB	2.4 KB	2.4 KB
Number of Iterations	3	103	19^{α}
CPU time per iteration	7.8 Sec	4.4 Sec	4.4 Sec

PI: Policy Iteration in [1].
FVI: Fractional cost Value Iteration in [4] with adjusted parameter and error tolerance 0.001.
IVI: Incremental Value Iteration; $\delta_0 = 10$, error tolerances: $\epsilon = 0.001$, $\sigma = 0.0001$.
$^{\alpha}$ The total number of iterations after solving four standard MDPs.

it follows that

$$-\frac{\sigma}{2\epsilon-\sigma} \le \frac{\delta_{n+1}-\eta^{\mathcal{L}^*}}{\delta_n-\eta^{\mathcal{L}^*}} \le 1-\frac{B_l}{B_u}\left(1-\frac{\sigma}{2\epsilon-\sigma}\right), \text{ and}$$

$$\left|\frac{\delta_{n+1}-\eta^{\mathcal{L}^*}}{\delta_n-\eta^{\mathcal{L}^*}}\right| \le \max\left\{\frac{\sigma}{2\epsilon-\sigma}, 1-\frac{B_l}{B_u}\left(1-\frac{\sigma}{2\epsilon-\sigma}\right)\right\}.$$

b) For the algorithm, the condition (26) must hold during iterations. However, $|\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}|$ decreases geometrically as $|\delta_n - \eta^{\mathcal{L}^*}|$ by a) and (23). Thus the algorithm terminates after a finite number of iterations. For each of the iterations, steps 2) and 3) are standard value iteration, which converges under the assumption that all stationary policies are unichain and that every optimal policy has an aperiodic transition matrix (see [3, p. 370]). Therefore, the ergodicity assumption used by time aggregated MDPs and fractional cost MDPs is sufficient to guarantee the convergence of our algorithm. The assumption is much weaker than that required by the algorithm in [4]: a state is admissible with a positive probability from any state under any action. Notice that such a positive probability is crucial for that algorithm in the aspects of implementation, convergence and optimality (see [4, eqs. (11), (13), and (15)], respectively). In contrast, our algorithm has no such strong requirement.

c) By (25) and the stopping criterion $|\hat{\eta}^{\delta_n}| \le \epsilon, |\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}| \le \epsilon + \sigma/2$ and $|\tilde{\eta}^{\phi^*}| \le \epsilon + \sigma/2$. When $\delta_n \le \eta^{\mathcal{L}^*}$, by (15)

$$\eta^{\phi^*} = \delta_n + \frac{\tilde{\eta}^{\phi^*}}{\bar{n}^{\phi^*}} \le \eta^{\mathcal{L}^*} + \frac{\tilde{\eta}^{\phi^*}}{\bar{n}^{\phi^*}} \le \eta^{\mathcal{L}^*} + \frac{\epsilon+\sigma/2}{B_l}. \tag{27}$$

When $\delta_n \ge \eta^{\mathcal{L}^*}$

$$\eta^{\phi^*} = \delta_n + \frac{\tilde{\eta}^{\phi^*}}{\bar{n}^{\phi^*}} = \eta^{\mathcal{L}^*} - \frac{\tilde{\eta}^{\mathcal{L}^*}}{\bar{n}^{\mathcal{L}^*}} + \frac{\tilde{\eta}^{\phi^*}}{\bar{n}^{\phi^*}}.$$

In step 3), $sp(\hat{g}_{m+1} - \hat{g}_m) \le \sigma$ implies $\tilde{\eta}^{\phi^*} - \tilde{\eta}^{\tilde{\phi}^*_{\delta_n}} \le \sigma$ ([3]). Therefore

$$\eta^{\phi^*} \le \eta^{\mathcal{L}^*} - \frac{\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}}{\bar{n}^{\mathcal{L}^*}} + \frac{\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}} + \sigma}{\bar{n}^{\phi^*}}.$$

By b) of Theorem 1, $\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}} \le 0$. Thus

$$\eta^{\phi^*} \le \eta^{\mathcal{L}^*} - \frac{\tilde{\eta}^{\tilde{\phi}^*_{\delta_n}}}{\bar{n}^{\mathcal{L}^*}} + \frac{\sigma}{\bar{n}^{\phi^*}} \le \eta^{\mathcal{L}^*} + \frac{\epsilon+3\sigma/2}{B_l}. \tag{28}$$

Combining (27) and (28), the conclusion follows. □

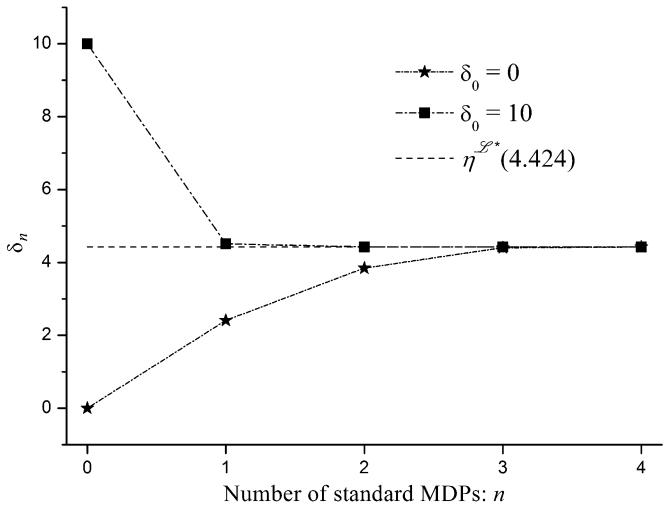

Fig. 1. δ_n approach $\eta^{\mathcal{L}^*}$ from $\delta_0 = 0, 10$ after solving 4 standard MDPs.

V. NUMERICAL TESTING

A multicomponent replacement problem is tested to compare the storage and computational requirements of the three algorithms, i.e., Policy Iteration (PI) [1], Fractional cost Value Iteration (FVI) [4] and our Incremental Value Iteration (IVI). The problem is sketched as follows. An asset consists of several components. A component must be replaced if it expires after running for a predefined lifetime or fails. Replacing any component incurs a common setup cost besides new component cost. To share the setup cost, those components that are close to expiration may also be replaced by opportunity. The problem is to minimize the average cost through proper joint replacement decisions. Variations of the problem can be found in [2]. The state is the combinations of component status (expired, failed or running with certain remaining lives). For illustration purpose, consider an asset with three components whose new lifetimes are all 10. The setup cost and the cost of a component are set as 10. Each component may fail with probability 0.01 at any time unit. Because of such independencies between component lifetime and failure rate, replacement decisions only need to be made for the states (i.e., $\mathcal{S}_1$) with component expirations or failures. Therefore, applying time aggregation approach cuts the number of states from 1331 to 602. The $H_f^{\mathcal{L}}(i)$, $i \in \mathcal{S}_1$, is just the cost at i because no cost is encountered until next replacement. The $\tilde{P}^{\mathcal{L}}$ can be calculated by (4) or directly based on the relations among states. The three algorithms are tested on Windows, PIV 2.4 GHz, 512 MB RAM and MATLAB 6.5. The results are presented as Table I and Fig. 1.

Two value iteration algorithms, FVI and IVI, may only store the potential vector $\tilde{g}^{\mathcal{L}}$ with $|\mathcal{S}_1|$ elements which are updated (e.g., according to (24) for IVI) with computational order $|\mathcal{A}||\mathcal{S}_1|^2$ during iterations. In contrast, policy iteration requires extra storage and computations with rough orders $|\mathcal{S}_1|^2$ and $|\mathcal{A}||\mathcal{S}_1|^2 + |\mathcal{S}_1|^3$, respectively, to compute the steady-state probabilities and solve the Poisson equation (12) at each iteration [1]. For this three-component example, Table I shows that both value iteration algorithms take less CPU time at each iteration and need less storage than policy iteration. If the number of compents is larger, policy iteration would not work. Consider for example a six-component problem with new lifetime 10. The policy iteration will become intractable because $|\mathcal{S}_1|$ is of the order $O(10^6)$, the number of elements to be stored is of the $O(10^{12})$ (more than 1 Terabyte if an element requires 4 Bytes memory), and the computation is of the order $O(10^{18})$. However, value iteration may still doable because the storage requirement is about several Megabytes and the computational burden is also

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