A Multiple-Derivative and Multiple-Delay Paradigm for Decentralized Controller Design: Introduction using the Canonical Double-Integrator Network

Yan Wan   Sandip Roy   Ali Saberi   Anton Stoorvogel

August 10, 2008

Abstract

We are engaged in a major effort to design decentralized controllers for modern networks, that is fundamentally based on 1) applying feedback of multiple derivatives of local observations and 2) implementing these derivative feedbacks using multiple-delay controllers. Here, we fully motivate and introduce the design paradigm in the context of a canonical sensing-network model, namely a network of saturating double integrators with general sensing topology that is subject to measurement delays. In this context, we illustrate that our design paradigm yields practical high-performance (in particular, group pole-placement) decentralized controllers that exploit the network topology while distributing the complexity and actuation requirements among the agents.

1 Introduction

Decentralized feedback systems have long been of interest to the controls community [1–3]. In recent years, research in decentralized control has been re-invigorated by interest in such applications as cooperative control of autonomous vehicle teams, data fusion in sensor networks, and virus-spreading control, among others, (see the overviews [4–7], see also, e.g., [8–10]). In particular, the novel characteristics of these *sensing-agent networks* (networks of highly-limited autonomous agents with distributed communication/sensing capabilities [6]) has brought about a focus on understanding the role played by a network's topology in permitting stabilization and high-performance control. This focus has again made clear that very little is known about *designing* high-performance controllers for decentralized systems—even for the very specially structured sensing-agent networks—and hence new tools for design are badly needed.

We are engaged in a major effort to design stabilizing and high-performance yet practical controllers for decentralized systems, that is fundamentally based on 1) locally using feedback of multiple *derivatives* of the observation and 2) using multiple-delay control schemes to implement these multiple-derivative controllers. We show that this new methodology is capable of addressing many of the complexities that are common to modern decentralized systems (such as sensing-agent networks), including very general observation topologies, saturation nonlinearities, and inherent network delays. We shall describe aspects of this systematic methodology for design in several installments [11, 17]. In this paper, we fully motivate and introduce the design methodology using a canonical but very widely applicable sensing-agent network model, namely a network of double-integrator agents with general sensing/communicating topology (e.g. [8,12]). The complementary installments demonstrate application to uniform rank and more general decentralized plant models (including for modern infrastructure networks) [11], and flesh out the implementation of multiple-derivative feedback using multiple-delay controllers.

Given the long history of decentralized control, the reader may well wonder why new techniques are needed for decentralized controller design. In fact, the study of sensing-agent networks, as well as certain infrastructure networks such as air traffic management systems [13] and electric power systems [14], has made it clear that *a single agent cannot possibly provide the actuation or complexity required to control the whole network, and further the controllers must exploit the network topology to cooperatively achieve performance requirements*. Unfortunately, the bulk of the traditional decentralized control theory views the network as a disturbance that must be dominated by the local dynamics [3], and hence does not permit design of controllers that exploit the network topology.

The seminal work of Wang and Davison [1] does make the role played by the network explicit, in that it gives necessary and sufficient conditions for stabilization of decentralized systems based on *fixed modes* (see also, e.g., [2, 15, 16]). Their methodology is very much applicable to modern networks, and we have used it to address the foundational problem of determining whether a sensing-agent network can be stabilized [8]. Unfortunately, Wang and Davison's perturbation-based approach does not permit *constructive design* of practical high-performance or even stabilizing controllers. While several works have extended [1] toward allowing eigenvalue placement (and hence high performance) in addition to stabilization, these approaches essentially concentrate the complexity and extent of actuation/observation at a single agent, and hence also are unsuitable for our applications [3]. For these reasons, new tools for decentralized controller design are critically needed.

In this document, we develop a multiple-derivative and multiple-delay paradigm for controlling decentralized systems. Fundamentally, the derivatives of local observations provide the local controllers with information about the entire network's state, and so permit control. To develop practical implementations of these multiple-derivative controllers for modern (e.g. sensing-agent) networks, we pursue multiple-delay approximations for the multiple-derivative controllers (i.e., feedback controls where the actuation signals are combinations of multiple delayed observations), see [17] for further development of multiple-delay controllers. This use of delayed observations may at first seem surprising since delays often serve to destabilize feedback control systems [18], but it is also well known that properly-selected delays can be used effectively in control [19, 20]. This derivative/delay paradigm is a very natural one for decentralized systems, for which centralized notions of state estimation fail, and hence delays/derivatives provide the only known approach to finding the global state from local observations. What is surprising is that we can achieve not only stabilization but effective pole placement, while using only one more delayed observation (or one higher derivative) than is needed for centralized control. We thus are able to construct fully decentralized controllers with quite low complexity and distributed actuation effort. In this paper, we conceptualize and illustrate the delay-based decentralized control paradigm, using as a canonical example the double-integrator-network model.

Decentralized systems, and in particular sensing-agent networks, are strongly impacted both by constraints on the agents and network limitations and variations. An essential advantage of our delay-based control methodology is its effectiveness even in the presence of these harsh constraints/limitations. Specifically, **actuator saturation** nonlinearities are ubiquitous in sensing-agent network applications [8, 9]. While controller design under saturation has been extensively studied for centralized systems [21], design under saturation for decentralized systems is wholely unknown (see [22] for partial *existence* conditions). In fact, our multiple-derivative/delay control scheme provides a natural avenue for design under actuation saturation. Further using low-gain ideas, we can naturally design multiple-delay controllers that stabilize networks with actuator saturation. Also, network communications/sensing are always subject to delays, and so controlling networks with inherent delays is critically important. Since our control strategy systematically uses delayed observations, it is eminently suited for networks with inherent delays. In particular, we show that networks with arbitrary and inhomogeneous delays can be stabilized with a low-gain controller. These results for networks with saturation and delay indicate the wide applicability of our methodology for practical controller design.

We stress here that the delay-based control methodology is applicable to general linear time-invariant decentralized control systems, and so the reader may wonder why we have chosen to introduce the methodology using only a canonical example. In fact, focusing on the double-integrator network permits a clearer and simpler presentation for two reasons: 1) it permits a full characterization of the derivative-based and hence delay-based controller's performance and implementation from first principles (Sections 2 and 3), without requiring the complicated *special coordinate basis* (see [23] and also [11]) , and 2) the time-scaling properties of the double-integrator network permits simple design of low-gain multiple-delay controllers (Section 4). We feel strongly that presenting results in this simpler context allows us to expose the conceptual underpinnings of derivative/delay-based decentralized control, and to clearly develop the (rather intricate) tools for analyzing multiple-delay controllers. We also note that sensing-agent networks, and in particular double-integrator networks, are of wide current interest [6, 8] and so deserve an explicit treatment.

2 Controlling the Double-Integrator Network

In this section, we illustrate our methodology of using multiple derivative feedbacks to achieve stabilization and high-performance control, in the context of a decentralized **double inte-**

grator network (see [8]). In this simple network model, each agent has a double-integrator internal dynamics, and observes only a linear combination of the states of some agents. Although simple, this network model is widely applicable, including for various autonomous-vehicle control and sensor-networking tasks [7, 8, 12]. We present our controller design for this simple but very widely-applicable decentralized network, to highlight the conceptual foundation for the methodology. Specifically, we show that, using linear feedback of derivatives of observations up to order 2 for each agent, we can stabilize the double integrator network. Moreover, we can place groups of poles at arbitrary locations or in desirable ranges using high gain control. The fundamental concept underlying this design is that feedback of derivatives of the output up to the relative degree of the local plant (2 in our case) gives each agent enough information about the global state to permit high-performance control through, essentially, plant inversion; we notice that one more derivative is needed then for centralized control of the plant, see the literature on asymptotic time-scale and eigenstructure assignment (ATEA design) and our recent application of it to multiple-delay control of centralized plants [20,24]. In this section, we first present the controller design (Section 2.1), and then give a conceptual discussion of the design method and its characteristics (Section 2.2).

2.1 Multiple-Derivative Controller Design

Formally, consider a linear time-invariant (LTI) system consisting of n double-integrator agents, i.e. described by

$$\begin{aligned} \ddot{\mathbf{x}}(t) &= \mathbf{u}(t) \\ \mathbf{y}(t) &= G\mathbf{x}(t), \end{aligned} \tag{1}$$

where $\mathbf{x}(t) \in \mathcal{R}^n$ represents the positions of the n agents, $\mathbf{u} \in \mathcal{R}^n$ and $\mathbf{y} \in \mathcal{R}^n$ are the inputs and observations respectively, and matrix $G = \{G_{ij}\}_{n \times n}$. Note that each agent i has only one input u_i and makes only one observation y_i, which is a linear combination of the positions of other agents, i.e., $y_i = [G_{i1}, ..., G_{in}]\,\mathbf{x}$. Further, each agent i has its own local feedback control law, which constructs its input u_i from its local observation y_i. We refer to this system as a **double-integrator network** (see [8]).

The condition to stabilize such a network using a linear time-invariant controller is that G has full rank (from [8], based on Wang and Davison's classical existence result [1]). Unfortunately, the existence condition seemingly does not translate to a simple controller design: the system can not always be stabilized with a static decentralized feedback controller $u(t) = Ky(t)$ (K diagonal), nor can it always be stabilized with position and velocity feedback (i.e. using a control law of the form $\mathbf{u}(t) = K_1 G\mathbf{x}(t) + K_2 G\dot{\mathbf{x}}(t)$, K_1 and K_2 diagonal) [8]. However, by introducing one more derivative to the feedback control—specifically by using the control law $\mathbf{u}(t) = k_1 k_3 \mathbf{y}(t) + k_2 k_3 \dot{\mathbf{y}}(t) + k_3 \ddot{\mathbf{y}}(t)$, where k_1, k_2, k_3 are some properly chosen scalars — it turns out that we suddenly gain the ability to achieve stabilization and high performance (in particular, a "group" pole placement, where groups of n poles are placed at desirable locations). It is valuable to note that all agents have the same gains k_1, k_2 and k_3:

one does not even need to employ agent-specific gains for stabilization and high-performance control.

We can design this multiple-derivative-based control law using a simple algorithm. We first describe the algorithm, and then formally show that the two tasks (stabilization and high-performance control) can be completed using the multiple-derivative-based controller.

Algorithm The following is the algorithm for designing the multiple-delay-based controller:

1) Choose two constants, say k_1 and k_2, such that the roots of $\lambda^2 + k_1\lambda + k_2$ are at desirable locations.

2) Choose k_3 sufficiently large, and apply the control law

$$\mathbf{u} = k_1k_3\mathbf{y} + k_2k_3\dot{\mathbf{y}} + k_3\ddot{\mathbf{y}} \tag{2}$$

We will show that the algorithm yields not only stabilizing controllers, but ones with closed-loop poles near to the roots of $\lambda^2 + k_1\lambda + k_2$. The above algorithm for designing the derivative-based controller, and the justification that it achieves stabilization/performance goals, is essentially based on using the second derivative of the observation in feedback with high gain (k_3 large) to effectively permit local control of agents' states.

In Theorem 1, we state the main result concerning stabilization of the double integrator network using the multiple-derivative-based controller (Equation 2).

Theorem 1 *Consider the double-integrator network described in (1), where G is non-singular. The network can be stabilized using the multiple-derivative control law (Equation 2) with k_1, k_2 and k_3 satisfying: $a_{i1} > 0$, $\frac{a_{i1}a_{i2}-a_{i3}}{a_{i1}} > 0$, $\frac{a_{i1}^2a_{i4}+a_{i3}(a_{i3}-a_{i1}a_{i2})}{a_{i3}-a_{i1}a_{i2}} > 0$ and $a_{i4} > 0$ for all i, where $a_{i1} = -2Re(\frac{k_2k_3\lambda_i}{1-k_3\lambda_i})$, $a_{i2} = -2Re(\frac{k_1k_3\lambda_i}{1-k_3\lambda_i}) + \left|\frac{k_2k_3\lambda_i}{1-k_3\lambda_i}\right|^2$, $a_{i3} = 2Re(\frac{k_1k_3\lambda_i}{1-k_3\lambda_i})Re(\frac{k_2k_3\lambda_i}{1-k_3\lambda_i}) + 2Img(\frac{k_1k_3\lambda_i}{1-k_3\lambda_i})Img(\frac{k_2k_3\lambda_i}{1-k_3\lambda_i})$, $a_{i4} = \left|\frac{k_1k_3\lambda_i}{1-k_3\lambda_i}\right|^2$, and λ_i is the ith eigenvalue of G. As a special case, if G has real eigenvalues, the network can be stabilized with k_1, k_2 and k_3 satisfying: $k_1 > 0$, $k_2 > 0$ and $k_3 > \frac{1}{min(\lambda(G)>0)}$ (where $min(\lambda(G) > 0)$ denotes the minimum positive eigenvalue of G) .*

Proof: We study the closed-loop poles of the system using the control law (Equation 2). The state-space of the closed loop system thus is $\dot{\mathbf{X}} = A_c\mathbf{X}$, where $\mathbf{X} = \begin{bmatrix}\dot{\mathbf{x}} \\ \mathbf{x}\end{bmatrix}$ and $A_c = \begin{bmatrix}(I-k_3G)^{-1}k_2k_3G & (I-k_3G)^{-1}k_1k_3G \\ I & 0\end{bmatrix}$. Denote $\begin{bmatrix}x_1 \\ x_2\end{bmatrix}$ as the right eigenvector of A_c, we have $A_c\begin{bmatrix}x_1 \\ x_2\end{bmatrix} = \lambda\begin{bmatrix}x_1 \\ x_2\end{bmatrix}$, which implies that $x_1 = \lambda x_2$, and $\lambda(I-k_3G)^{-1}k_2k_3Gx_2 + (I-k_3G)^{-1}k_1k_3Gx_2 = \lambda x_1$. The latter yields: $\lambda k_2k_3Gx_2 + k_1k_3Gx_2 = \lambda^2(I-k_3G)x_2$, or $(\lambda k_2k_3 + k_1k_3 + \lambda^2k_3)Gx_2 = \lambda^2x_2$. This means that x_2 must be an eigenvector of G with a eigenvalue, say λ_i. Hence, we have $(\lambda k_2k_3+k_1k_3+\lambda^2k_3)\lambda_i = \lambda^2$, or $\lambda^2 - \frac{k_2k_3\lambda_i}{1-k_3\lambda_i}\lambda - \frac{k_1k_3\lambda_i}{1-k_3\lambda_i} = 0$. The closed-loop poles are the roots of the characteristic equations $\lambda^2 - \frac{k_2k_3\lambda_i}{1-k_3\lambda_i}\lambda - \frac{k_1k_3\lambda_i}{1-k_3\lambda_i} = 0$, for all i. Placing the roots of $\lambda^2 - \frac{k_2\lambda_i}{1-k_3\lambda_i}\lambda - \frac{k_1\lambda_i}{1-k_3\lambda_i} = 0$ in the Open Left Half Plane (OLHP) is equivalent to placing the zeros of $(\lambda^2 - \frac{k_2 k_3 \lambda_i}{1-k_3\lambda_i}\lambda - \frac{k_1 k_3 \lambda_i}{1-k_3\lambda_i^*})(\lambda^2 - \frac{k_2 k_3 \lambda_i^*}{1-k_3\lambda_i^*}\lambda - \frac{k_1 k_3 \lambda_i^*}{1-k_3\lambda_i^*}) = 0$ into the OLHP, since the latter has two conjugate complex root pairs, each pair specifying a root of the original equation. As a special case, when λ_i is real, the latter has two repeated roots corresponding to each root in the original equation. The Routh Criterion naturally leads to the conditions for stabilization. □

It is easy to check that the conditions on the gains in Theorem 1 can always be satisfied by choosing k_1 and k_2 positive and k_3 sufficiently large. Theorem 1 states that by introducing the derivatives of observations $\mathbf{y}(t), \dot{\mathbf{y}}(t)$ and $\dot{\mathbf{y}}(t)$ into the control law as in (2), the decentralized system can be stabilized whenever G has full rank. This result is significant in that it gives an explicit controller design for stabilization of a double integrator network, rather than only giving conditions for the existence of such a controller. We will see that this design far outperforms single-channel-based designs (Section 2.2).

Using derivatives in the control law also allows performance design, e.g., placing groups of closed-loop poles at pre-defined positions or within ranges.

Theorem 2 *Consider the double-integrator network described in (1), where G is non-singular. The closed-loop poles of the network can be placed arbitrarily near to any two pre-defined locations x_A and x_B (on the real axis or as a conjugate pair) using the multiple-derivative controller (Equation 2), by setting k_3 sufficiently large and choosing k_1 and k_2 such that $k_2 = -(x_A + x_B)$ and $k_1 = x_A x_B$.*

Proof: From the proof of Theorem 1, we know that the closed-loop roots are the zeros of the characteristic equations $\lambda^2 - \frac{k_2 k_3 \lambda_i}{1-k_3\lambda_i}\lambda - \frac{k_1 k_3 \lambda_i}{1-k_3\lambda_i} = 0$, for all i. Hence, when k_3 is sufficiently large, the coefficients of the characteristic equation approach the coefficients of the quadratic equation $\lambda^2 + k_2\lambda + k_1 = 0$. From the continuous dependence of roots on parameters, the closed-loop poles thus approach the roots of this characteristic equation. The result follows with just a little algebra. □

This theorem states that, by using sufficiently high gains, we can place all the closed-loop poles arbitrarily close to any two predefined locations, with n poles at each location. Moreover, one can see through the root locus that when we decrease k_3 from a high value, the poles that are originally close to each pre-defined location separate, with speeds dependent on k_1, k_2, k_3, and λ_i. Thus, by choosing proper k_1, k_2 and k_3, we can place the poles in specified ranges.□

Motivated by Theorem 2, we define **group pole placement** as the task of placing a group of poles of a decentralized system near (or within a range of) some pre-defined positions in the complex plane. Note that group pole placement is different from exact pole placement in that we only place sets of poles within a range or near a location, rather than design the exact location of each pole. However, group pole placement is a much stronger achievement than stabilization. Group pole placement design allows us to set such performance statistics as the dominant eigenvalue and dominant eigenvalue ratio [10], and further through an inverse-optimality argument the design can be shown to achieve phase-margin requirements [25]. The derivative-based controller permits us to achieve group pole placement, with the locations of the groups of poles and the closeness of the poles within each group depending on the values of the gains that we have chosen.

2.2 Discussion: Concepts and Comparisons

It is worthwhile to further discuss our multiple derivative controller, so as to better interpret how it works (Section 2.2.1) and compare it to existing approaches for decentralized control (Section 2.2.2).

2.2.1 A Structural Interpretation

Our design methodology is based on using multiple-derivative-control—and specifically using one higher derivative than is needed for centralized pole placement (see the work on ATEA design [24])—to achieve high-performance decentralized control. The use of derivative-based control for decentralized systems is sensible (broadly speaking), in that derivatives of linear-system outputs identify the global state and so should facilitate control at each channel of a decentralized system. What is surprising is that precisely as many derivatives as the relative degree of the local plant, or one more than is needed for centralized control, is sufficient to provide each agent with enough state information to permit stabilization and group pole placement. Here, we give some further conceptual discussion of this special characteristic of the multiple-derivative control.

Specifically, let us argue that our controller, which uses an "extra derivative", implicitly and distributedly provides each agent with the *local* state information and so permits simple control of n sets of "local" dynamics. To see why this is the case, notice that the positions can be found from the observations as $\mathbf{x} = G^{-1}\mathbf{y}$, and similarly the velocities can be found as $\dot{\mathbf{x}} = G^{-1}\dot{\mathbf{y}}$. Thus, if each agent can be given the statistics $\mathbf{h}_i^T\mathbf{y}$ and $\mathbf{h}_i^T\dot{\mathbf{y}}$, where $\mathbf{h}_i^T$ is the ith row of the matrix G^{-1}, then it has available the local position and velocity. However, the use of the extra derivative in the decentralized controller implicitly does exactly this. In particular, note that upon application of the multiple-derivative control, the closed-loop dynamics are $\ddot{\mathbf{x}} = (I-k_3G)^{-1}k_1k_3\mathbf{y}+(I-k_3G)^{-1}k_2k_3\dot{\mathbf{y}}$, where we have written the right-hand side in terms of the observation $\mathbf{y}$ rather than the state. Thus, $\ddot{\mathbf{x}} = k_1(\frac{1}{k_3} - G)^{-1}\mathbf{y} + k_2(\frac{1}{k_3} - G)^{-1}\dot{\mathbf{y}}$. For k_3 sufficiently large, $\ddot{x}_i \approx k_1\mathbf{h}_i^T\mathbf{y} + k_2\mathbf{h}_i^T\dot{\mathbf{y}} = k_1x_i + k_2\dot{x}_i$. That is, each agent is approximately feeding back its local state and derivative, i.e. locally using a proportional-derivative controller, to achieve performance requirements. Such local control of identical double-integrators is of course straightforward, and so we automatically infer the possibility for group pole placement. We notice that this approach is a fundamentally distributed one, in the sense that actions at each channel are together permitting computation of the local state and control using it.

The above discussion clarifies that, fundamentally, the derivative-based controller achieves high performance by distributedly providing each agent with local state information from the network observation. Several further points regarding this viewpoint are worthwhile:

a) We note the great difference of this approach to the traditional approach taken in decentralized control, where knowledge of local state is *assumed* and is used to dominate the network interactions rather than being meshed with them [3].

b) The above discussion clarifies that decentralized control is greatly simplified *whenever* the agents are provided with the statistics $\mathbf{h}_i^T\mathbf{y}$ and $\mathbf{h}_i^T\dot{\mathbf{y}}$, whether by derivative-based control or through another means. For instance, direct communication of appropriate observations so as to permit computation is an alternative.

c) This viewpoint motivates us to seek better characterizations of the matrix G^{-1}, in terms of the network topology codified in G. The relationship between the structure of the topology matrix G and that of its inverse is generally complicated. Even when the matrix G is sparse, G^{-1} may be dense, and hence we see that conceptually the statistics needed by each agent require observations from throughout the network (which are rather slickly provided to the agent through use of one higher derivative). These statistics are much simplified, or amenable to interesting interpretations, for special classes of topology matrices such as those with a slow-coherent structure. We leave it to future work to make precise the structure of G^{-1} for these special classes

2.2.2 Comparison with the Dominant Channel Approach

We have argued that our design is fundamentally different from the existing approaches for decentralized pole placement [2], in that it distributes the complexity and actuation among the channels (agents) rather than concentrating them at one channel. Here, we make explicit the advantage in complexity and actuation provided by our approach. Precisely, we show through an example that high complexity and large actuation may be needed in a single channel when the existing methods are used, as compared to when the multiple derivative controller is used. We recall that in the existing approaches, the entire network dynamics are made controllable and observable from a single channel through static feedback, and then a linear dynamic controller is implemented at this channel for pole placement (using standard method for centralized systems); it is this dominant channel approach that we will compare to our multiple-delay-based design.

First, let us compare the controller complexities for dominant channel-based approach and the multiple-derivative design, in the context of the double integrator network. For the multiple-derivative design, each agent requires precisely three signals (y_i, $\dot{y}_i$, and $\ddot{y}_i$), which are linearly combined to generate the actuation signal, regardless of the network topology; these signals can be approximated arbitrarily well using either 3 variously-delayed observations or a lead compensator with two poles, see the implementation section (Section 3) for further details. On the other hand, as we shall show below, the dominant channel approach necessitates use of a dynamic controller of order $2n$ at a single agent for certain network topologies. The comparison is formalized in the following theorem:

Theorem 3 *Consider a double-integrator network with full graph matrix* $G = \begin{bmatrix} & 1 & & \\ & & \ddots & \\ & & & 1 \\ 1 & & & \end{bmatrix}$. *If the dominant channel approach for decentralized pole placement is used, then one agent requires a controller that has dynamic order $2n$. In contrast, the multiple derivative design* *requires a linear feedback of the observation and its first two derivatives, which can be implemented using three delayed observations or else a dynamic controller of order 2 at each channel for group pole placement.*

Proof: The result is automatic for the multiple-derivative design.

In the dominant channel approach, a static linear feedback is applied to each agent, and then this closed-loop system is controlled from any single channel (which we can choose WLOG to be the channel n, from symmetry). Specifically, let us consider applying the controls $u_i = p_i y_i$ for channels $1, \ldots, n-1$ and the control $u_n = p_n y_n + w_n$, and then consider designing a further feedback in channel n (from y_n to w_n). With a little algebra, we find that the (SISO) transfer function from w_n to y_n for the example full graph matrix G is $\frac{1}{s^{2n}+P}$, where $P = p_1 p_2 \ldots p_n$. We claim that feedback control of this plant for the purpose of pole placement requires a dynamic controller of order $2n$. To see why, consider applying a strictly proper linear dynamic controller $\frac{r(s)}{q(s)}$ to the plant. Then notice that the characteristic polynomial of the closed-loop system is $\gamma(s) = s^{2n}q(s) + Pq(s) + r(s)$. If the degree of $q(s)$ is m, notice that the only non-zero coefficients in the characteristic polynomial are those for the terms $s^{2n}, \ldots, s^{2n+m}$, and $1, \ldots, s^m$. Then we require $m \geq 2n-1$ to achieve stability, let alone pole placement. In fact, even a controller of order $m = 2n-1$ is not sufficient for pole placement since the ratio between the coefficients of s^{4n-1} and s^{2n-1} is fixed. Thus, a controller of order $2n$ is needed for pole placement. $\square$

Let us also, through an example, compare the actuation needed by the agents for the multiple-derivative design and the dominant-single-channel approach. In particular, we consider a double-integrator network with $n = 5$ agents, and the cyclic graph G from Theorem 3. In this example, we assume that the agents are initially at position 1 and have velocity 0.

For the multiple-derivative design, we use parameter $k_1 = 1$, $k_2 = 2$, and $k_3 = 25$. We notice that this design aims to place all the eigenvalues at $s = -1$. It turns out that the gain k_3 has been chosen large enough that, in fact, the eigenvalues are to the left of $s = -0.8$. We note that the actuation signal for each agent is identical in this example. This common actuation signal is shown in Figure 1.

For the dominant single-channel approach, we initially apply unity static gains at each agent. We then place all the poles at -0.8 using dynamic feedback at (WLOG) agent 5, where we have conservatively chosen to place the poles at -0.8 to ensure that the comparison of the two controllers is fair. The actuation signal for agent 5 for this controller is shown in Figure 1. We notice that the magnitude of the required actuation signal is very large compared to that for the multiple-delay-based design (roughly, by a factor of 10^4 over a single agent's actuation in the multiple-delay design). Very similar results are obtained for other initial conditions, and for designs where randomly-chosen static gains are used rather than uniform ones. This need for large actuation is not surprising: in particular, a large effort is needed to move Agent 1 using the controller at Agent 5 (and in fact this also cause large swings in the location of Agent 1 in part of the transient).

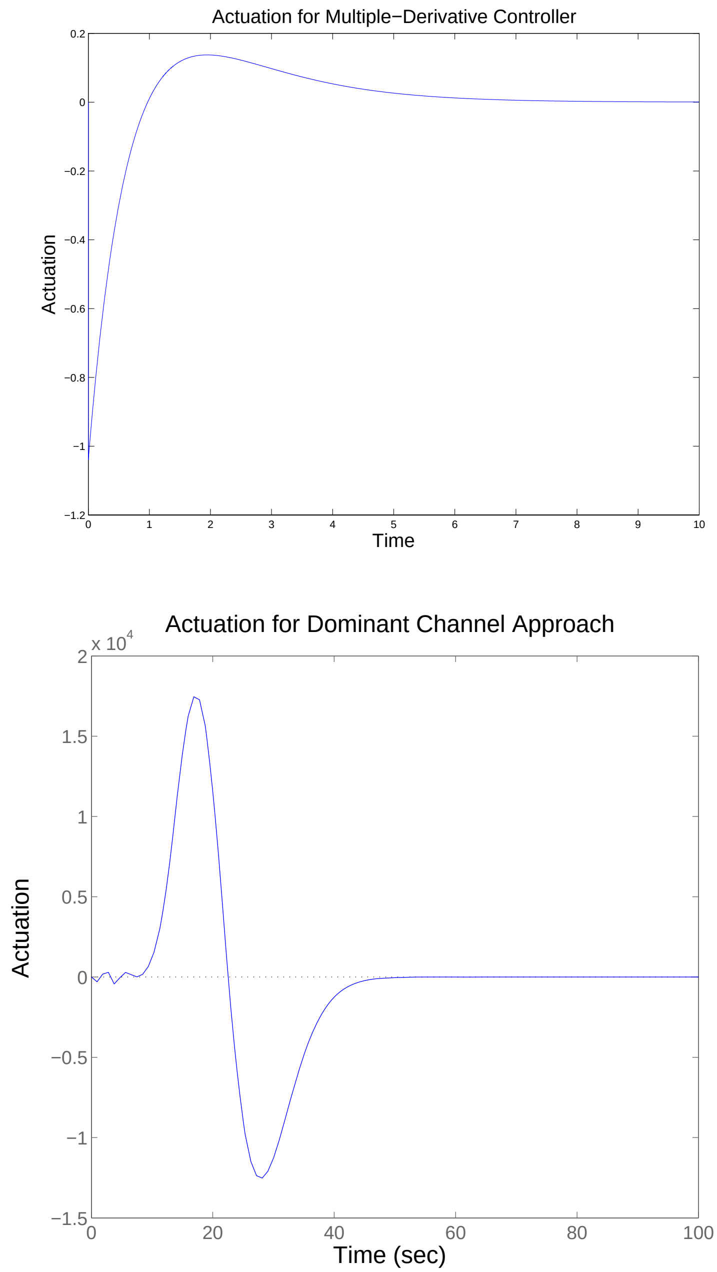

Figure 1: *Top:* Actuation needed for multiple derivative control. *Bottom:* Actuation needed for control through a single dominant channel.

3 Implementation Issues: Overview

So far, using the double-integrator network as a canonical example, we have introduced a philosophy for decentralized control based on each channel (agent) using multiple derivatives of its local measurements. Here, we shall discuss the realistic implementation of such multiple-derivative controllers, with a particular focus on *multiple-delay control* (see e.g. [17, 19, 20] for background). In the interest of space, our development here is only an overview, with a focus on providing the user with working controllers for typical network topologies, discussing briefly the implementation for general topologies, and explaining to the reader the richness of this multiple-derivative-control task. Further details can be found in our related works [17, 20].

Fundamentally, our control scheme for the double integrator requires that each agent obtain and feed back the first two derivatives of its observations, or in other words the derivatives up to the relative degree of the local plant. Traditionally, derivatives of observations are obtained through either *1)* explicit measurement (e.g., measurement of vehicle velocities) or *2)* approximation of the derivative with a proper transfer function, e.g. through use of lead compensation or, less commonly, multiple-delay approximations [17, 19, 20]. Approximation of derivatives up to one less than the relative degree of a plant can be done systematically, so that a feedback system using these approximations has performance arbitrarily close to one actually using the derivatives. That is, the finite closed-loop poles upon use of the approximating controller can be made to approach the ones for the derivative-based control system, while all other poles[1] are driven to $-\infty$.

On the other hand, approximations of derivatives of order *equal* to the relative degree of the plant in the feedback can, if improperly used, produce highly unstable spurious dynamics (while in many other cases harmlessly replicating the derivative-based control). This possibility for instability essentially results from the effective delay that is imposed by any implementable approximation combined with the possible non-continuity of this derivative, at least at an initial time, see e.g. [26] for a treatment of the phenomenon. Luckily, for the double-integrator-network, even the most basic multiple-delay-based or lead-compensation-based control implementations are successful for many typical network topologies. In the cases where these basic schemes are not successful, a variety of alternatives are available, including *1)* clever selection of the approximation used by each agent, *2)* adaptation to eliminate potential unstable dynamics, and *3)* explicit use of the relationship between the highest derivative and the lower derivatives and input through communication of a few observations/inputs between agents. We have shown that the first of these alternatives is sufficient for addressing approximation for arbitrary network topologies [17], but each alternative may have certain advantages/disadvantages in implementation and deserves further study. It is this wealth of alternative approaches that yields a rich problem space in the arena of approximating multiple-derivative-based control. Here, let us describe the basic approximation scheme and delineate the network topologies to which it applies (without proof in the interest

[1]Notice that the multiple-delay-controlled system is infinite-dimensional and has an infinite number of such poles.

of space), and then ruminate on the alternatives.

As noted above, either lead compensation or multiple-delay-based designs can be used. Since networks are very often subject to intrinsic delays, designs using multiple delayed observations are naturally applicable to network tasks, and so we shall focus on multiple-delay-based controls. A basic approach to implement the multiple-derivative controller is to approximate derivatives with identical delays, e.g., the first derivative $\dot{y}(t)$ can be approximated as $\frac{y(t)-y(t-\tau)}{\tau}$, where τ is a small delay. Higher derivatives can similarly be approximated by interpolating the observation with a polynomial [19,20]. Here, we consider using the decentralized control law $\mathbf{u}(t) = \alpha_1 \mathbf{y}(t-\tau_1) + \alpha_2 \mathbf{y}(t-\tau_2) + \alpha_3 \mathbf{y}(t-\tau_3)$, where $0 \le \tau_1 < \tau_2 < \tau_3$, α_1, α_2, and α_3 are some properly chosen scalars, to stabilize the double integer network. The controller parameters can be selected using the following simple algorithm.

1) Choose two constants, say k_1 and k_2, such that the roots of $\lambda^2 + k_1\lambda + k_2$ are at desirable locations.

2) Choose a set of times $0 \le \bar{\tau}_1 \le \bar{\tau}_2 \le \bar{\tau}_3$, which will specify the relative spacing in time between the delayed measurements used by the controller, see Equation 3 below. The delays $\bar{\tau}_1$, $\bar{\tau}_2$, and $\bar{\tau}_3$ need to be properly chosen so that a large k_3 does not introduce closed-loop poles in the ORHP (see [17] for the details).

3) Apply the control law:

$$\begin{aligned} \mathbf{u} = \frac{k_3}{\Delta}&\left[k_1\left(\bar{\tau}_2\bar{\tau}_3^2 - \bar{\tau}_2^2\bar{\tau}_3\right) - k_2\frac{\bar{\tau}_2^2-\bar{\tau}_3^2}{\epsilon} + 2\frac{\bar{\tau}_3-\bar{\tau}_2}{\epsilon^2}\right]\mathbf{y}(t-\epsilon\bar{\tau}_1) + \\ \frac{k_3}{\Delta}&\left[k_1\left(\bar{\tau}_3\bar{\tau}_1^2 - \bar{\tau}_3^2\bar{\tau}_1\right) - k_2\frac{\bar{\tau}_3^2-\bar{\tau}_1^2}{\epsilon} + 2\frac{\bar{\tau}_1-\bar{\tau}_3}{\epsilon^2}\right]\mathbf{y}(t-\epsilon\bar{\tau}_2) + \\ \frac{k_3}{\Delta}&\left[k_1\left(\bar{\tau}_1\bar{\tau}_2^2 - \bar{\tau}_2\bar{\tau}_1^2\right) - k_2\frac{\bar{\tau}_1^2-\bar{\tau}_2^2}{\epsilon} + 2\frac{\bar{\tau}_2-\bar{\tau}_1}{\epsilon^2}\right]\mathbf{y}(t-\epsilon\bar{\tau}_3), \end{aligned} \tag{3}$$

where $\Delta = \begin{vmatrix} 1 & \bar{\tau}_1 & \bar{\tau}_1^2 \\ 1 & \bar{\tau}_2 & \bar{\tau}_2^2 \\ 1 & \bar{\tau}_3 & \bar{\tau}_3^2 \end{vmatrix}$, k_3 is chosen sufficiently large, and ϵ is a sufficiently small number.

This multiple-delay controller is based on the approximation of the multiple-derivative controller $\mathbf{u} = k_1k_3\mathbf{y} + k_2k_3\dot{\mathbf{y}} + k_3\ddot{\mathbf{y}}$, i.e. the multiple-derivative controller that we showed in Section 2 to achieve group pole placement. From the results in [17], and using the fact that G can always be pre-scaled by a diagonal gain matrix in decentralized control, we recover that the delay-based controller (Equation 3) with $\mathbf{y} = KG\mathbf{x}$ is equivalent in a pole-placement sense to the multiple-derivative control[2] whenever the eigenvalues of KG are in the OLHP. Matrices G whose eigenvalues can be placed in the OLHP by a diagonal scaling include all those that have a sequence of n nested principal minors with full rank (e.g., [8]). Positive definite matrices such as grounded Laplacian matrices and diagonally-dominant matrices, which are common in many applications such as autonomous vehicle control and sensor network management ones, of course fall in this class (and require only pre-scaling by $-I$).

[2]Notice that we have excluded this pre-scaling up to this point for the sake of simplicity of presentation.

Let us complete the discussion with a claim. We expect that derivative-based controllers can be implemented for *arbitrary* topologies simply by using inhomogeneous multiple-delay controllers.

We also note that multi-lead-compensator architecture can achieve stabilization and pole placement of a double integrator network with arbitrary topology [27].

4 Stabilization Under Constraint and Delay

Limitations, e.g., measurement delays and input saturation, are common in decentralized systems. These limitations pose further difficulty for stabilization. In the centralized setting, low-gain techniques have been long used for designing stabilizing controllers under saturation constraints and intrinsic delays [28, 29]. In the decentralized setting, Stoorvogel and coworkers recently obtained a check for the existence of a stabilizing control under actuator saturation [22], for plants without defective jw-axis eigenvalues. However, no effort has been devoted to designing stabilizing controllers for decentralized systems with these limitations. In this section, we will explore how to stabilize a double integrator network with measurement delay and/or input saturation using a multiple-delay controller. The scaling property associated with pure integrators facilitates the analysis, and so allows us to highlight the concepts underlying low-gain delay for decentralized systems with little technical complexity; we shall address the low-gain design more generally in future work.

It is worth stressing that the low-gain (scaling) arguments that we use here could have alternatively been used to design multiple-derivative controllers that operate in the presence of saturation and delay. However, it is critical that our implementation of the controllers—which may involve using certain high-gains, e.g. in approximating derivatives as delay differences—operate in the presence of derivatives and delays, and hence we find it more instructive to work directly with the multiple-delay controllers. For ease of presentation, we will show how the basic multiple-delay controller introduced in Section 3 can be modified to permit control under delay and actuator saturation, although similarly other multiple-delay-based or lead-compensation implementations can be adapted.

4.1 Design for Networks with Measurement Delay

Intrinsic delays in measurement pose one of the primary challenges in achieving control of network dynamics. It is well known that measurement delay can cause poor performance and even instability in linear control systems. However, as delays are inherent to our controller implementation, it is possible for us to account for this measurement delay,

Specifically, consider an LTI system consisting of n double integrators with measurement delay, i.e. described by

$$\begin{aligned}\ddot{\mathbf{x}}(t) &= \mathbf{u}(t) \\ y_i(t) &= G_i\mathbf{x}(t-\tau_i),\end{aligned} \tag{4}$$

where $\mathbf{x}(t)$ represents the positions of the n agents and $\mathbf{u}(t)$ the inputs, $y_i(t)$ is the observation made by agent i, $G_i = \begin{bmatrix} G_{i1} & \dots & G_{in} \end{bmatrix}$, and $\tau_i \geq 0$ is the (known) time delay in the measurement made by agent i. Since the delays are assumed known, we can further delay the observations in appropriate channels for the purpose of feedback, so WLOG let us henceforth assume a common delay τ which is the maximum of $\tau_1, \dots, \tau_n$ in each channel.

In order to stabilize (4), let us use a modification of the delay-based decentralized controller given in Equation 3:

$$\begin{aligned}
\mathbf{u}(t) = {} & \frac{k_3}{\rho^2\Delta}\left[k_1\left(\bar{\tau}_2\bar{\tau}_3^2 - \bar{\tau}_2^2\bar{\tau}_3\right) - k_2\frac{\bar{\tau}_2^2 - \bar{\tau}_3^2}{\epsilon} + 2\frac{\bar{\tau}_3 - \bar{\tau}_2}{\epsilon^2}\right]\mathbf{y}(t) + \\
& \frac{k_3}{\rho^2\Delta}\left[k_1\left(\bar{\tau}_3\bar{\tau}_1^2 - \bar{\tau}_3^2\bar{\tau}_1\right) - k_2\frac{\bar{\tau}_3^2 - \bar{\tau}_1^2}{\epsilon} + 2\frac{\bar{\tau}_1 - \bar{\tau}_3}{\epsilon^2}\right]\mathbf{y}(t - \rho\epsilon\bar{\tau}_2 + \tau) + \\
& \frac{k_3}{\rho^2\Delta}\left[k_1\left(\bar{\tau}_1\bar{\tau}_2^2 - \bar{\tau}_2\bar{\tau}_1^2\right) - k_2\frac{\bar{\tau}_1^2 - \bar{\tau}_2^2}{\epsilon} + 2\frac{\bar{\tau}_2 - \bar{\tau}_1}{\epsilon^2}\right]\mathbf{y}(t - \rho\epsilon\bar{\tau}_3 + \tau),
\end{aligned} \tag{5}$$

where we choose ϵ, $\bar{\tau}_i > 0$ and k_i , $i = 1, 2, 3$ in the same way as for the basic multiple-delay controller, and then choose ρ such that $\rho = \frac{\tau}{\epsilon\bar{\tau}_1}$. We formalize the design of the control law for a double integrator network with measurement delay in Theorem 4:

Theorem 4 *Consider the double-integrator network with measurement delay described in (4), where G is nonsingular, and assume that the corresponding undelayed double-integrator can be stabilized using the basic multiple-delay controller (3). The network can be stabilized using the delay-based control law (Equation 5) by choosing sufficiently small ϵ, $\rho = \frac{\tau}{\epsilon\bar{\tau}_1}$, and k_1, k_2 and k_3 satisfying the conditions given in Theorem 1.*

Proof: It is easy to check that the closed-loop system is

$$\begin{aligned}
\ddot{\mathbf{x}}(t) = {} & \frac{k_3}{\rho^2\Delta}\left[k_1\left(\bar{\tau}_2\bar{\tau}_3^2 - \bar{\tau}_2^2\bar{\tau}_3\right) - k_2\frac{\bar{\tau}_2^2 - \bar{\tau}_3^2}{\epsilon} + 2\frac{\bar{\tau}_3 - \bar{\tau}_2}{\epsilon^2}\right]G\mathbf{x}(t - \rho\epsilon\bar{\tau}_1) \\
& + \frac{k_3}{\rho^2\Delta}\left[k_1\left(\bar{\tau}_3\bar{\tau}_1^2 - \bar{\tau}_3^2\bar{\tau}_1\right) - k_2\frac{\bar{\tau}_3^2 - \bar{\tau}_1^2}{\epsilon} + 2\frac{\bar{\tau}_1 - \bar{\tau}_3}{\epsilon^2}\right]G\mathbf{x}(t - \rho\epsilon\bar{\tau}_2) \\
& + \frac{k_3}{\rho^2\Delta}\left[k_1\left(\bar{\tau}_1\bar{\tau}_2^2 - \bar{\tau}_2\bar{\tau}_1^2\right) - k_2\frac{\bar{\tau}_1^2 - \bar{\tau}_2^2}{\epsilon} + 2\frac{\bar{\tau}_2 - \bar{\tau}_1}{\epsilon^2}\right]G\mathbf{x}(t - \rho\epsilon\bar{\tau}_3).
\end{aligned} \tag{6}$$

According to the scaling property presented in [19], a scaling of each delay term in the control law by a factor of ρ together with a scaling of $\frac{1}{\rho}$ in each corresponding gain does not change the stability of the system. Hence the closed-loop system (6) is stable if and only if the system (1) using control law (3) is stable. Thus, we can design ϵ, $0 < \bar{\tau}_1 < \bar{\tau}_2 < \bar{\tau}_3$ and k_i, $i = 1, 2, 3$ according to Theorem 1, and choose $\rho = \frac{\tau}{\epsilon\bar{\tau}_1}$ to achieve stabilization under delay. □

This theorem allows us to design the control law to stabilize a decentralized network with measurement delay. Essentially, by using the low-gain technique [19] and choosing the scaling factor ρ to match $\rho\epsilon\bar{\tau}_1$ with the measurement delay τ, we can absorb the measurement delay into the delay in the control law, and hence transform the closed-loop dynamics of the system with measurement delay into exactly the same form discussed in Section 2. A design that achieves stability can thus be implemented. The performance of the controller can be optimized by choosing $\bar{\tau}_i$ and k_i, $i = 1, 2, 3$ such that ρ is minimized.

Remark: The above result applies to systems with known measurement delay. However, the design can straightforwardly be adapted to plants with unknown but upper-bounded delays, which are also of common interest [18]. In particular, noting that the multiple-derivative controller achieves stability for all sufficiently large k_3, we see that the plant can be stabilized using the controller (5) with $\rho \geq \frac{\tau_{max}}{\epsilon\tau_1}$, where τ_{max} is the upper bound on the delay. We notice that this generalization requires that the open-loop $j\omega$-axis eigenvalues of the plant are in fact at the origin.

4.2 Controller Design for Networks with Input Saturation

In general, low gain techniques can help to resolve instability caused by input saturation [19]. In our setting, as we discussed in Section 4.1, we can simultaneously decrease the input gain and introduce more delay to the decentralized control law. With such scaling, we can guarantee that actuators do not saturate while leaving the closed-loop (linear) system's poles in the OLHP, and hence ensure stability. For the following decentralized system with input saturation:

$$\begin{aligned} \ddot{\mathbf{x}}(t) &= \sigma(\mathbf{u}(t)) \\ \mathbf{y}(t) &= G\mathbf{x}(t), \end{aligned} \tag{7}$$

we use the following control law:

$$\begin{aligned} \mathbf{u}(t) = \frac{k_3}{\rho^2\Delta}&\left[k_1\left(\bar{\tau}_2\bar{\tau}_3^2 - \bar{\tau}_2^2\bar{\tau}_3\right) - k_2\frac{\bar{\tau}_2^2 - \bar{\tau}_3^2}{\epsilon} + 2\frac{\bar{\tau}_3 - \bar{\tau}_2}{\epsilon^2}\right]\mathbf{y}(t - \rho\epsilon\bar{\tau}_1) \\ +\frac{k_3}{\rho^2\Delta}&\left[k_1\left(\bar{\tau}_3\bar{\tau}_1^2 - \bar{\tau}_3^2\bar{\tau}_1\right) - k_2\frac{\bar{\tau}_3^2 - \bar{\tau}_1^2}{\epsilon} + 2\frac{\bar{\tau}_1 - \bar{\tau}_3}{\epsilon^2}\right]\mathbf{y}(t - \rho\epsilon\bar{\tau}_2) \\ +\frac{k_3}{\rho^2\Delta}&\left[k_1\left(\bar{\tau}_1\bar{\tau}_2^2 - \bar{\tau}_2\bar{\tau}_1^2\right) - k_2\frac{\bar{\tau}_1^2 - \bar{\tau}_2^2}{\epsilon} + 2\frac{\bar{\tau}_2 - \bar{\tau}_1}{\epsilon^2}\right]\mathbf{y}(t - \rho\epsilon\bar{\tau}_3), \end{aligned} \tag{8}$$

where we choose ϵ, $\bar{\tau}_i$ and k_i , $i = 1, 2, 3$ in the same way as for the basic multiple-delay controller (i.e., assuming no saturation), and then choose ρ so that the actuator does not saturate. We formalize the design of the control law for a double integrator network with input saturation in Theorem 5:

Theorem 5 *Consider the double-integrator network with input saturation described in (7), where G is nonsingular. Assume that the corresponding (unsaturated) double-integrator network can be stabilized using the basic multiple-delay controller (3). The network can be semiglobally*[3] *stabilized using the multiple-delay control law (Equation 8). Specifically, for any bounded set of initial conditions $\mathcal{W}$ and for fixed k_1, k_2 and k_3 satisfying the conditions*

[3]See [30] for a full introduction to semiglobal stabilization.

in Theorem 1 and sufficiently small ϵ, we can choose ρ sufficiently large such that $\mathcal{W}$ is in the domain of attraction.

Proof: According to the scaling property presented in [19], as long as the actuators never saturate, the response $x(t)$ for the scaled system is exactly a time-scaled version of the response to the unscaled system (Equation 3). We shall use this fact to prove stability. First, from stability of the original unscaled system, we obtain that, for the given compact set of initial conditions, there is a bound on the absolute value of all state variables $x(t)$ over all $t \geq 0$. Thus, there is also a bound, say Γ, on the inputs over all $t \geq 0$. By choosing $\rho < \sqrt{\frac{1}{\Gamma}}$, invoking the time-scaling of the state, and noting that thus the input is scaled by a factor of Γ, we obtain that the inputs do not saturate. We thus automatically recover semi-global stability. □

The low gain technique used in Theorem 5 achieves stabilization under saturation, at a cost of scaling the closed-loop poles a factor of $\frac{1}{\rho}$. We note that the delay-based design in principle allows us to tune ρ so that the state exactly reaches the saturation boundary but does not exceed it, given the set of possible initial conditions. Thus, we can indeed achieve a good low-gain design with the methodology.

Acknowledgements

This work was partially supported by the National Science Foundation under Grant ECS-0528882 and by the National Aeronautics and Space Administration under Grant NNA06CN26A.

References

[1] S. Wang and E. J. Davison, "On the stabilization of decentralized control system," *IEEE Transactions on Automatic Control*, vol. AC-18, pp. 473-478, October 1973.

[2] J. P. Corfmat and A. S. Morse, "Decentralized control of linear multivariable systems," *Automatica*, vol. 12, pp. 479-495, 1976.

[3] D. Siljak, *Decentralized Control of Complex Systems,* Academic Press: Boston, 1994.

[4] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, "Future directions in control in an information-rich world," *IEEE Control Systems Magazine*, pp. 20-33, April 2003.

[5] J. Baillieul and P. J. Antsaklis, "Control and communication challenges in networked real-time systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9-28, Jan. 2007.

[6] S. Roy, A. Saberi, and A. Stoorvogel, "Toward a control theory for networks," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10-11, p. 897, July 2007.

[7] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multi-vehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71-82, 2007.

[8] S. Roy, A. Saberi, and K. Herlugson, "Formation and alignment of distributed sensing agents with double-integrator dynamics," IEEE Press Monograph on *Sensor Network Operations*, Jan. 2006.

[9] Y. Wan, S. Roy, and A. Saberi, "Designing spatially-heterogeneous strategies for control of virus spread," *IET Systems Biology*, vol. 2, No. 4, pp. 184-201, 2008. Short version in the *Proceedings of the 2007 Conference on Decision and Control*, New Orleans, LA, December 2007.

[10] S. Roy and A. Saberi,"Scaling: a Canonical design problem for networks," *International Journal of Control*, vol. 80, no. 8, pp. 1342-1353, August 2007.

[11] Y. Wan, S. Roy, A. Saberi, and A. Stoorvogel, "A multiple-derivative and multiple-delay paradigm for decentralized controller design: uniform rank systems," submitted to *Automatica*.

[12] G. Lafferriere, A. Williams, J. Caughman, and J. J. P. Veerman, "Decentralized control of vehicle formations," *Systems and Control Letters*, vol. 54, no. 9, pp. 899-910, 2004.

[13] Y. Wan and S. Roy, "A scalable methodology for evaluating and designing air traffic flow management strategies under uncertainty," in *Proceedings of the 2007 AIAA GNC Conference and Exhibit*, Hilton Head SC, Aug. 2007, also to appear in the *IEEE Transactions on Intelligent Transportation Systems*.

[14] C. L. DeMarco, "Control structures for competitive market-driven power systems," *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando FL, Dec. 2001.

[15] B. D. O. Anderson and J. B. Moore, "Time-varying feedback laws for decentralized control," *IEEE Transactions on Automatic Control*, Vol. 26, No. 5, pp. 1133-1139, October 1981.

[16] J. P. Corfmat and A. S. Morse, "Stabilization with decentralized feedback control," *IEEE Transactions on Automatic Control*, Vol. 21, No. 1, Feb. 1976. pp. 679-682.

[17] S. Roy, Y. Wan, A. Saberi, and A. Stoorvogel, "Multiple-delay approximation of multiple-derivative controllers: pole placement, neutral-type dynamics, and decentralized design applications," in preparation.

[18] J. K. Hale and S. M. V. Lunel, *Introduction to Functional Difference Equations*, Epring-Verlag, 1993.

[19] S.-I. Niculescu and W. Michiels, "Stabilizing a chain of integrators using multiple delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 802-807, May 2004.

[20] S. Roy, A. Saberi and Y. Wan, "On multiple-delay static output feedback stabilization of LTI plants," submitted to the *International Journal of Control*.

[21] A. Saberi, A. Stoorvogel, and P. Sannuti, *Output Regulation and Control Problems with Regulation Constraints*, Springer-Verlag, 1999.

[22] A. Stoorvogel, A. Saberi, C. Deliu, and P. Sannuti, "Decentralized stabilization of time-invariant systems subject to actuator saturation," *Advanced Strategies in Control Systems with Input and Output Constraints*, LNCIS series (S. Tarbouriech et al eds.), Springer-Verlag, June 2006.

[23] P. Sannuti and A. Saberi, "A special coordinate basis of multivariable linear systems, finite and infinite zero structure, squaring down, and decoupling," *International Journal of Control*, vol. 45, no. 5, pp. 1655-1704, May 1987.

[24] A. Saberi, B. M. Chen, and P. Sannuti, *Loop Transfer Recovery: Analysis and Design*, Springer-Verlag, 1993.

[25] E. Kreindler and A. Jameson, "Optimality of linear control systems," IEEE Transactions on Automatic Control, vol. 17, no. 3, pp. 349-351, June 1972.

[26] W. Michiels and T. Vyhlidal, "An eigenvalue based approach for stabilization of linear time-delay systems of neutral type," *Automatica*, vol. 41, pp. 991-998, 2005.

[27] Y. Wan, S. Roy, A. Saberi, and A. Stoorvogel, "Design of multi-lead-compensators for stabilization and pole placement of double-integrator networks," in preparation.

[28] Z. Lin and H. Fang, "On asymptotic stabilizability of linear systems with delayed input," *IEEE Transaction n Automatic Control*, vol. 52, no. 6, June 2007.

[29] Z. Lin and A. Saberi, "Semi-global exponential stabilization on linear systems subject to "input saturation" via linear feedbacks," *Systems and Control Letters*, vol. 21, pp. 225-239, 1993.

[30] A. Saberi, A.Stoorvogel, and P. Sannuti, *Output Regulation and Control Problems with Regulation Constraints*, Springer-Verlag: New York, 1999.