Decentralized Regulation of Nonlinear Discrete-Time Multi-Agent Systems

by

Nasim Alsadat Shams

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis focuses on decentralized deadbeat output regulation of discrete-time nonlinear plants that are composed of multiple agents. These agents interact, via scalar-valued signals, in a known structured way represented with a graph. This work is motivated by applications where it is infeasible and/or undesirable to introduce control action within each plant agent; instead, control agents are introduced to interact with certain plant agents, where each control agent focuses on regulating a specific plant agent, called its target. Then, two analyses are carried out to determine if regulation is achieved: targeting analysis is used to determine if control laws can be found to regulate all target agents, then growing analysis is used to determine the effect of those control laws on non-target plant agents. The strength of this novel approach is the intuitively-appealing notion of each control agent focusing on the regulation of just one plant agent. This work goes beyond previous research by generalizing the class of allowable plant dynamics, considering not only arbitrary propagation times through plant agents, but also allowing for non-symmetrical influence between the agents. Moreover, new necessary and sufficient algebraic conditions are derived to determine when targeting succeeds. The main contribution of this work, however, is the development of new easily-verifiable conditions necessary for targeting and/or growing to succeed. These new conditions are valuable due to their simplicity and scalability to large systems. They concern the positioning of control agents and targets as well as the propagation time of signals through the plant, and they help significantly with design decisions. Various graph structures (such as queues, grids, spiders, rings, etc.) are considered and for each, these conditions are used to develop a control scheme with the minimum number of control agents needed.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Daniel E. Davison for his guidance, support and constructive feedback throughout the process of completing my M.A.Sc degree and this thesis. I would also like to thank Professors Nielsen and Sundaram for their comments on this thesis.

Finally, I would like to thank my family, Neda, Nadia and Jahangir for their support and belief in me. They have always been a constant source of love, concern, support and strength to me. I greatly appreciate all that they have done for me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis deals with the decentralized output regulation of nonlinear discrete-time multi-agent systems. By "multi-agent" systems we mean that the plant is composed of subsystems, called agents, that interact with one another. By "output regulation" we mean that it is desirable to drive to zero the output signals of all plant agents; in particular, we focus on deadbeat regulation, whereby the output signals are driven to zero in finite time. As the first step to control a multi-agent plant, we introduce *control agents*, placed at strategic locations among the plant agents, with each control agent focusing on regulating a specific plant agent, called its *target*. A novel analysis approach is then used to determine if control laws can be found for the control agents so that the plant is successfully regulated. The analysis has two stages, namely *targeting analysis* and *growing analysis*. Through targeting analysis, it is determined if the control agents are capable of driving the outputs of all target plant agents to zero. Then, growing analysis is applied to determine if the same control laws can drive the outputs of non-target plant agents to zero.

The targeting approach used in this thesis has the advantage that it is intuitively easy to understand and, we argue, often desirable in practical problems. As an example, a teacher who tries to keep a class of noisy students in control might naturally focus on controlling a few troublemaker students in the class rather than try to control every single student simultaneously, which could be overwhelming. Another key characteristic of our approach to regulation is that it relies on using separate control agents to bring about control action. This method is beneficial for any multi-agent situation where it is infeasible and/or undesirable to place a controller within all plant agents. Many social and economic systems fall in this category. For instance, police officers trying to control an unruly crowd at a protest, a central bank trying to influence a country's economy, or government leaders trying to influence a country's population or culture towards a certain issue can all benefit from this approach. In addition to socio-economic systems, this technique can also be potentially beneficial in some industry and military applications. A few such examples are listed in the following:

- Unmanned autonomous vehicles (UAVs): A targeting approach can potentially be used to control a number of unmanned autonomous vehicles by influencing only a small number of target UAVs through a few controller UAVs. Using this approach, simpler UAVs can be developed as the plant agents, potentially resulting in a considerable reduction in the production and maintenance costs.

- Distributed energy generation: Controlling a network consisting of thousands of small-scale power generators (such as wind turbines and solar bank cells) is very challenging. The idea of having central power authorities focus on controlling only a few target generators can greatly simplify the problem.

- Traffic control: In order to control traffic congestion, a targeting approach can be used by the controllers to control the traffic signals at target intersections or to control the speed limits at target roads, rather than try to control the traffic signals or speed limits at all the intersections and roads.

- Water management: The idea of targeting can be used to control a few target dams or pumps. This approach can greatly simplify the control of all dams and pumps and potentially result in a more effective water management system.

The idea of targeting was introduced by Spieser and Davison to stabilize crowds modelled by suggestibility theory, as characterized in a well-known book by LeBon (see [5, 20, 23, 22, 21, 3]). This control strategy was first used to stabilize the people in a queue [22]. The idea was then extended to one-dimensional crowds [20] and two-dimensional crowds [23]. Targeting and growing analyses were primitive at this stage and were developed only to handle the specific psychological dynamics of a crowd where the propagation times through plant agents were assumed to be constant (i.e., one time step) and plant agents were assumed to influence each other in a symmetrical manner (i.e., each agent influenced its neighbour as much as it was affected by it). Later in [5], the important notion of "dependency graphs" was introduced to deal with targeting analysis. This thesis extends the previous research by generalizing the class of plant dynamics considered, allowing for arbitrary propagation times through plant agents and control agents, and also allowing for non-symmetrical influence between the agents. Targeting and growing analyses are also generalized. Moreover, new necessary and sufficient algebraic conditions are derived to determine when targeting succeeds. Finally, new easily-verifiable necessary conditions have been developed for targeting and/or growing to succeed [18]. While these new conditions are valuable due to their simplicity and scalability to large systems, they have also provided us with much better understanding of targeting and growing. As a result, it is much easier now to deal with the design problem. In fact, a better understanding of certain structures (such as queues, grids, spiders, rings, etc.) has been established in this thesis.

Apart from the aforementioned previous research, the use of targeting to control a multi-agent system is, to the best of the author's knowledge, completely a novel approach. However, the fact that the control agents are separately introduced to the plant and are distinctly different from the plant agents, puts this work in the classical *decentralized control* framework [2, 4, 25, 19]. Also, this work is related to the modern co-operative control theory [15, 14, 24, 7, 12, 17, 16, 13, 9, 8] due to the importance of the graph structure and that some degree of communication and co-ordination is usually needed among the control agents. Leader-follower problems within the formation control literature (a subfield of co-operative control) are in particular similar to this work in that there is a distinction between the types of the agents (leaders or followers) in the plant similar to what is used in the notion of targeting (control agents or plant agents). Moreover, the graph structure is emphasized in the leader-follower problems ([9] and [16]). Finally, at a higher level, this research can be related to the work in the structured systems literature [6] where great emphasize is placed on the graph structure of the system.

A brief overview of the chapters of this thesis is given in the following. Chapter 2 first introduces a collection of notation and terminology that is used throughout this thesis. Then, it provides the system model by formulating the dynamics of the plant agents and control agents in the system. Finally, various assumptions are made that mainly concern targeting assignment, communication and sensing requirements, signal propagation time in the system, and targeting and growing processes. These assumptions have been introduced either as a necessity to be able to achieve the desired control objective or to simplify the analysis.

In Chapter 3, the control objective is formalized and the two problems related to this objective are stated. Trying to achieve deadbeat output regulation of all the plant agents, the first problem deals with finding, if possible, a set of such control laws for a given plant where the placement of control agents and targeting assignment are known. The second problem is to determine the number of control agents required to regulate a given plant, where they should be placed, and which plant agents they should target.

Chapter 4 addresses the first problem. Targeting analysis and growing analysis are described in detail in this chapter. Targeting analysis is used to determine if the control agents are able to zero[1] all the targets, while growing analysis is used to determine the effect of the control laws on non-target plant agents. To achieve regulation of the plant, both analyses must be carried out successfully. It is shown that two important factors, involved in regulation of any plant, are the graph structure and the signal propagation time

[1]We say that a plant agent is "zeroed" if the control laws drive the plant agent output to zero in finite time.

in the plant. This chapter ends with a section on the communication requirements among the control agents and the sensing requirements for control agents to regulate a plant. It is shown that communication and sensing requirements are not unique for a given control scheme and there is (generally) a trade-off between the two.

In Chapter 5, a set of necessary conditions, as the main contribution of this work, are discussed to ensure targeting analysis and/or growing analysis are successful. These necessary conditions can help a designer determine how many control agents should be used, where control agents should be placed, and what targeting assignment should be used to ensure regulation can be achieved. Implications of these necessary conditions are provided in Chapter 6. The second problem is addressed in this chapter for different graph structures, including queues, grids, spiders, rings, wheels, complete graphs and null graphs. Finally, Chapter 7 summarizes the main results of the thesis and provides a number of interesting and promising directions to further this research.

Chapter 2

System Model

This chapter describes in detail the model of the system considered in our work. First a collection of notation and terminology are introduced to represent a controlled multi-agent plant (i.e., a multi-agent plant with a given control scheme). Next, the dynamics of the plant and control agents are defined. Finally, the required assumptions for the system are explained thoroughly with the use of examples where needed.

2.1 Notation and Terminology

Consider a plant composed of n distinct agents that interact in a known manner. Assume m control agents are used to control the plant. In order to represent the plant and the control agents, a directed graph is used. The vertices of the graph are the plant agents and the control agents, denoted $O_1, ..., O_n$ and $X_{n+1}, ..., X_{n+m}$, respectively. The directed edges of the graph represent the interactions between the agents; specifically, a directed edge from O_i to O_j means that the output of O_i is an input to O_j (similarly for control agents). Also, the target of control agent X_i for $n+1 \leq i \leq n+m$ is denoted T_i where $T_i = O_j$ for some j in the interval $1 \leq j \leq n$.

Define the set of *neighbours* of O_i (for $1 \leq i \leq n$) as

$$\begin{aligned}\mathcal{N}(O_i) &= \{O_j : \text{ there is a directed edge from } O_j \text{ to } O_i \text{ for } 1 \leq j \leq n\} \\ &\cup \{X_j : \text{ there is a directed edge from } X_j \text{ to } O_i \text{ for } n+1 \leq j \leq n+m\}.\end{aligned}$$

In an analogous way, define the set of neighbours of X_i (for $n+1 \leq i \leq n+m$) as

$$\begin{aligned}\mathcal{N}(X_i) &= \{O_j : \text{ there is a directed edge from } O_j \text{ to } X_i \text{ for } 1 \leq j \leq n\} \\ &\cup \{X_j : \text{ there is a directed edge from } X_j \text{ to } X_i \text{ for } n+1 \leq j \leq n+m\}.\end{aligned}$$

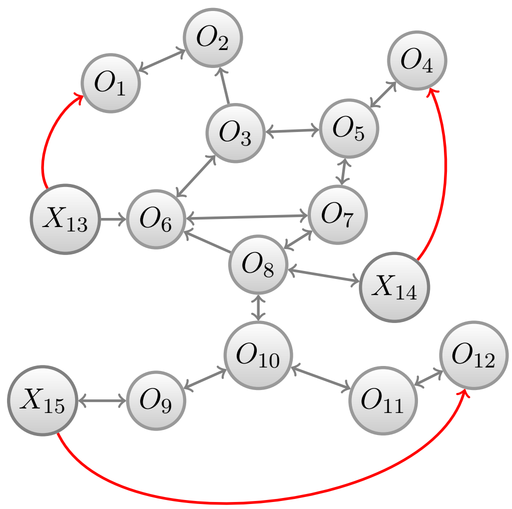


Figure 2.1: A controlled plant composed of 12 plant agents (O_1-O_{12}) and 3 control agents (X_{13}-X_{15}).

Figure 2.1 shows a system composed of 12 plant agents and 3 control agents. Target assignments are shown using red arrows. Thus, $T_{13} = O_1$, $T_{14} = O_4$ and $T_{15} = O_{12}$. As an example, the set of neighbours of O_8, X_{13}, and X_{14} are $\mathcal{N}(O_8) = \{O_7, O_{10}, X_{14}\}$, $\mathcal{N}(X_{13}) = \emptyset$, and $\mathcal{N}(X_{14}) = \{O_8\}$, accordingly.

Adapting terminology from graph theory, a *path* refers to a sequence of nodes that follow directed edges such that no nodes appear in the sequence more than once. For the system in Figure 2.1, there are only two paths from X_{13} to O_1: $X_{13} \to O_6 \to O_3 \to O_2 \to O_1$, and $X_{13} \to O_6 \to O_7 \to O_5 \to O_3 \to O_2 \to O_1$, while there exists no path from any agent to X_{13}.

2.2 The System Model

The dynamics of the plant agents and control agents are formulated in this section. We denote the scalar output signal of O_i (for $1 \leq i \leq n$) at time k by $y_i[k]$. Similarly, we represent the scalar output signal of X_i (for $n+1 \leq i \leq n+m$) by $y_i[k]$. Also, $Y_i[k]$ is defined to be the set of output signals of all neighbours of O_i or X_i; i.e., $Y_i[k] = \{y_j[k] : O_j \in \mathcal{N}(O_i) \text{ or } X_j \in \mathcal{N}(O_i)\}$ or $Y_i[k] = \{y_j[k] : O_j \in \mathcal{N}(X_i) \text{ or } X_j \in \mathcal{N}(X_i)\}$, respectively. For example, for the system in Figure 2.1 where $\mathcal{N}(O_8) = \{O_7, O_{10}, X_{14}\}$, we have $Y_8[k] = \{y_7[k], y_{10}[k], y_{14}[k]\}$.

The dynamics of plant agent O_i (for $1 \leq i \leq n$) are given as below:

$$x_i[k+1] = f_i(x_i[k], Y_i[k]) \quad (2.1)$$
$$y_i[k] = h_i(x_i[k]), \quad (2.2)$$

while those of control agent X_i (for $n+1 \leq i \leq n+m$) are

$$x_i[k+1] = f_i(x_i[k], Y_i[k], u_i[k]) \quad (2.3)$$
$$y_i[k] = h_i(x_i[k], u_i[k]), \quad (2.4)$$

where $x_i[k]$ represents the state vector of O_i (for $1 \leq i \leq n$) or that of X_i (for $n+1 \leq i \leq n+m$), and $u_i[k]$ denotes the scalar control signal of X_i (for $n+1 \leq i \leq n+m$), where $u_i[k]$ possibly is a function of various states and other control signals (see Assumptions A_2 and A_3 below).

2.3 Assumptions

For the system with dynamics (2.1)–(2.4), seven standing assumptions are required, as explained in the following. Assumptions A_1-A_4 are taken almost word for word from [5]. Davison and Spieser introduce two other assumptions in [5] concerning the set of neighbours of each control agent, and the distances[1] between the control agents and the targets, and then try to relax them. In this work, however, these assumptions have been removed. In fact, Chapters 4 and 5 provide a number of theorems for successful targeting and/or growing that concern these issues. The last three assumptions are specific to this work. Assumption A_5 deals with the signal propagation times through the system, and the last two assumptions are critically required for regulation of any plant with generalized dynamics of (2.1)–(2.4). The first four assumptions are as follows:

Assumption A_1: There is at least one path from each control agent to its associated target.

Assumption A_2: Control agents can communicate among themselves, with no time delays.

Assumption A_3: Each control agent can sense the state of every plant agent or control agent.

Assumption A_4: Each control agent targets a specific plant agent, but there are no duplicate targets. Hence, in total there are m distinct targets.

Assumption A_1 is necessary for X_i to be able to successfully zero T_i. Assumption A_2 is included since the control laws developed in this work often require control agents to

[1]The reason for using the notion of *distance* between the control agents and the targets is explained in Assumption A_5.

be able to communicate with other control agents; for simplicity, we assume that all such communications are possible with no delays. Assumption A_3, likewise, is made partly out of necessity and partly out of simplicity. This is because the control laws developed require some state information; also, loosening the assumption complicates several matters. Assumption A_4 is included to simplify analysis and to avoid redundancy in control effort.

The next assumption concerns the signal propagation time around the system. Let $\delta_i \geq 0$ be the *propagation time* through control agent X_i (for $n+1 \leq i \leq n+m$) which is defined as the time (measured in samples) required for a change in $u_i[k]$ to propagate through the dynamics (2.3)–(2.4) and to result in a change in $y_i[k]$. In the special case where the dynamics (2.3)–(2.4) are linear, δ_i is the relative degree of the transfer function from u_i to y_i. Similarly, for any $O_j \in \mathcal{N}(O_i)$ or $X_j \in \mathcal{N}(O_i)$ or $O_j \in \mathcal{N}(X_i)$, define $\delta_{ji} \geq 1$ to be the time required for a change in $y_j[k]$ to propagate through the dynamics (2.1)–(2.2) or (2.3)–(2.4) and to result in a change in $y_i[k]$. Our analysis requires the mild assumption that the propagation times do not depend on the values of the signals:

Assumption A_5: The dynamics (2.3)–(2.4) are such that the propagation time δ_i is independent of $u_i[k]$, and the dynamics (2.1)–(2.2) and (2.3)–(2.4) are such that the propagation time δ_{ji} is independent of $y_j[k]$.

Assumption A_5 implies that the propagation time along any path in the plant is constant. In the previous work done by Spieser and Davison (see [5, 20, 23, 22, 21, 3]), and our earlier work in [18], we use the notion of *distance* between control agents and targets to refer to the signal propagation time in the system. This is a valid assumption for non-directed graphs with equal propagation times through all agents. In this thesis, however, we generalize our work by allowing for arbitrary propagation times through plant agents and/or control agents and also by allowing for non-symmetrical influence between plant agents and/or control agents. Thus, the notion of distance is no more valid for our purpose, and we specifically use propagation times through the plant agents and control agents. In fact, much of the analysis in this work depends on the propagation time along paths that originate at control agents and end at targets, so additional notation is warranted: if there is a path from X_i to O_j (for $n+1 \leq i \leq n+m$ and $1 \leq j \leq n$), define $\Delta(X_i, O_j)$ to be the time required for a change in $u_i[k]$ to propagate through the path, resulting in a change in $y_j[k]$. If there is only one such path, say $X_i \to O_\alpha \to O_\beta \to \ldots \to O_\zeta \to O_j$, then $\Delta(X_i, O_j) = \delta_i + \delta_{i\alpha} + \delta_{\alpha\beta} + \cdots + \delta_{\zeta j}$; if there are multiple paths from X_i to O_j, then $\Delta(X_i, O_j)$ is the smallest such sum. If there are no paths from X_i to O_j, then the convention $\Delta(X_i, O_j) = \infty$ is used. As an example, consider the system in Figure 2.2 with the following dynamics:

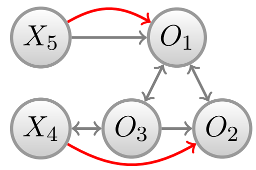


Figure 2.2: A system composed of 3 plant agents and 2 control agents.

$$
\begin{aligned}
x_1[k+1] &= y_5[k] + y_2[k] - y_3[k],\ \ y_1[k] = x_1[k] && (2.5)\\
x_2[k+1] &= -x_2[k] + 2y_1[k] - 3y_3^3[k],\ \ y_2[k] = x_2[k] && (2.6)\\
x_3[k+1] &= x_3[k] + y_1[k] + y_4[k],\ \ y_3[k] = x_3[k] && (2.7)\\
x_4[k+1] &= u_4[k] - y_3[k],\ \ y_4[k] = x_4[k] && (2.8)\\
x_{5,1}[k+1] &= 2u_5[k], && (2.9)\\
x_{5,2}[k+1] &= x_{5,1}[k],\ \ y_5[k] = x_{5,2}[k]. && (2.10)
\end{aligned}
$$

For this example, Assumptions A_1 and A_4 are satisfied by construction. In addition, Assumption A_5 is satisfied, leading to:

$$
\begin{aligned}
\Delta(X_4, O_1) &= \delta_4 + \delta_{43} + \delta_{31} = 1 + 1 + 1 = 3,\\
\Delta(X_4, O_2) &= \delta_4 + \delta_{43} + \delta_{32} = 1 + 1 + 1 = 3,\\
\Delta(X_4, O_3) &= \delta_4 + \delta_{43} = 1 + 1 = 2,\\
\Delta(X_5, O_1) &= \delta_5 + \delta_{51} = 2 + 1 = 3,\\
\Delta(X_5, O_2) &= \delta_5 + \delta_{51} + \delta_{12} = 2 + 1 + 1 = 4,\\
\Delta(X_5, O_3) &= \delta_5 + \delta_{51} + \delta_{13} = 2 + 1 + 1 = 4.
\end{aligned}
$$

The next assumption is that each control agent, considered by itself, should be able to zero its target, while the deeper issue of whether or not control agents are able to simultaneously zero their targets (called *targeting analysis*) is left to Chapter 4:

Assumption A_6: Each control agent (say X_i), considered by itself, is able to zero its target (say $T_i = O_j$) in the sense that, if all control signals other than that of X_i are presumed to be known for all time, then the control signal $u_i[k]$ can always be found, possibly dependent on the state of various agents at time k and on the other presumed-known control signals, to force $y_j[k + \Delta(X_i, O_j)] = 0$. (Note that $\Delta(X_i, O_j) < \infty$, by Assumption A_1.)

Assumption A_6 can be thought of as a controllability-like requirement. It is simply a matter of iterating through the plant dynamics to determine, for each control agent, whether or not Assumption A_6 is satisfied. As an example, consider again the system in Figure 2.2 with dynamics (2.5)–(2.10). Control agent X_4 has target $T_4 = O_2$ with $\Delta(X_4, O_2) = 3$. Direct iteration through the plant dynamics over three samples yields an expression for $y_2[k+3]$ as follows:

$$\begin{aligned} y_2[k+3] &= 4x_1[k]-5x_2[k]-9x_3^3[k]-2x_4[k]-2x_{5,2}[k] \\ &\quad +4u_5[k-1]+3(x_3[k]+x_1[k]+x_4[k])^3 \\ &\quad -3(x_1[k]+x_2[k]-x_3[k]+x_4[k]+x_{5,2}[k]+u_4[k])^3. \end{aligned} \tag{2.11}$$

Setting this expression to zero yields

$$\begin{aligned} u_4[k] &= -x_1[k]-x_2[k]+x_3[k]-x_4[k]-x_{5,2}[k]+\{4x_1[k] \\ &\quad -5x_2[k]-9x_3^3[k]-2x_4[k]-2x_{5,2}[k]+4u_5[k-1] \\ &\quad +3(x_3[k]+x_1[k]+x_4[k])^3\}^{1/3} \times 3^{-1/3}. \end{aligned} \tag{2.12}$$

Thus, it is possible to find $u_4[k]$, dependent on the presumed-known $u_5[k-1]$ and various state components at time k, to force $y_2[k+3] = 0$. This means Assumption A_6 is satisfied for X_4. Similarly, for X_5, we obtain $\Delta(X_5, T_5) = 3$, and the expression for $y_1[k+3]$ is

$$\begin{aligned} y_1[k+3] &= -3x_1[k]+2x_2[k]-x_3[k]+3x_3^3[k] \\ &\quad -x_4[k]+x_{5,2}[k]-u_4[k] \\ &\quad -3(x_3[k]+x_1[k]+x_4[k])^3+2u_5[k]. \end{aligned} \tag{2.13}$$

Hence the control signal that forces $y_1[k+3] = 0$ is

$$\begin{aligned} u_5[k] &= 0.5\{3x_1[k]-2x_2[k]+x_3[k]-3x_3^3[k]+x_4[k] \\ &\quad -x_{5,2}[k]+u_4[k]+3(x_3[k]+x_1[k]+x_4[k])^3\}. \end{aligned} \tag{2.14}$$

From the above equation, it can be seen that $u_5[k]$ depends on the presumed-known $u_4[k]$ and various state components at time k. Hence, Assumption A_6 is satisfied also for X_5.

Finally, Assumption A_7 below is central to the *growing analysis* of Chapter 4. Specifically, it allows for the possibility that, by having control agents zero targets, other non-target plant agents will also be zeroed by the same control laws. The assumption is as follows:

Assumption A_7: The dynamics of O_i in (2.1)–(2.2) satisfy the property that, if all of the signals in Y_i (for $1 \le i \le n$) are fixed at zero, except for one (call that one y_j), then, for any $\bar{k} \ge 0$, $y_i[k] = 0$ (for $k \ge \bar{k}$) implies $y_j[k] = 0$ (for $k \ge \bar{k}$).

The example in Figure 2.2 with dynamics (2.5)–(2.10) satisfies this assumption: for O_1, if y_1 is held at zero then, since y_1 and x_1 are equal, x_1 must also be held at zero, which implies $y_5 + y_2 - y_3$ is held at zero, and therefore if any two of y_5, y_2, and y_3 are held at zero, the remaining one is also necessarily zero; similarly, for O_2, if y_2 is held at zero then x_2 must also be held at zero, which implies $2y_1 - 3y_3^3$ is held at zero, and therefore one of y_1 or y_3 must be held at zero if the other one is held at zero; likewise, for O_3, if y_3 is held at zero then x_3 must be held at zero, which implies $y_1 + y_4$ is held at zero, and so if one of y_1 or y_4 is held at zero, the other must be held at zero too.

Having stated all the required assumptions, we formalize the control objective and the two problems we deal with in this thesis in the following chapter.

Chapter 3

Problem Formulation

The control objective in this thesis is deadbeat regulation of the outputs of all plant agents; That is, it is desirable to drive the output signals of all plant agents to zero in finite time. We define *settling time* as below:

Definition 1. *Settling time, denoted λ, is the time it takes for the control agents to achieve deadbeat output regulation of the plant agents from any initial condition (IC), i.e.,*

$$\lambda = \max_{\text{IC}} \min\{\overline{\lambda} : y_i[k] = 0 \ \forall k \geq \overline{\lambda}, 1 \leq i \leq n\}.$$

If full-state regulation is desired, so that the state vectors x_i (for $1 \leq i \leq n$) are driven to zero instead of just the outputs y_i, then additional assumptions on the plant agent dynamics in (2.1)–(2.2) are needed. For example, deadbeat output regulation implies full-state regulation if the systems $x_i[k+1] = f_i(x_i[k], 0)$ (for $1 \leq i \leq n$) are assumed to be asymptotically stable. A more complete examination of full-state regulation is left as a topic for future work.

Different questions might need to be answered when one is trying to regulate a plant or trying to verify whether or not regulation is achievable for a plant. As an example, consider the controlled plant in Figure 3.1 with dynamics (2.5)–(2.10) which was first introduced in the previous chapter. The first basic question is whether or not the control agents —in the given setting— can achieve deadbeat output regulation of the plant. Other questions that might arise include: Is it possible to achieve the control objective with a fewer number of control agents? Where should we place the control agents? How should we assign targets to the control agents? What is the optimal settling time? How will the control scheme be affected when disturbances are introduced to the plant?

For the purpose of this thesis, we consider two problems that are stated in the following. It should be noted that, throughout this thesis, the term "regulation" refers to "deadbeat output regulation."

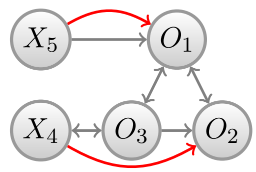


Figure 3.1: A system composed of 3 plant agents and 2 control agents.

Problem 1: For a given plant, set of control agents with a specific placement, and targeting assignment, find, if possible, a set of m computable control laws that regulate the plant.

Problem 2: For a given plant, determine how many control agents are needed, where they should be placed, and how targets should be assigned so that regulation is achievable using computable control laws.

We use the term *computable control laws* in the following sense:

Definition 2. *The control signals $u_{n+1}[\cdot], \ldots, u_{n+m}[\cdot]$ are* computable *if, for every time k, there exists a permutation of $(u_{n+1}[k], \ldots, u_{n+m}[k])$ such that each entry (say, $u_p[k]$) in the permutation can be determined using only*

- *the values of the entries to the left of $u_p[k]$,*
- *control signal data from time $k-1$ or earlier, and/or*
- *state data from time k or earlier.*

In other words, computability requires that the control laws be causal and that control signals at any time k can be evaluated *sequentially*. The need for this notion of computability arises since, if two or more control signals need to be solved concurrently, then (in general) a set of algebraic nonlinear equations must be solved; such equations may not even have a solution.

To make the notion of computability more clear, consider again the system in Figure 3.1 with dynamics (2.5)–(2.10). In the previous chapter we found the control signals $u_4[k]$ and $u_5[k]$ by forcing $y_1[k+3] = 0$ and $y_2[k+3] = 0$ to hold for any time $k \geq 0$ (note that the settling time is $\lambda = \max\{3, 3\} = 3$). Recall that $u_4[k]$ and $u_5[k]$ are:

$$\begin{aligned} u_4[k] \;&=\; -x_1[k]-x_2[k]+x_3[k]-x_4[k]-x_{5,2}[k]+\{4x_1[k] \\ &\quad -5x_2[k]-9x_3^3[k]-2x_4[k]-2x_{5,2}[k]+4u_5[k-1] \\ &\quad +3(x_3[k]+x_1[k]+x_4[k])^3\}^{1/3} \times 3^{-1/3}, \end{aligned} \tag{3.1}$$

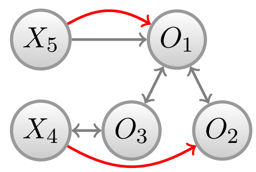


Figure 3.2: The system of Figure 3.1 with a small modification (i.e., no interaction between O_2 and O_3).

$$u_5[k] = 0.5\{3x_1[k] - 2x_2[k] + x_3[k] - 3x_3^3[k] + x_4[k] - x_{5,2}[k] + u_4[k] + 3(x_3[k] + x_1[k] + x_4[k])^3\}. \quad (3.2)$$

It can be seen from (3.1) and (3.2) that $u_4[k]$ depends on $u_5[k-1]$ and some state components at time k, while $u_5[k]$ depends on $u_4[k]$ and various state components at time k. Thus, $u_4[k]$ and $u_5[k]$ in (3.1) and (3.2) are computable in the sense of Definition 2, with the permutation $(u_4[k], u_5[k])$.

Now, suppose the link from O_3 to O_2 in Figure 3.1 is broken completely, resulting in Figure 3.2 with the following simplified dynamics for O_2:

$$x_2[k+1] = -x_2[k] + 2y_1[k], \; y_2[k] = x_2[k]. \quad (3.3)$$

Dynamics (2.5), (2.7)–(2.10), and (3.3) result in $\Delta(X_4, O_2) = 4$ and $\Delta(X_5, O_1) = 3$, and the control laws that force $y_2[k+4] = 0$ and $y_1[k+3] = 0$ are found to be:

$$u_4[k] = -5x_1[k] + 4.5x_2[k] - x_3[k] + 2x_{5,2}[k] + 2u_5[k] - 2u_5[k-1], \quad (3.4)$$

$$u_5[k] = 0.5\{3x_1[k] - 2x_2[k] + x_3[k] + x_4[k] - x_{5,2}[k] + u_4[k]\}. \quad (3.5)$$

From (3.4) and (3.5), $u_4[k]$ depends on $u_5[k]$, $u_5[k-1]$, and various state components at time k, while $u_5[k]$ depends on $u_4[k]$ and a number of state components at time k. Thus, control laws $u_4[k]$ and $u_5[k]$ cannot be computed sequentially; i.e., $u_4[k]$ and $u_5[k]$ in (3.4) and (3.5) are not computable in the sense of Definition 2.

We deal with the first problem in Chapter 4. To address the second problem, we develop necessary conditions for targeting and/or growing to succeed in Chapter 5. Then, in Chapter 6 we consider different graph structures such as queues, grids, spiders, etc., and investigate the design problem for each.

Chapter 4

A Decentralized Control Approach to Regulation

This chapter deals with Problem 1. Under Assumptions A_1–A_7 with known plant and control agent dynamics, known control agent placement, and known targeting assignment, a novel approach is used to determine if regulation is achievable. This approach is carried out in two steps: targeting analysis and growing analysis. Targeting analysis answers the question as to whether or not there exist computable control laws that zero all m targets. If such control laws exist, we say that targeting *succeeds* or *works*; otherwise, we say targeting *fails* or *does not work*. Growing analysis determines the behaviour of non-target plant agents when the same computable control laws are used. If all non-target plant agents are zeroed, then we say that growing *succeeds* or *works*. If not, we say growing *fails* or *does not work*. Regulation is achieved if targeting analysis and growing analysis both succeed. Finally, we close this chapter with a section on the communication requirements among the control agents and the sensing requirements for control agents to regulate a plant.

4.1 Targeting Analysis

If only one control agent is introduced to the plant, then under the given set of assumptions, targeting succeeds. In particular, Assumptions A_1 and A_6 directly imply that control signal $u_{n+1}[k]$ can be found to zero T_{n+1}.

If two or more control agents are placed among the plant agents (i.e., $m \geq 2$), Assumptions A_1 and A_6 imply that each control agent, by itself, is able to zero its target. However, it does not follow that the resulting control laws $u_{n+1}[k], \ldots, u_{n+m}[k]$ are computable, in the sense described in Definition 2. That is, the m control laws, when considered together, may not be causal or may not be able to be evaluated sequentially. The purpose of targeting analysis is to make this determination.

In the previous chapter, we considered a system consisting of three plant agents and two control agents with dynamics (2.5)–(2.10) in Figure 3.1. Iterating through the plant dynamics, we found the expressions (3.1) and (3.2) for the control laws, and discussed in detail in what sense these control laws follow the notion of computability. Also, a variation of this system with dynamics (2.5), (2.7)–(2.10), and (3.3) was considered in Figure 3.2 in which a link between two of the plant agents was removed. By deriving the expressions (3.4) and (3.5) for the control laws, it was shown that regulation of this plant is not achievable; i.e., there exist no computable control laws for regulation of this plant. Hence, it is easy to see that the determination as to whether or not targeting succeeds depends strongly on the structure of the underlying graph.

The other important factor that affects targeting is the propagation time through the plant agents and control agents. To illustrate this point, consider again another variation of the controlled plant in Figure 3.1. Suppose the dynamics of O_2 are modified so that an extra sample is required to process y_3:

$$x_{2,1}[k+1] \;=\; y_3[k] \tag{4.1}$$
$$x_{2,2}[k+1] \;=\; -x_{2,2}[k]+2y_1[k]-3x_{2,1}^3[k], y_2[k]=x_{2,2}[k]. \tag{4.2}$$

Repeating the earlier calculations with these new dynamics results in $\Delta(X_4, T_4) = 4$ and $\Delta(X_5, T_5) = 3$, and targeting analysis concludes that the control law, $u_4[k]$, that forces $y_2[k + 4] = 0$ depends on $u_5[k]$, while the control law, $u_5[k]$, that forces $y_1[k + 3] = 0$ depends on $u_4[k]$. Therefore $u_4[k]$ and $u_5[k]$ depend on one another, and the control laws are not computable; i.e., targeting fails.

The above examples are relatively simple to analyze. For more complicated systems, the use of a *dependency graph* to keep track of the dependencies among $u_{n+1}[\cdot], \ldots, u_{n+m}[\cdot]$ helps greatly. The notion of dependency graphs was first introduced by Davison and Spieser in [5]. The dependency graph has as nodes the signals $u_{n+1}[\cdot], \ldots, u_{n+m}[\cdot]$ at times $\ldots, k-2, k-1, k, k+1, k+2, \ldots$, with directed edges from $u_i[l-1]$ to $u_i[l]$ (for $-\infty < l < \infty$ and for each i in the interval $n+1 \leq i \leq n+m$) to indicate that $u_i[l-1]$ must be computed before $u_i[l]$ can be computed. In addition, directed edges are drawn, for any $j \neq i$ where $\Delta(X_j, T_i) < \infty$, from $u_j[l + \Delta(X_i, T_i) - \Delta(X_j, T_i)]$ to $u_i[l]$ (for $-\infty < l < \infty$) to capture the dependence of $u_i[\cdot]$ on $u_j[\cdot]$. This dependency arises from the fact that after $\Delta(X_i, T_i)$ time steps, the control signals $u_i[l]$ and $u_j[l + \Delta(X_i, T_i) - \Delta(X_j, T_i)]$ both reach the target agent T_i (for $-\infty < l < \infty$); consequently, $u_j[l + \Delta(X_i, T_i) - \Delta(X_j, T_i)]$ must be known to be able to compute $u_i[l]$.

As an example, once again we consider the controlled plant composed of three plant agents and two control agents with dynamics (2.5)–(2.10) and construct its dependency

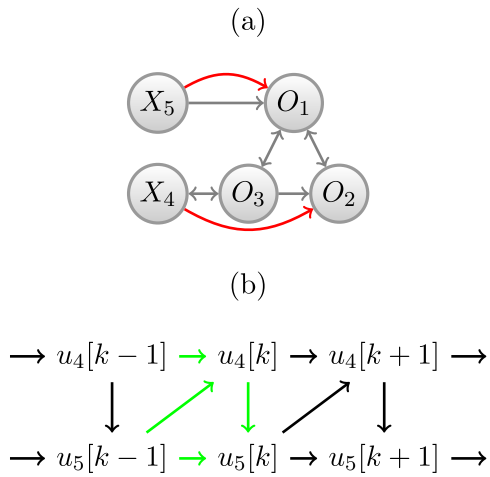


Figure 4.1: A system composed of 3 plant agents and 2 control agents with dynamics (2.5)–(2.10): (a) the graph structure, (b) the corresponding dependency graph.

graph (see Figure 4.1). Having $\Delta(X_4, T_4) = 3$, $\Delta(X_4, T_5) = 3$, $\Delta(X_5, T_4) = 4$, and $\Delta(X_5, T_5) = 3$, we need to draw arrows from $u_4[l]$ to $u_5[l]$, and from $u_5[l-1]$ to $u_4[l]$ (for $-\infty < l < \infty$) in the dependency graph. The graph structure and the dependency graph for this example are shown in Figure 4.1. This dependency graph has no loops, which is consistent with our earlier conclusion: it is possible to compute the control signals sequentially. The sequence in which the control signals should be evaluated can be read off the green arrows in the dependency graph: $u_4[k]$ can be computed using control-signal information from time $k-1$ (in addition, of course, to state information at time k), while $u_5[k]$ can be computed using the value of $u_4[k]$ in addition to control-signal information from time $k-1$ (and, again, state information at time k).

Similarly, we can draw the dependency graph for the two variations of the above example; i.e., (i) the controlled plant with dynamics (2.5),(2.7)–(2.10) and (3.3) as shown in Figure 4.2(a) for ease of reference, and (ii) the controlled plant with dynamics (2.5), (2.7)–(2.10) and (4.1)–(4.2) as shown in Figure 4.2(b). Dynamics of these controlled plants both generate the same dependency graph shown in Figure 4.2(c). Note that there are loops in this dependency graph, capturing the fact that $u_4[k]$ cannot be computed without knowing $u_5[k]$, and vice-versa; consequently, there do not exist computable control laws to zero both targets.

As a more complicated example, consider the system in Figure 4.3(a). Assuming for simplicity that all δ_i and δ_{ji} values are 1, the dependency graph in Figure 4.3(b) results. As it can be seen in this figure, the dependency graph has no loops, so targeting succeeds; i.e., there exists a set of computable control laws that can zero all targets in Figure 4.3(a). The control law sequence can be observed from the dependency graph: at time k, first X_{13}

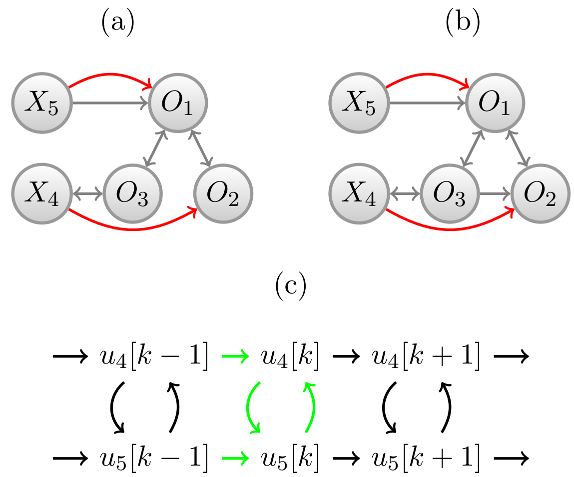


Figure 4.2: Controlled plants composed of three plant agents and two control agents: (a) the graph structure for the system with dynamics (2.5),(2.7)–(2.10) and (3.3), (b) the graph structure for the system with dynamics (2.5), (2.7)–(2.10), and (4.1)–(4.2), (c) the corresponding dependency graph for both systems.

computes $u_{13}[k]$ which depends on control-signal information before time k, in addition to the state information at time k. Then, X_{13} passes the value of $u_{13}[k]$ to X_{14} who similarly computes $u_{14}[k]$ which depends on control-signal information from before time k and on $u_{13}[k]$, in addition to the state information at time k. Next, X_{14} passes the values of $u_{14}[k]$ to X_{15} so that X_{15} can compute $u_{15}[k]$ using the control-signal information from before time k and $u_{14}[k]$, in addition to the state information at time k.

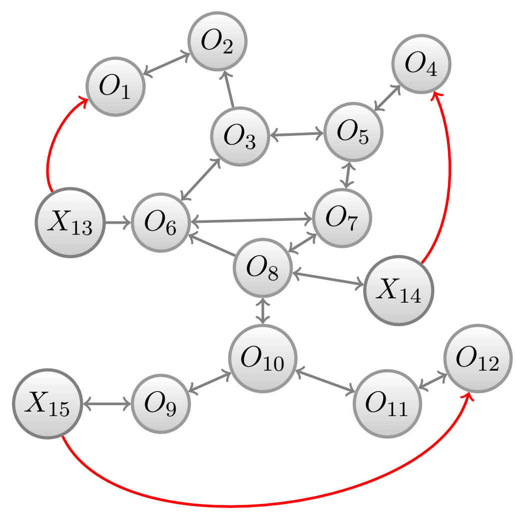
The above discussion, which extends recent work in [5] and [18], is summarized in parts (a) and (b) of the following theorem. Part (c) of the theorem is new, and, in addition to being potentially useful in automating targeting analysis, is employed in Chapter 5.

Theorem 1. *For a given plant, given set of $m \geq 2$ control agents, and given targeting assignment, the following three conditions are equivalent:*
(a) Targeting succeeds.
(b) The dependency graph has no loops.
(c) For every p in the interval $2 \leq p \leq m$ and every permutation involving p of the m control agents (denoted $(\bar{X}_1, \ldots, \bar{X}_p)$ with corresponding targets $(\bar{T}_1, \ldots, \bar{T}_p)$),

$$\sum_{i=1}^{p} \Delta(\bar{X}_i, \bar{T}_i) < \sum_{i=1}^{p-1} \Delta(\bar{X}_i, \bar{T}_{i+1}) + \Delta(\bar{X}_p, \bar{T}_1). \tag{4.3}$$

Proof: The equivalence of (a) and (b) follow immediately from the definition of computability and from the way the dependency graph is constructed. Condition (c) is a

(a)



(b)

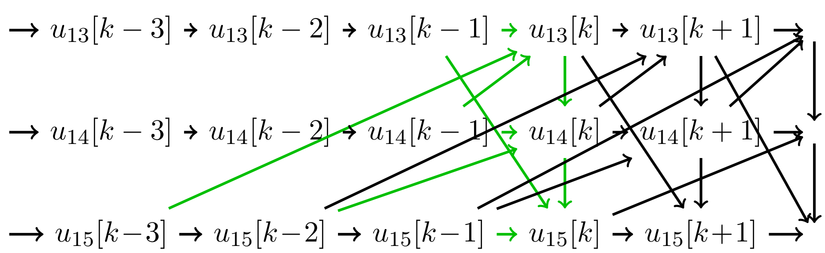


Figure 4.3: A controlled plant composed of 12 plant agents and 3 control agents, assuming all δ_i and δ_{ji} values are 1: (a) the graph structure, (b) the corresponding dependency graph.

mathematical statement that there are no loops in the dependency graph involving nodes on all p (where $2 \leq p \leq m$) of the m "horizontal lines" of the dependency graph, and therefore condition (c) is equivalent to condition (b). To derive (4.3), fix p (for $2 \leq p \leq m$) and fix the specific permutation $(\bar{X}_1, \ldots, \bar{X}_p)$ under the constraints that $\Delta(\bar{X}_i, \bar{T}_{i+1}) < \infty$ (for $1 \leq i \leq p-1$) and $\Delta(\bar{X}_p, \bar{T}_1) < \infty$. If these constraints are violated, then (4.3) holds trivially. Note that the left-hand side of (4.3) is necessarily finite by Assumption A_1. Think of the dependency graph, and focus on the p "horizontal lines" that correspond to $\bar{u}_1, \ldots, \bar{u}_p$ (where $\bar{u}_p$ is the control signal for $\bar{X}_p$). The dependency graph has various arrows, including arrows

$$\begin{aligned}
&\text{from } \bar{u}_1[k + \Delta(\bar{X}_2, \bar{T}_2) - \Delta(\bar{X}_1, \bar{T}_2)] \text{ to } \bar{u}_2[k],\\
&\text{from } \bar{u}_2[k + \Delta(\bar{X}_3, \bar{T}_3) - \Delta(\bar{X}_2, \bar{T}_3)] \text{ to } \bar{u}_3[k],\\
&\qquad\vdots\\
&\text{from } \bar{u}_{p-2}[k + \Delta(\bar{X}_{p-1}, \bar{T}_{p-1}) - \Delta(\bar{X}_{p-2}, \bar{T}_{p-1})] \text{ to } \bar{u}_{p-1}[k],\\
&\text{from } \bar{u}_{p-1}[k + \Delta(\bar{X}_p, \bar{T}_p) - \Delta(\bar{X}_{p-1}, \bar{T}_p)] \text{ to } \bar{u}_p[k],\\
&\text{from } \bar{u}_p[k + \Delta(\bar{X}_1, \bar{T}_1) - \Delta(\bar{X}_p, \bar{T}_1)] \text{ to } \bar{u}_1[k].
\end{aligned}$$

The dependency graph also has duplicates of each of the above arrows, generated by replacing k with $k \pm 1$, $k \pm 2$, etc. Specifically, on defining

$$\begin{aligned}
a_p &= \Delta(\bar{X}_p, \bar{T}_p) - \Delta(\bar{X}_{p-1}, \bar{T}_p),\\
a_{p-1} &= \Delta(\bar{X}_{p-1}, \bar{T}_{p-1}) - \Delta(\bar{X}_{p-2}, \bar{T}_{p-1}),\\
&\vdots\\
a_3 &= \Delta(\bar{X}_3, \bar{T}_3) - \Delta(\bar{X}_2, \bar{T}_3),\\
a_2 &= \Delta(\bar{X}_2, \bar{T}_2) - \Delta(\bar{X}_1, \bar{T}_2),\\
a_1 &= \Delta(\bar{X}_1, \bar{T}_1) - \Delta(\bar{X}_p, \bar{T}_1),
\end{aligned}$$

the dependency graph has arrows

$$\begin{aligned}
&\text{from } \bar{u}_1[k+a_2+a_3+\cdots+a_p] \text{ to } \bar{u}_2[k+a_3+\cdots+a_p],\\
&\text{from } \bar{u}_2[k+a_3+a_4+\cdots+a_p] \text{ to } \bar{u}_3[k+a_4+\cdots+a_p],\\
&\qquad\vdots\\
&\text{from } \bar{u}_{p-2}[k+a_{p-1}+a_p] \text{ to } \bar{u}_{p-1}[k+a_p],\\
&\text{from } \bar{u}_{p-1}[k+a_p] \text{ to } \bar{u}_p[k],\\
&\text{from } \bar{u}_p[k] \text{ to } \bar{u}_1[k-a_1],
\end{aligned}$$

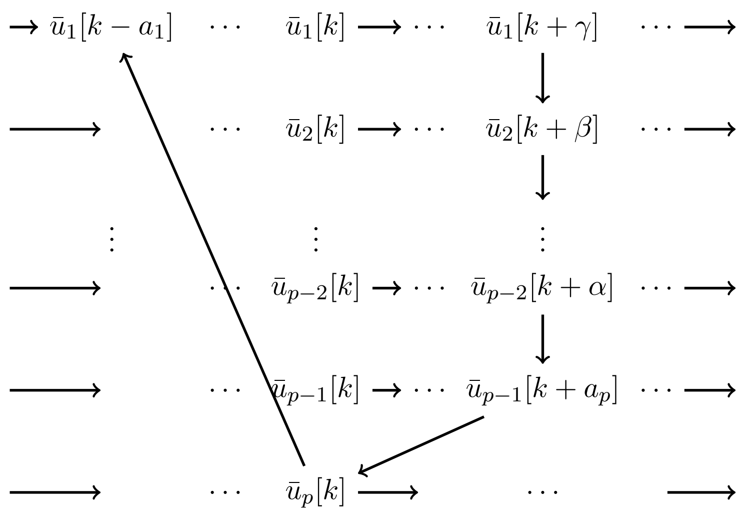


Figure 4.4: Dependency graph for the proof of Theorem 1(c). Note that $\alpha = a_{p-1} + a_p$, $\beta = a_3 + \cdots + a_{p-1} + a_p$, and $\gamma = a_2 + a_3 + \cdots + a_{p-1} + a_p$.

as can be seen in Figure 4.4. By construction, the first $p-1$ of the above arrows connect together, forming a path from the $\bar{u}_1$ "horizontal line" down to the $\bar{u}_p$ "horizontal line" of the dependency graph. The last arrow then connects the end of the path back to a node on the $\bar{u}_1$ "horizontal line." Hence, a loop involving all p arrows does not exist if and only if the end of the last arrow lies to the *right* of the start of the path, i.e., $a_2 + a_3 + \cdots + a_p < -a_1$; substituting in for $a_1, \ldots, a_p$ yields (4.3). □

To help understand condition (c) of Theorem 1, consider the case where $m = 2$. Condition (c) then generates a single inequality:

$$\Delta(X_1, T_1) + \Delta(X_2, T_2) < \Delta(X_1, T_2) + \Delta(X_2, T_1).$$

The theorem states this is a necessary and sufficient condition for targeting to succeed. Consider again the plant with dynamics (2.5)–(2.10) in Figure 4.1(a) with $\Delta(X_4, T_4) = 3$, $\Delta(X_4, T_5) = 3$, $\Delta(X_5, T_4) = 4$, and $\Delta(X_5, T_5) = 3$. The above inequality holds for this example, implying that targeting works, which is consistent with our earlier analysis with the use of dependency graph shown in Figure 4.1(b). On the other hand, the plant with dynamics (2.5),(2.7)–(2.10) and (3.3) in Figure 4.2(a) with $\Delta(X_4, T_4) = 4$, $\Delta(X_4, T_5) = 3$, $\Delta(X_5, T_4) = 4$, and $\Delta(X_5, T_5) = 3$ violates the aforementioned inequality, indicating that targeting fails for this system which was shown earlier with the existence of loops in the corresponding dependency graph in Figure 4.2(c). When $m = 3$, condition (c) of Theorem 1 generates three distinct inequalities for $p = 2$:

$$\Delta(X_1, T_1) + \Delta(X_2, T_2) < \Delta(X_1, T_2) + \Delta(X_2, T_1),$$

$$\Delta(X_1, T_1) + \Delta(X_3, T_3) < \Delta(X_1, T_3) + \Delta(X_3, T_1),$$
$$\Delta(X_2, T_2) + \Delta(X_3, T_3) < \Delta(X_2, T_3) + \Delta(X_3, T_2),$$

and two distinct inequalities for $p = 3$:

$$\Delta(X_1, T_1) + \Delta(X_2, T_2) + \Delta(X_3, T_3) < \Delta(X_1, T_2) + \Delta(X_2, T_3) + \Delta(X_3, T_1),$$

$$\Delta(X_1, T_1) + \Delta(X_2, T_2) + \Delta(X_3, T_3) < \Delta(X_1, T_3) + \Delta(X_3, T_2) + \Delta(X_2, T_1).$$

All five of these inequalities, together, are necessary and sufficient for targeting to succeed. In general, condition (c) generates a total of

$$\sum_{p=2}^{m} C_p^m \cdot (p-1)! = \sum_{p=2}^{m} m!/[p(m-p)!]$$

distinct inequalities.

4.2 Growing Analysis

Assume at this point that targeting succeeds, so the m control agents have computable control laws to zero their respective targets. We now turn to growing analysis to determine whether the m control laws happen to zero non-target plant agents in addition to the targets. Assumption A_7 makes this analysis a straightforward extension of our earlier work (see [5] and [18]). The analysis, in the form of an algorithm called the Growing Analysis Algorithm (GAA), is provided in Figure 4.5. In the analysis, Ω denotes the set of agents that are necessarily zeroed by the control laws that zero the targets. The set Ω "grows" as the algorithm proceeds, explaining the name of this analysis. The following theorem establishes two facts:

Theorem 2. *For a given plant, given set of $m \geq 1$ control agents, and given targeting assignment, assume targeting succeeds. Then the following hold:*
(a) The Growing Analysis Algorithm terminates after a finite number of iterations.
(b) If growing succeeds, then regulation of the plant is achieved with settling time $\lambda = \max\{\Delta(X_i, T_i) : n+1 \leq i \leq n+m\}$.
(c) The time complexity of the Growing Analysis Algorithm is $O(n^2)$.

Proof: The Growing Analysis Algorithm terminates after a finite number of iterations since it requires Step 2 to be executed, at most, n times. If growing succeeds, then Ω contains all n of the plant agents, and therefore the control laws zero every plant agent, which is the definition of regulation in this thesis. By Assumption A_7, all plant agents are

Step 1: Initialize $\Omega = \{T_{n+1}, \dots, T_{n+m}\}$.

Step 2: Determine if there exists a $O_j \in \Omega$ such that all agents in $\mathcal{N}(O_j)$, except for exactly one (call it O_q or X_q, depending on the type of agent), are elements of Ω. Then necessarily O_q (or X_q) is zeroed. Augment Ω with O_q (or X_q).

Step 3: Repeat Step 2 until either:

– all of $O_1, \dots, O_n$ are in Ω, in which case growing succeeds, or
– no O_j can be found satisfying the condition of Step 2, and at least one plant agent does not appear in Ω, in which case growing fails.

Figure 4.5: The Growing Analysis Algorithm (GAA).

Number of elements in Ω	Number of elements examined (worst case)
m	m
m+1	m+1
m+2	m+2
⋮	⋮
n-1	n-1
n	-

Table 4.1: Data for complexity analysis of the Growing Analysis Algorithm.

zeroed no later than the time instant when all targets are zeroed, which is determined by the largest value of $\Delta(X_i, T_i)$ for $n+1 \le i \le n+m$. Finally, the information listed in Table 4.1 is used for the complexity analysis of the Growing Analysis Algorithm. In Step 1, m elements are included in Ω. In the second step, in the worst case, all the m elements have to be checked to continue to Step 3. If Step 2 is carried out successfully, then we have to consider, in the worst case, $m+1$ elements in Step 3 and so on. Once all the n plant agents are in Ω, the GAA is terminated. Overall, the number of examinations made in the worst case is found to be:

$$\begin{aligned} \sum_{i=m}^{n-1} i &= \sum_{i=0}^{n-1} i - \sum_{i=0}^{m-1} i \\ &= \frac{n(n-1)}{2} - \frac{m(m-1)}{2}. \end{aligned}$$

Thus, the time complexity of the GAA is $O(n^2)$. □

As an example, consider once again the system with dynamics (2.5)–(2.10) in Figure 4.1(a). It was previously shown that targeting succeeds and that Assumption A_7 holds. Applying the Growing Analysis Algorithm to this system yields:

$$\begin{aligned}
\Omega &= \{O_2, O_1\} \text{ [from Step 1]}, \\
\Omega &= \{O_2, O_1, O_3\} \text{ [from Step 2 with } O_j = O_2 \text{ and } O_q = O_3].
\end{aligned}$$

Consequently, growing succeeds and the control laws in (3.1) and (3.2) regulate the plant with settling time $\lambda = \max\{3, 3\} = 3$. As another example, imagine the same system, but with X_5 or X_4 removed, leaving only one control agent. Now targeting succeeds, but growing fails. Finally, consider again the system in Figure 4.3(a) for which we showed there are no loops in the dependency graph in the previous section, implying that targeting works. Under Assumption A_7, the Growing Analysis Algorithm proceeds (non-uniquely) with

$$\begin{aligned}
\Omega &= \{O_1, O_4, O_{12}\}, \\
\Omega &= \{O_1, O_4, O_{12}, O_2\}, \\
\Omega &= \{O_1, O_4, O_{12}, O_2, O_3\}, \\
\Omega &= \{O_1, O_4, O_{12}, O_2, O_3, O_5\}, \\
\Omega &= \{O_1, O_4, O_{12}, O_2, O_3, O_5, O_6\}, \\
\Omega &= \{O_1, O_4, O_{12}, O_2, O_3, O_5, O_6, O_7\}, \\
\Omega &= \{O_1, O_4, O_{12}, O_2, O_3, O_5, O_6, O_7, O_8\}, \\
\Omega &= \{O_1, O_4, O_{12}, O_2, O_3, O_5, O_6, O_7, O_8, O_{11}\}, \\
\Omega &= \{O_1, O_4, O_{12}, O_2, O_3, O_5, O_6, O_7, O_8, O_{11}, O_{10}\}, \\
\Omega &= \{O_1, O_4, O_{12}, O_2, O_3, O_5, O_6, O_7, O_8, O_{11}, O_{10}, O_9\},
\end{aligned}$$

so growing succeeds. Thus, regulation is achieved with settling time $\lambda = \max\{5, 5, 5\} = 5$. (All δ_i and δ_{ji} values are assumed to be 1.) It should be noted that, in contrast to targeting analysis, growing analysis depends *only* on the graph structure of the plant and *not* the propagation time through the plant agents and control agents.

4.3 Communication and Sensing Requirements

In Chapter 2, we made assumptions regarding the communication requirements among the control agents and the sensing requirements for control agents to regulate a plant using

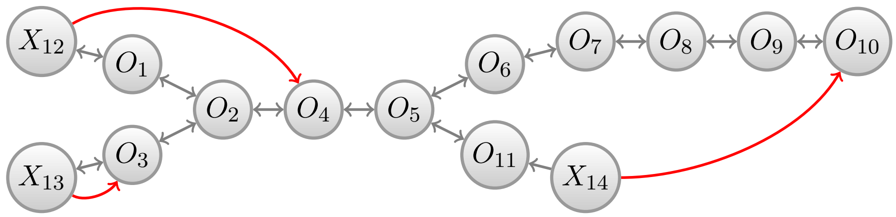


Figure 4.6: A controlled plant composed of 11 plant agents and 3 control agents.

the targeting approach (see Assumptions A_2 and A_3). For simplicity, we are assuming no constraints on communication among the control agents and sensing capabilities of the control agents. In practice, however, not all control agents need to communicate nor need to sense the state of all the plant agents and/or control agents in the system. Communication and sensing requirements are not unique for a given control scheme. In fact, communication requirements are tightly interrelated with sensing requirements and intuitively, there is (generally) a trade-off between the two: the more sensing ability the control agents have, the less communication is needed among them to regulate the plant. For instance, if each control agent can sense the whole state of the plant and other control agents, there is no need for control agents to communicate with each other. Thus, communication and sensing requirements can be determined depending on the application.

In the following, we use an example to show how communication and sensing requirements can be determined for regulation of a given plant. Consider a controlled plant as in Figure 4.6 with the following dynamics:

$$
\begin{aligned}
x_1[k+1] &= -x_1[k] + y_2[k] + y_{12}[k],\ \ y_1[k] = x_1[k] && (4.4)\\
x_2[k+1] &= y_1[k] - y_3[k] + y_4[k],\ \ y_2[k] = x_2[k] && (4.5)\\
x_3[k+1] &= y_2[k] - 2y_{13}[k] - y_4[k] + y_5[k],\ \ y_3[k] = x_3[k] && (4.6)\\
x_4[k+1] &= -y_2[k] + y_5[k],\ \ y_4[k] = x_4[k] && (4.7)\\
x_5[k+1] &= x_5[k] + y_4[k] - y_6[k] - 2y_{11}[k],\ \ y_5[k] = x_5[k] && (4.8)\\
x_6[k+1] &= -x_6[k] + y_5[k] - 3y_7[k],\ \ y_6[k] = x_6[k] && (4.9)\\
x_7[k+1] &= y_6[k] + 2y_8[k],\ \ y_7[k] = x_7[k] && (4.10)\\
x_8[k+1] &= -x_8[k] - y_7[k] + y_9[k],\ \ y_8[k] = x_8[k] && (4.11)\\
x_9[k+1] &= y_8[k] + y_{10}[k],\ \ y_9[k] = x_9[k] && (4.12)\\
x_{10}[k+1] &= x_{10}[k] + y_9[k],\ \ y_{10}[k] = x_{10}[k] && (4.13)\\
x_{11}[k+1] &= y_5[k] - y_{14}[k],\ \ y_{11}[k] = x_{11}[k] && (4.14)\\
x_{12}[k+1] &= 2u_{12}[k] + y_1[k],\ \ y_{12}[k] = x_{12}[k] && (4.15)\\
x_{13,1}[k+1] &= u_{13}[k] - y_3[k], && (4.16)\\
x_{13,2}[k+1] &= x_{13,1}[k], && (4.17)\\
x_{13,3}[k+1] &= x_{13,2}[k],\ \ y_{13}[k] = x_{13,3}[k] && (4.18)\\
x_{14}[k+1] &= -u_{14}[k],\ \ y_{14}[k] = x_{14}[k]. && (4.19)
\end{aligned}
$$

To find the control laws u_{12}, u_{13}, and u_{14} for this system, we need to find the expressions $y_4[k+\Delta(X_{12},O_4)]$, $y_3[k+\Delta(X_{13},O_3)]$ and $y_{10}[k+\Delta(X_{14},O_{10})]$ and set them all to zero (refer to Assumption A_6). Based on dynamics (4.4)–(4.19), we have $\Delta(X_{12},O_4) = 4$, $\Delta(X_{13},O_3) = 4$, and $\Delta(X_{14},O_{10}) = 8$. In the following, we find the expressions for $y_4[k+4]$, $y_3[k+4]$ and $y_{10}[k+8]$ by direct iteration through the plant dynamics:

$$
\begin{aligned}
y_4[k+4] &= x_4[k+4] = -y_2[k+3] + y_5[k+3]\\
&= -y_1[k+2] + y_3[k+2] + y_5[k+2] - y_6[k+2]\\
&\quad -2y_{11}[k+2]\\
&= y_1[k+1] + y_4[k+2] - 2y_5[k+1] + 3y_7[k+1]\\
&\quad -2y_{11}[k+1] - y_{12}[k+1] - 2y_{13}[k+1] + 2y_{14}[k+1]\\
&= -2y_1[k] - 2y_4[k] - 3y_5[k] + 5y_6[k] + 6y_8[k] + 4y_{11}[k]\\
&\quad +y_{12}[k] - 2x_{13,2}[k] + 2y_{14}[k] - 2u_{12}[k] - 2u_{14}[k]\\
&= -2x_1[k] - 2x_4[k] - 3x_5[k] + 5x_6[k] + 6x_8[k] + 4x_{11}[k]\\
&\quad +x_{12}[k] - 2x_{13,2}[k] + 2x_{14}[k] - 2u_{12}[k] - 2u_{14}[k], \qquad (4.20)
\end{aligned}
$$

$$
\begin{aligned}
y_3[k+4] &= x_3[k+4] = y_2[k+3] - 2y_{13}[k+3] \\
&= y_1[k+2] - y_3[k+2] + y_4[k+2] - 2u_{13}[k] + 2y_3[k] \\
&= -y_1[k+1] - y_2[k+1] + y_5[k+1] + y_{12}[k+1] \\
&\quad +2y_{13}[k+1] - 2u_{13}[k] + 2y_3[k] \\
&= y_1[k] - y_2[k] + 3y_3[k] + y_5[k] - y_6[k] - 2y_{11}[k] \\
&\quad -y_{12}[k] + 2x_{13,2}[k] + 2u_{12}[k] - 2u_{13}[k] \\
&= x_1[k] - x_2[k] + 3x_3[k] + x_5[k] - x_6[k] - 2x_{11}[k] \\
&\quad -x_{12}[k] + 2x_{13,2}[k] + 2u_{12}[k] - 2u_{13}[k], \qquad (4.21)
\end{aligned}
$$

$$
\begin{aligned}
y_{10}[k+8] &= x_{10}[k+8] = x_{10}[k+7] + y_9[k+7] \\
&= y_8[k+6] + y_9[k+6] + 2y_{10}[k+6] \\
&= -y_7[k+5] + 3y_9[k+5] + 3y_{10}[k+5] \\
&= -y_6[k+4] + y_8[k+4] + 3y_9[k+4] + 6y_{10}[k+4] \\
&= -y_5[k+3] + y_6[k+3] + 2y_7[k+3] \\
&\quad +2y_8[k+3] + 7y_9[k+3] + 9y_{10}[k+3] \\
&= -y_4[k+2] + 2y_6[k+2] - 5y_7[k+2] + 9y_8[k+2] \\
&\quad +11y_9[k+2] + 16y_{10}[k+2] + 2y_{11}[k+2] \\
&= y_2[k+1] + 3y_5[k+1] - 7y_6[k+1] - 15y_7[k+1] \\
&\quad -8y_8[k+1] + 25y_9[k+1] + 27y_{10}[k+1] - 2y_{14}[k+1] \\
&= y_1[k] - y_3[k] + 4y_4[k] - 4y_5[k] - 11y_6[k] + 29y_7[k] \\
&\quad +3y_8[k] + 19y_9[k] + 52y_{10}[k] - 6y_{11}[k] + 2u_{14}[k] \\
&= x_1[k] - x_3[k] + 4x_4[k] - 4x_5[k] - 11x_6[k] + 29x_7[k] \\
&\quad +3x_8[k] + 19x_9[k] + 52x_{10}[k] - 6x_{11}[k] + 2u_{14}[k]. \qquad (4.22)
\end{aligned}
$$

Setting (4.20), (4.21), and (4.22) to zero, control laws are found to be:

$$
\begin{aligned}
u_{12}[k] &= \frac{1}{2}(-2x_1[k] - 2x_4[k] - 3x_5[k] + 5x_6[k] + 6x_8[k] \\
&\quad +4x_{11}[k] + x_{12}[k] - 2x_{13,2}[k] + 2x_{14}[k] - 2u_{14}[k]), \qquad (4.23)
\end{aligned}
$$

$$
\begin{aligned}
u_{13}[k] &= \frac{1}{2}(x_1[k] - x_2[k] + 3x_3[k] + x_5[k] - x_6[k] \\
&\quad -2x_{11}[k] - x_{12}[k] + 2x_{13,2}[k] + 2u_{12}[k]), \qquad (4.24)
\end{aligned}
$$

$$
\begin{aligned}
u_{14}[k] &= \frac{-1}{2}(x_1[k] - x_3[k] + 4x_4[k] - 4x_5[k] - 11x_6[k] + 29x_7[k] \\
&\quad +3x_8[k] + 19x_9[k] + 52x_{10}[k] - 6x_{11}[k]). \qquad (4.25)
\end{aligned}
$$

Looking at expressions (4.23), (4.24), and (4.25), we can deduce that:

- Control agent X_{14} needs to sense the state of all the plant agents except O_2 to be able to compute $u_{14}[k]$.
- Control agent X_{12} needs to measure the state of plant agents O_1, O_4, O_5, O_6, O_8, and O_{11}, and the state of X_{14} and partly X_{13}, and also communicate with control agent X_{14} to obtain $u_{14}[k]$ to compute $u_{12}[k]$.
- Control agent X_{13} can compute u_{13} if it is able to measure the state of plant agents O_1, O_2, O_3, O_5, O_6, and O_{11}, and the state of X_{12}. Also X_{13} must communicate with control agent X_{12} to obtain $u_{12}[k]$.

Manipulating these expressions, however, can result in different communication and sensing requirements. As an example, substituting (4.25) into (4.23) yields:

$$\begin{aligned} u_{12}[k] &= \frac{1}{2}(-x_1[k] - x_3[k] + 2x_4[k] - 7x_5[k] - 6x_6[k] \\ &\quad +29x_7[k] + 9x_8[k] + 19x_9[k] + 52x_{10}[k] - 2x_{11}[k] \\ &\quad +x_{12}[k] - 2x_{13,2}[k] + 2x_{14}[k]). \end{aligned} \tag{4.26}$$

And substituting (4.26) into (4.24) gives:

$$\begin{aligned} u_{13}[k] &= \frac{1}{2}(-x_2[k] + 2x_3[k] + 2x_4[k] - 6x_5[k] - 7x_6[k] \\ &\quad +29x_7[k] + 9x_8[k] + 19x_9[k] + 52x_{10}[k] \\ &\quad -4x_{11}[k] + 2x_{14}[k]). \end{aligned} \tag{4.27}$$

If expressions (4.25), (4.26) and (4.27) are used by the control agents to compute the control laws, then the control agents will no longer need to communicate with each other. The sensing requirements for control agents X_{12} and X_{13} also change: Control agent X_{12} has to sense the state of O_3, O_7, O_9, and O_{10} in addition to the previous sensing requirements. Control agent X_{13} does not need to sense the state of O_1 and X_{12} any more; however, X_{13} now has to sense O_4, O_7, O_8, O_9, O_{10}, and X_{14}.

Constructing the dependency graph can also be helpful in determining the communication requirements (if any) among the control agents. Figure 4.7 shows the dependency graph for the system in Figure 4.6. Note that this dependency graph is consistent with (4.23), (4.24), and (4.25), even though this might not be apparent from the expressions. However, further iterations through the plant dynamics confirms the constructed dependency graph. For instance, further iterations of (4.23) show that $u_{12}[k]$ depends on $u_{13}[k-2]$ in addition to $u_{14}[k]$ as seen in Figure 4.7. The communication requirements can be determined based on the sequence in which the control signals must be computed; i.e., as it can

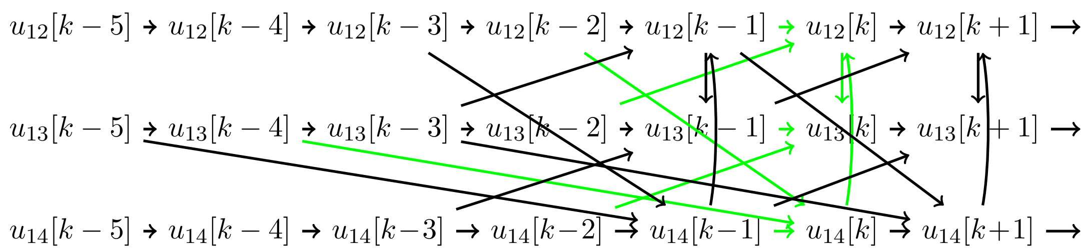


Figure 4.7: The dependency graph of the system in Figure 4.6 with no loops, implying that targeting succeeds.

k	$x_1[k]$	$x_2[k]$	$x_3[k]$	$x_4[k]$	$x_5[k]$	$x_6[k]$	$x_7[k]$	$x_8[k]$	$x_9[k]$	$x_{10}[k]$	$x_{11}[k]$
0	2	-1	1	4	-1	2	3	-1	-2	1	2
1	-3	5	-1	0	-3	-12	0	-4	0	-1	-1
2	98	-2	5	-8	11	9	-20	4	-5	-1	-45.5
3	83	85	-2	13	85	62	17	11	3	-6	-31
4	-568	98	0	0	98	-28	84	-25	5	-3	347
5	-1066	-568	0	0	-568	-126	-78	-54	-28	2	312
6	-156	-1066	0	0	-1066	-208	-234	104	-52	-26	-351
7	0	-156	0	0	-156	-156	0	78	78	-78	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0

Table 4.2: Simulation results of the states of the plant agents in Figure 4.6. Note that the settling time is $\lambda = 8$.

be seen in Figure 4.7, at time k, first X_{14} computes $u_{14}[k]$ using control-signal information from time $k-1$ (in addition, of course, to state information at time k), next communicates that information to X_{12}. Then X_{12} similarly computes $u_{12}[k]$, and passes the value of $u_{12}[k]$ to X_{13} so that X_{13} can compute $u_{13}[k]$. As mentioned before, regulation of this system can be achieved through other communication and sensing requirements set for the control agents (refer to (4.25), (4.26), and (4.27)).

Finally, to confirm that regulation is achieved using the control laws in (4.23), (4.24), and (4.25), the plant with dynamics (4.4)–(4.19) was simulated in MATLAB using arbitrary initial conditions. The obtained values of the states of the plant agents and control agents over time steps $k = 1$ to $k = 8$ have been summarized in Tables 4.2 and 4.3. As it can be seen in Table 4.2, all the plant agents get zeroed at $k = 8$ as the settling time for this plant is $\lambda = \max\{4, 4, 8\} = 8$. Lastly, a summary of the values of the control signals over the specified time steps is given in Table 4.4.

Time Step (k)	$x_{12}[k]$	$x_{13,1}[k]$	$x_{13,2}[k]$	$x_{13,3}[k]$	$x_{14}[k]$
0	0	0	0	0	0
1	90	42.5	0	0	42.5
2	183	49	42.5	0	42
3	-570	-284	49	42.5	-262
4	-1732	-533	-284	49	-214
5	-654	-78	-533	-284	-217
6	910	0	-78	-533	-1066
7	156	0	0	-78	-156
8	0	0	0	0	0
9	0	0	0	0	0

Table 4.3: Simulation results of the states of the control agents in Figure 4.6.

Time Step (k)	$u_{12}[k]$	$u_{13}[k]$	$u_{14}[k]$
0	44	43.5	-42.5
1	93	48	-42
2	-334	-279	262
3	-907.5	-535	214
4	-43	-78	217
5	988	0	1066
6	156	0	156
7	0	0	0
8	0	0	0
9	0	0	0

Table 4.4: Simulation results of the control signals in Figure 4.6.

Chapter 5

Necessary Conditions for Targeting and/or Growing to Succeed

In the previous chapter we dealt with Problem 1. This chapter addresses Problem 2. To help a designer determine how many control agents should be used, where control agents should be placed, and what targeting assignment should be used, two theorems are given in this chapter that provide several intuitively-appealing necessary conditions to ensure targeting and/or growing succeed. Theorem 3 uses Condition (c) of Theorem 1 (that is, inequality (4.3)) to derive two necessary conditions for targeting to succeed. Theorem 4 gives two necessary conditions for growing to work.

Theorem 3 is stated below. Part (b) of the theorem concerns intersections between fastest paths from control agents to their respective targets. A *fastest path* from X_i to T_i refers to a (possibly non-unique) path from X_i to T_i, say $X_i \to O_\alpha \to O_\beta \to \ldots \to O_\zeta \to O_j$, such that:

$$\delta_i + \delta_{i\alpha} + \delta_{\alpha\beta} + \cdots + \delta_{\zeta j} = \Delta(X_i, O_j).$$

For example, consider again the controlled plant composed of three plant agents and two control agents with dynamics (2.5)–(2.10) as shown in Figure 5.1(a). Given that $\Delta(X_4, T_4) = 3$, the only fastest path from X_4 to T_4 in this figure is $X_4 \to O_3 \to O_2$ with $\delta_4 + \delta_{43} + \delta_{32} = 3$; $X_4 \to O_3 \to O_1 \to O_2$ is a path, but not a fastest path as $\delta_4 + \delta_{43} + \delta_{31} + \delta_{12} = 4$.

Theorem 3. *For a given plant, given set of $m \geq 2$ control agents, and given targeting assignment, assume that targeting succeeds. Then both the following hold:*
(a) Propagation times along the paths from control agents to respective targets must be, on average, less than propagation times along the paths from control agents to all other

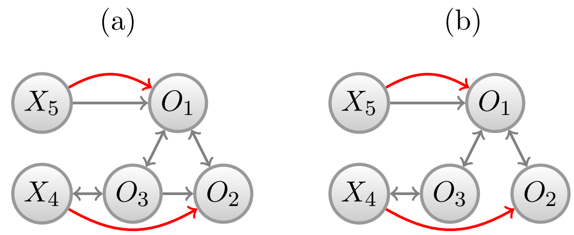


Figure 5.1: Controlled plants composed of three plant agents and two control agents: (a) the graph structure for the system with dynamics (2.5)–(2.10), (b) the graph structure for the system with dynamics (2.5), (2.7)–(2.10), and (3.3).

targets:

$$\frac{1}{m}\sum_{i=n+1}^{n+m}\Delta(X_i,T_i) < \frac{1}{m(m-1)}\sum_{i=n+1}^{n+m}\sum_{\substack{j=n+1\\ j\neq i}}^{n+m}\Delta(X_i,T_j) \tag{5.1}$$

(b) There are no nodes in common between a fastest path connecting X_i to T_i (for $n+1 \leq i \leq n+m$) and a fastest path connecting X_j to T_j (for $n+1 \leq j \leq n+m, j \neq i$).

Proof: For $p = 2$, condition (4.3) generates $m(m-1)/2$ unique inequalities that are necessary for targeting to succeed, i.e.,

- For (X_{n+1}, X_{n+2}):

$$\Delta(X_{n+1},T_{n+1}) + \Delta(X_{n+2},T_{n+2}) < \Delta(X_{n+1},T_{n+2}) + \Delta(X_{n+2},T_{n+1}),$$

- For (X_{n+1}, X_{n+3}):

$$\Delta(X_{n+1},T_{n+1}) + \Delta(X_{n+3},T_{n+3}) < \Delta(X_{n+1},T_{n+3}) + \Delta(X_{n+3},T_{n+1}),$$

$$\vdots$$

- For (X_{n+1}, X_{n+m}):

$$\Delta(X_{n+1},T_{n+1}) + \Delta(X_{n+m},T_{n+m}) < \Delta(X_{n+1},T_{n+m}) + \Delta(X_{n+m},T_{n+1}),$$

- For (X_{n+2}, X_{n+3}):

$$\Delta(X_{n+2},T_{n+2}) + \Delta(X_{n+3},T_{n+3}) < \Delta(X_{n+2},T_{n+3}) + \Delta(X_{n+3},T_{n+2}),$$

- For (X_{n+2}, X_{n+4}):

$$\Delta(X_{n+2}, T_{n+2}) + \Delta(X_{n+4}, T_{n+4}) < \Delta(X_{n+2}, T_{n+4}) + \Delta(X_{n+4}, T_{n+2}),$$

$$\vdots$$

- For (X_{n+2}, X_{n+m}):

$$\Delta(X_{n+2}, T_{n+2}) + \Delta(X_{n+m}, T_{n+m}) < \Delta(X_{n+2}, T_{n+m}) + \Delta(X_{n+m}, T_{n+2}),$$

$$\vdots$$

- For (X_{n+m-2}, X_{n+m-1}):

$$\Delta(X_{n+m-2}, T_{n+m-2})+\Delta(X_{n+m-1}, T_{n+m-1}) < \Delta(X_{n+m-2}, T_{n+m-1})+\Delta(X_{n+m-1}, T_{n+m-2}),$$

- For (X_{n+m-2}, X_{n+m}):

$$\Delta(X_{n+m-2}, T_{n+m-2}) + \Delta(X_{n+m}, T_{n+m}) < \Delta(X_{n+m-2}, T_{n+m}) + \Delta(X_{n+m}, T_{n+m-2}),$$

- For (X_{n+m-1}, X_{n+m}):

$$\Delta(X_{n+m-1}, T_{n+m-1}) + \Delta(X_{n+m}, T_{n+m}) < \Delta(X_{n+m-1}, T_{n+m}) + \Delta(X_{n+m}, T_{n+m-1}).$$

Add these $m(m-1)/2$ inequalities together to get

$$(m-1)\sum_{i=n+1}^{n+m} \Delta(X_i, T_i) < \sum_{i=n+1}^{n+m} \sum_{\substack{j=n+1 \\ j \neq i}}^{n+m} \Delta(X_i, T_j).$$

Multiply each side by $1/m(m-1)$ to obtain (5.1).

To prove condition (b), we show that, if there is an intersection between a fastest path from, say, $\bar{X}_1$ to $\bar{T}_1$ and a fastest path from $\bar{X}_2$ to $\bar{T}_2$, then necessarily (4.3) is violated, implying that targeting fails. To this end, consider the setup in Figure 5.2 in which O_p denotes a node that lies on both a fastest path from $\bar{X}_1$ to $\bar{T}_1$ and a fastest path from $\bar{X}_2$ to $\bar{T}_2$. As indicated in the figure, define

$$a = \Delta(\bar{X}_1, O_p),$$

$$b = \Delta(O_p, \bar{T}_1),$$

$$c = \Delta(O_p, \bar{T}_2),$$

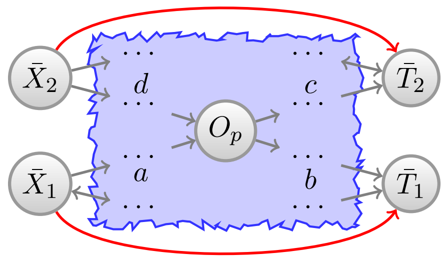


Figure 5.2: System for the proof of Theorem 3(b).

and

$$d = \Delta(\bar{X}_2, O_p).$$

Any number of plant agents and control agents can appear in the "cloud" in the figure.

Since $\bar{X}_1 \to \cdots \to O_p \to \cdots \to \bar{T}_1$ is assumed to be a fastest path, $\Delta(\bar{X}_1, \bar{T}_1) = a + b$. Likewise, $\Delta(\bar{X}_2, \bar{T}_2) = c + d$. Summing these two equalities leads to

$$\Delta(\bar{X}_1, \bar{T}_1) + \Delta(\bar{X}_2, \bar{T}_2) = a + b + c + d. \tag{5.2}$$

Moreover, since $\bar{X}_2 \to \cdots \to O_p \to \cdots \to \bar{T}_1$ is one path from $\bar{X}_2$ to $\bar{T}_1$, we know $\Delta(\bar{X}_2, \bar{T}_1) \leq b + d$ (see Figure 5.2). Similarly, $\Delta(\bar{X}_1, \bar{T}_2) \leq a + c$. Summing these last two inequalities leads to

$$\Delta(\bar{X}_2, \bar{T}_1) + \Delta(\bar{X}_1, \bar{T}_2) \leq a + b + c + d. \tag{5.3}$$

Compare (5.2) and (5.3) to conclude

$$\Delta(\bar{X}_1, \bar{T}_1) + \Delta(\bar{X}_2, \bar{T}_2) \geq \Delta(\bar{X}_2, \bar{T}_1) + \Delta(\bar{X}_1, \bar{T}_2),$$

which violates (4.3) for one of the permutations associated with $p = 2$. □

Theorem 3(a) is an encouraging result since, in some applications (certainly in the crowd-control application that motivated our work), it is desirable for control agents and respective targets to be close to one another; the theorem says that this, in fact, must be the case (on average) for targeting to succeed. Consider again the system in Figure 5.3. In the previous chapter, it was shown that, assuming all $\delta_i = 1$ and $\delta_{ji} = 1$, targeting succeeds for this example. By calculating the propagation times between control agents and targets, it is easy to verify that (5.1) holds as expected; i.e.,

$$\begin{aligned} \frac{1}{3}(\Delta(X_{13}, T_{13}) + \Delta(X_{14}, T_{14}) + \Delta(X_{15}, T_{15})) \quad < \quad & \frac{1}{6}(\Delta(X_{13}, T_{14}) + \Delta(X_{13}, T_{15}) + \Delta(X_{14}, T_{13}) \\ + \quad & \Delta(X_{14}, T_{15}) + \Delta(X_{15}, T_{13}) + \Delta(X_{15}, T_{14})) \end{aligned}$$

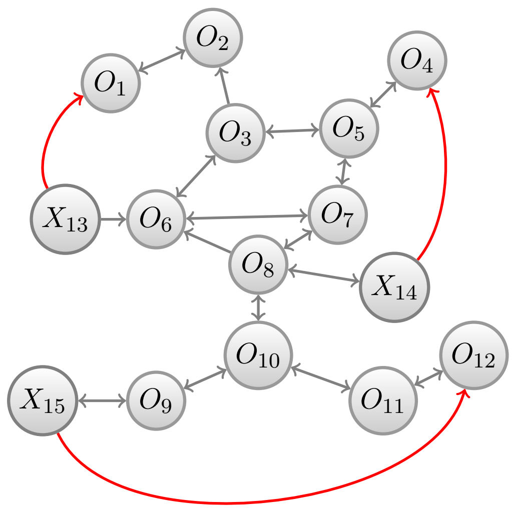


Figure 5.3: A controlled plant composed of 12 plant agents and 3 control agents.

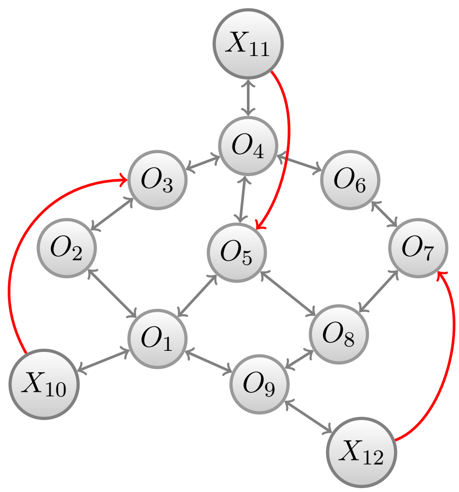
as $\Delta(X_{13}, T_{13}) = 5$, $\Delta(X_{14}, T_{14}) = 5$, $\Delta(X_{15}, T_{15}) = 5$, $\Delta(X_{13}, T_{14}) = 5$, $\Delta(X_{13}, T_{15}) = 6$, $\Delta(X_{14}, T_{13}) = 6$, $\Delta(X_{14}, T_{15}) = 5$, $\Delta(X_{15}, T_{13}) = 8$, and $\Delta(X_{15}, T_{14}) = 7$. As another example, Figure 5.4 illustrates a system for which targeting fails and yet (5.1) is satisfied, showing that, although necessary, (5.1) is not sufficient for targeting to succeed.

Condition (b) of Theorem 3 generalizes a result that appears in [18], which applies only to the specialized dynamics of the crowd-control application. Condition (b) is appealing because it is easy to verify and, when working on Problem 2, it can be used to greatly reduce the number of combinations of control agent locations and target locations that need to be considered. The implication of this result can be seen in Chapter 6. The condition is consistent with all previous examples: fastest paths do not intersect for the system in Figure 5.1(a) with dynamics (2.5)–(2.10) for which it was shown earlier that targeting succeeds. On the other hand, fastest paths do intersect for the system in Figure 5.1(b), implying that targeting fails, as previously verified. Finally, it is possible to find examples where fastest paths do not intersect, and yet targeting fails such as the system in Figure 5.5; hence, having non-intersecting fastest paths is a necessary, but not sufficient, condition for targeting to succeed.

The next theorem provides two necessary conditions for growing to succeed:

Theorem 4. *For a given plant, given set of $m \geq 1$ control agents, and given targeting assignment, assume that targeting succeeds. Then growing succeeds only if both the following hold:*
(a) Each plant agent lies on the fastest path from some X_i to its associated T_i.
(b) For each X_i, the fastest path from X_i to T_i is unique.

(a)



(b)

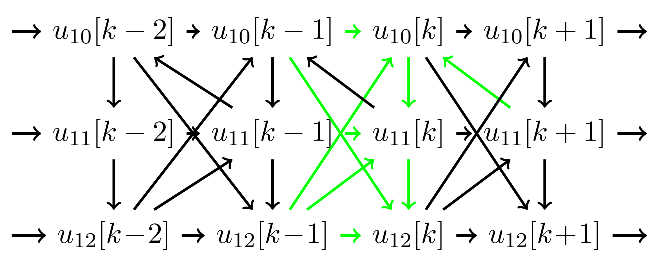


Figure 5.4: System with all $\delta_i = 1$ and $\delta_{ji} = 1$ for which targeting fails and yet (5.1) is satisfied as $\Delta(X_{10}, T_{10}) = 4$, $\Delta(X_{11}, T_{11}) = 3$, $\Delta(X_{12}, T_{12}) = 4$, $\Delta(X_{10}, T_{11}) = 3$, $\Delta(X_{10}, T_{12}) = 5$, $\Delta(X_{11}, T_{10}) = 3$, $\Delta(X_{11}, T_{12}) = 4$, $\Delta(X_{12}, T_{10}) = 5$, and $\Delta(X_{12}, T_{11}) = 4$; i.e., the left hand side of (5.1) has a value of $11/3$, and the right hand side has a value of 4: (a) the graph structure of the system, (b) the corresponding dependency graph with loops which shows targeting fails.

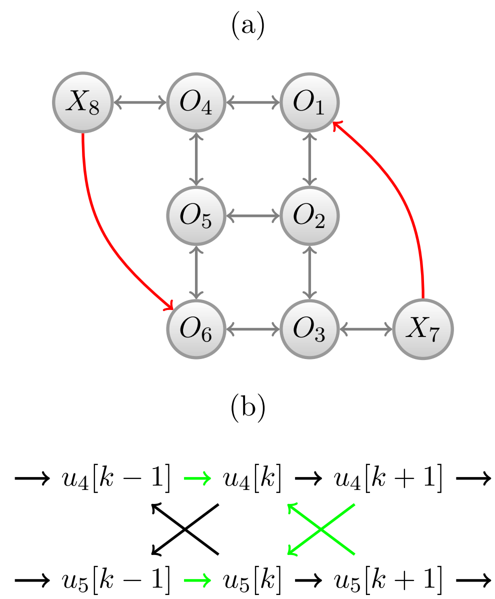


Figure 5.5: System for which fastest paths do not intersect, and yet targeting fails (assuming all $\delta_i = 1$ and $\delta_{ji} = 1$): (a) the graph structure of the system, (b) the corresponding dependency graph with loops which shows targeting fails.

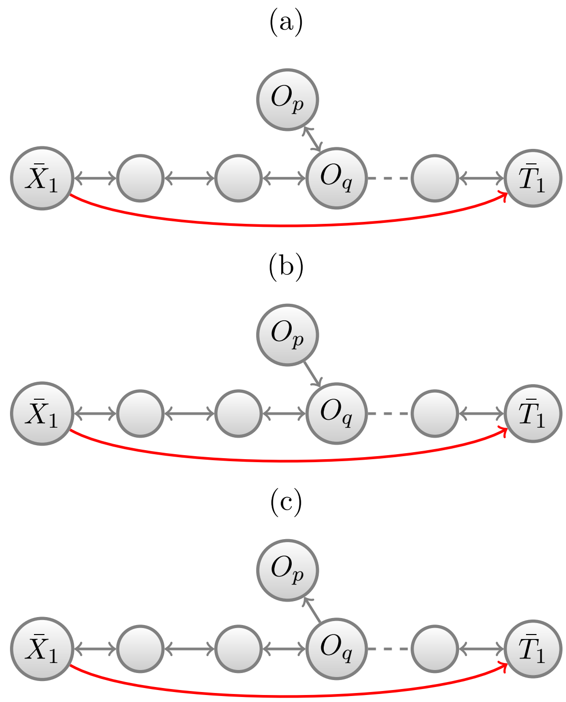


Figure 5.6: Systems for the proof of Theorem 4. The horizontal paths are the fastest paths from control agents to targets.

Proof: Label the m control agents and respective targets as $\bar{X}_1, \ldots, \bar{X}_m$ and $\bar{T}_1, \ldots, \bar{T}_m$. The crux of the proof is demonstrated in Figure 5.6(a)-(c), in which $m = 1$ and there is one path from $\bar{X}_1$ to $\bar{T}_1$, and in which O_p is an extra plant agent not on the path from $\bar{X}_1$ to $\bar{T}_1$, but instead it has a single link to that path, joining at node O_q. The types of links along the path from $\bar{X}_1$ to $\bar{T}_1$ (i.e., whether they are bidirectional or unidirectional) is not relevant here (subject, of course, to satisfying Assumption A_1). The three systems in Figure 5.6(a)-(c) differ only in the type of link between O_p and O_q. In Figures 5.6(a) and (b), the GAA terminates with $\Omega = \{\bar{T}_1, \ldots, O_q\}$, i.e., neither O_p nor nodes to the left of O_q are included in Ω, and therefore growing fails. Growing also fails in Figure 5.6(c), but this time the only node not in Ω is O_p. All other cases can be reduced to one of the three situations in Figures 5.6(a)-(c):

Case A: Consider an arbitrary plant with m control agents, where there is only one extra plant agent (again denoted O_p) that does not lie on any of the fastest paths. Figure 5.7 shows an example of such a situation for $m = 3$. It should be noted that we can arrange the fastest paths between control agents and respective targets as shown in Figure 5.7 because of the assumption that targeting works, implying that Theorem 3(b) holds (i.e., there are

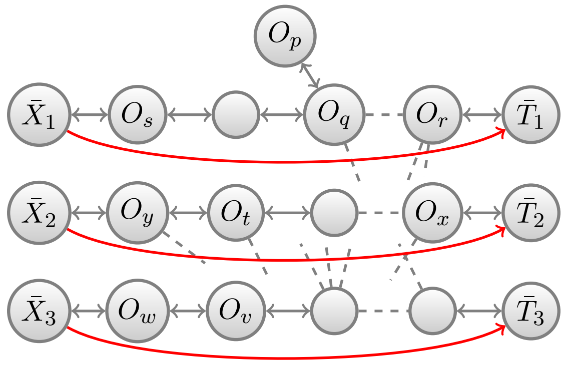


Figure 5.7: Systems for the proof of Theorem 4 (Case A). The horizontal paths are the fastest paths from control agents to targets.

no nodes in common between any two fastest paths). In Case A, node O_p has exactly one link to one of the fastest paths, assumed without loss of generality to be that from $\bar{X}_1$ to $\bar{T}_1$. Denote the node that is connected to O_p by O_q. There can be additional links connecting the various fastest paths (shown as dashed grey lines in Figure 5.7) subject, in Case A, to there being no additional links terminating at nodes to the left of O_q on the first fastest path. This situation is almost identical to those in Figures 5.6(a)-(c): at best, growing fails because O_p (and, depending on the type of link between O_p and O_q, possibly all nodes to the left of O_q on the fastest path from $\bar{X}_1$ to $\bar{T}_1$) is excluded from Ω.

Case B: Case B is identical to Case A except now there is assumed to be at least one link between a node to the left of O_q and some other node on a different fastest path; the grey lines in Figure 5.8 show an example of such a setup. As in Case A, there can also be links between agents to the right of O_q, or O_q itself, and other plant agents. Apply the GAA to this system, starting along the first fastest path: as in Case A, Ω potentially grows until it includes all nodes on the first fastest path between $\bar{T}_1$ and O_q. Now, if O_p is connected to O_q as in Figure 5.6(c), at best all the plant agents can be zeroed, except not O_p; thus, growing fails. On the other hand, if O_p is connected to O_q as in Figure 5.6(a) or (b), then once O_q is zeroed, growing cannot proceed further than O_q along the first fastest path. If there are no *outgoing* links from nodes to the left of O_q, then the nodes to the left of O_q will never be part of Ω, and growing fails. But if there is an outgoing link from a node (say O_s) to the left of O_q on the first fastest path to another node (say O_t) on another (say the second) fastest path, as shown with the diagonal grey lines in Figure 5.8, potentially the GAA results in $O_s \in \Omega$. However, applying the GAA in this situation leads to the conclusion that growing can possibly continue along the fastest path that contains O_t, but O_s does not lie on this fastest path (just like O_p, which does not lie on the first fastest path), so at best, Ω grows along the second fastest path from $\bar{T}_2$ until O_t. If there are no

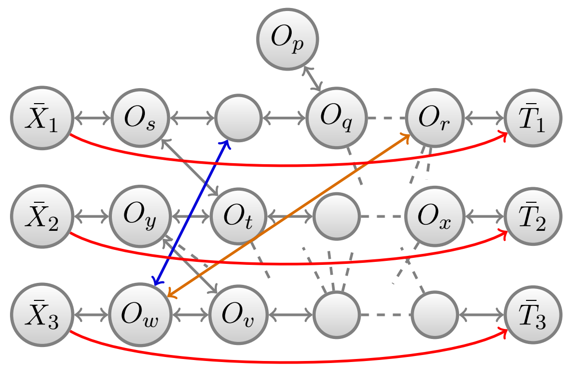


Figure 5.8: Systems for the proof of Theorem 4 (Case B). The horizontal paths are the fastest paths from control agents to targets.

links between agents to the left of O_t on the second fastest path and other plant nodes, growing cannot proceed beyond O_t on the second fastest path, leading to a situation like Case A, but now involving the second fastest path instead of the first fastest path. If there is no outgoing link from a node to the left of O_t on the second fastest path, then, at best, Ω excludes nodes to the left of O_t on the second fastest path (plus those to the left of O_q, which includes O_s, on the first fastest path), and growing fails. If, however, there is at least one outgoing link from a node to the left of O_t on the second fastest path to a node on another fastest path, the whole argument can be repeated for the latter fastest path. Following this reasoning, one of two situations arises:

Case B.1: If there are no outgoing links from nodes to the left of the link connected to the last fastest path (say at node O_v), again as shown with grey lines in Figure 5.8, then at best Ω excludes all nodes to the left of O_q, to the left of O_t, ..., and to the left of O_v. Hence, growing fails.

Case B.2: If there is at least one outgoing link from a node that is to the left of O_v (say O_w), that link can terminate at another node that lies to the *left* of O_q (if on the first fastest path) or the *left* of O_t (if on the second fastest path), etc. The blue line in Figure 5.8 shows such a link. Growing fails exactly as in Case B.1; the extra (blue) link has no effect. On the other hand, the outgoing link could terminate exactly on or to the right of O_q (if on the first fastest path) or exactly on or to the right of O_t (if on the second fastest path), etc. Denote the node at which the link terminates by O_r. The orange link in Figure 5.8 shows such a connection. Growing now fails (for the system composed of orange and grey lines in Figure 5.8) for the same reason it fails in Case A, except now O_r plays the role of O_q and O_w plays the role of O_p. If the orange link terminates along some other fastest path (say terminating at O_x instead of O_r in Figure 5.8), similar arguments can be made to conclude that growing still fails. Finally, the presence of links between any other pair

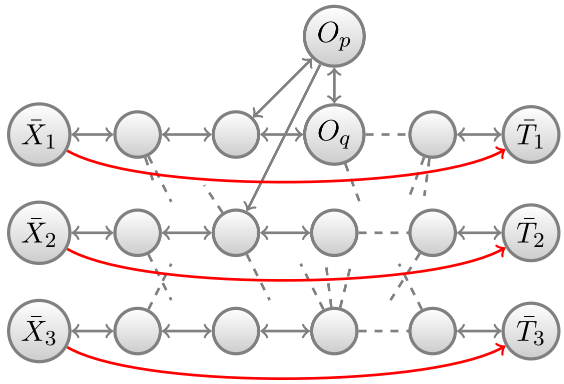


Figure 5.9: Systems for the proof of Theorem 4 (Case C). The horizontal paths are the fastest paths from control agents to targets.

of nodes excluding O_p (e.g., between O_s and O_y, or between $\bar{T}_1$ and O_x, in Figure 5.8) has no effect on these conclusions.

Case C: Case C is the same as Case B except multiple connections between O_p and the rest of the system are allowed. Assume, without loss of generality, that one of the connections terminates at the fastest path from $\bar{X}_1$ to $\bar{T}_1$, and let O_q denote the right-most connecting node on that path. An example is shown in Figure 5.9. Similar arguments as those used in Case B can be applied here, leading to the conclusion that once growing finishes, Ω includes, at most, the same nodes that would have resulted if the only connection to O_p was the one between O_p and O_q. Hence, growing fails.

Case D: Case D allows for any number of plant agents that do not lie on fastest paths between control agents and respective targets. An example of this setup is given in Figure 5.10. Once again, by applying the GAA first to the first fastest path, then moving on the second fastest path, etc., we conclude that growing fails just as in Case C. For example, with the system in Figure 5.10, Ω cannot contain O_p, O_r, or O_s (in addition to possibly other nodes, as before).

Cases A-D prove Condition (a) of Theorem 4, that is, growing works only if each plant agent lies on the fastest path from some X_i to its associated T_i. Condition (b) of Theorem 4 also follows from the arguments made in Cases A-D, where the presence of plant agents that do not lie on fastest paths between control agents and respective targets results in non-unique fastest paths from some $\bar{X}_i$ to $\bar{T}_i$. Note that we do not consider the case of non-unique fastest paths with plant agents that lie on some (other) fastest paths, since Condition (b) of Theorem 3 would be violated in such a case, contradicting our assumption that targeting succeeds. □

The conditions of Theorem 4 are easy to verify, even for large n. As examples, the

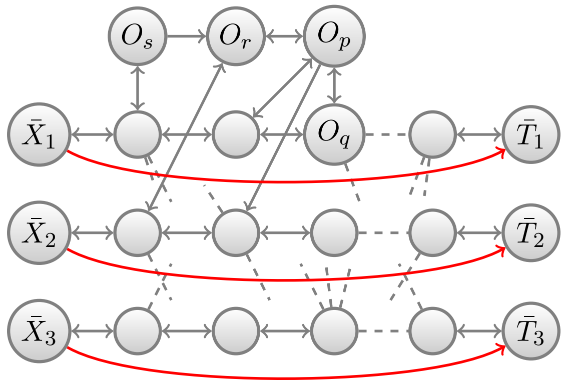


Figure 5.10: Systems for the proof of Theorem 4 (Case D). The horizontal paths are the fastest paths from control agents to targets.

system in Figures 5.1(a) meet both conditions in Theorem 4, consistent with earlier analysis that showed both targeting and growing succeed. On the other hand, if even one of the control agents were removed (leaving the other control agent and target untouched), then certain plant agents would not lie on a fastest path, and Theorem 4(a) implies that growing fails. Lastly, we conjecture that the converse of Theorem 4 holds; that is, assuming targeting works for a given plant, set of control agents and targeting assignment, growing succeeds if both the following hold: (a) each plant agent lies on the fastest path from some control agent to its associated target, (b) for each control agent X_i, the fastest path from X_i to T_i is unique. The proof of this conjecture is left for future work.

In the next chapter, these results are applied to certain graph structures (such as queues, grids, etc.) to help solve Problem 2; that is, to determine how many control agents are needed for the regulation of a given plant using computable control laws, where they should be placed, and how targets should be assigned.

Chapter 6

Examples of Various Graph Structures

In this chapter, we consider different graph structures and address Problem 2; i.e., we deduce the number of control agents, their proper placement and targeting assignment needed for both targeting and growing to succeed. For this purpose, we mostly make use of Theorems 3 and 4 as they greatly reduce the number of possible arrangements of control agents and targets for successful regulation of a system. The graph structures considered in the following are queues, grids, spiders, rings, wheels, complete graphs and null graphs. For simplicity, we assume that all the links in the following examples are bidirectional, and that the δ_i and δ_{ji} values are all equal to some $\delta \geq 1$, unless otherwise specified. We present a control scheme for each plant structure with the minimum number of control agents required for regulation. Note that the arguments made are just one way of using the aforementioned theorems to reach the following deductions. We also use Theorem 2(b) to specify the settling time in terms of δ. In the remainder of this chapter, we provide variations of a few of the examples studied earlier, assuming either unidirectional links in the plant or different δ_i and δ_{ji} values. Finally, we end this chapter with remarks on the difficulty level of the control problem with respect to the plant graph structure. It should be noted that *any* plant — subject to Assumptions A_2, A_3, A_5, and A_6 — can be regulated by setting $m = n$, linking control agent X_{n+i} directly to O_i and no other node, and using $T_{n+i} = O_i$. Hence, the minimum number of control agents needed for regulation is certainly never more than n.

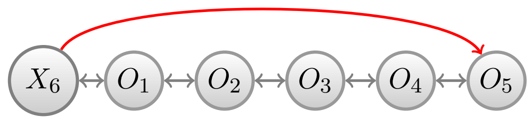


Figure 6.1: Example of a queue plant structure with $n = 5$ plant agents. To regulate the system, a control agent is placed and assigned to a target as shown.

6.1 Queue Structure

Consider a system with a queue plant structure, where n plant agents are arranged in a line with the neighbouring nodes connected to each other. Figure 6.1 shows a queue plant structure with $n = 5$ plant agents. It is easy to see that a queue plant structure can be regulated with just one control agent in the following configuration: a control agent is linked to a plant agent on one end of the queue while targeting the plant agent on the other end of the queue, as in Figure 6.1. Thus, the settling time for a queue is

$$\lambda = \Delta(X_{n+1}, T_{n+1}) = (n + 1)\delta.$$

Theorem 4 implies that any other arrangement of a single control agent and its target will result in failure of growing. Specifically, linking the control agent to any plant agent other than one of the two ends and/or targeting such a plant agent fails to satisfy condition (a) of Theorem 4. Failure of targeting, however, is not an issue for this structure, as any control agent, considered by itself, can zero its target under Assumption A_6. It should be noted that for this structure, the proposed control scheme will work for any value of δ_i and δ_{ji}, due to the simplicity of a queue structure.

As an example, consider the system in Figure 6.1. The implemented control scheme results in successful targeting as mentioned before. Applying the Growing Analysis Algorithm to this plant gives:

$$\begin{aligned}
\Omega &= \{O_5\}, \\
\Omega &= \{O_5, O_4\}, \\
\Omega &= \{O_5, O_4, O_3\}, \\
\Omega &= \{O_5, O_4, O_3, O_2\}, \\
\Omega &= \{O_5, O_4, O_3, O_2, O_1\}.
\end{aligned}$$

Hence, growing succeeds.

6.2 Grid Structure

We consider two variations of a grid plant structure in this section. First, we consider a $p \times q$ grid structure, where the $n = pq$ plant agents are arranged in a grid where adjacent nodes are linked to each other. Figure 6.2(a) shows a 3×2 grid system. Using Theorems 3 and 4, we can quickly discover that there are, in fact, not many ways in which targets and control agents can be arranged so as to be consistent with the necessary conditions. One such arrangement is to link the m control agents to plant agents along one of the shorter edges, and assign plant agents on the opposite edge as targets, as shown in Figure 6.2(a). Indeed, this control scheme fulfils the control objective with the minimum number of control agents needed, that is $m = \min\{p, q\}$, with settling time

$$\lambda = (\max\{p, q\} + 1)\delta.$$

If fewer than $\min\{p, q\}$ control agents are used, then growing fails since it is not possible to have all the plant agents on fastest paths between control agents and their respective targets as required by Theorem 4(a).

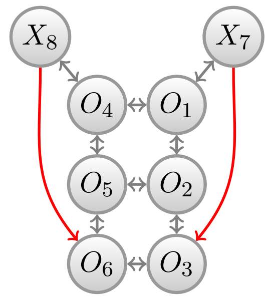
For example, consider the grid system in Figure 6.2(a). Assume that $\delta = 1$. The arrangement of control agents and targets leads to successful targeting as there are no loops in the dependency graph shown in Figure 6.2(b). Another way to justify this is to look at the propagation times between the control agents and the targets. Under the assumption that $\delta_i = \delta_{ji} = \delta$, it is clear that this arrangement results in a smaller propagation time from each control agent to its target than the propagation time from each control agent to any other target, implying that Theorem 1(c) holds; i.e., targeting works. Applying the Growing Analysis Algorithm to the plant, we can see that all nodes get zeroed:

$$\begin{aligned}
\Omega &= \{O_3, O_6\}, \\
\Omega &= \{O_3, O_6, O_5\}, \\
\Omega &= \{O_3, O_6, O_5, O_2\}, \\
\Omega &= \{O_3, O_6, O_5, O_2, O_4\}, \\
\Omega &= \{O_3, O_6, O_5, O_2, O_4, O_1\},
\end{aligned}$$

showing that growing works.

Next, we consider a variation of the $p \times q$ grid system, where diagonal bidirectional links are included among plant agents in addition to the links considered previously. We call this structure a "full" grid. Figure 6.3(a) shows an example of a 3×3 full grid system. Theorems 3 and 4 can be used again to eliminate a great number of possible arrangements of control agents and targets that result in failure of targeting and/or growing. For instance,

(a)



(b)

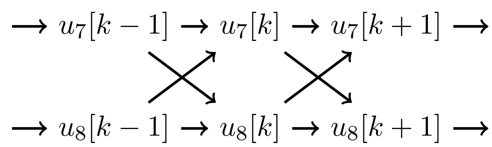


Figure 6.2: Example of a grid plant structure with: (a) $n = 3 \times 2$ plant agents where adjacent nodes are linked to each other. To regulate this system, control agents are introduced as shown. (b) the corresponding dependency graph with no loops; i.e., targeting works.

the control scheme used for the non-full grid structure does not work for a full grid as it fails to satisfy Theorem 4(b). Examining a $p \times q$ full grid structure carefully, we deduce that a possible control scheme for regulation is to place the control agents along two edges targeting the plant agents on the remaining edges in a diagonal manner, as in Figure 6.3(a). This implies that the minimum number of control agents needed is $m = p + q - 1$. Clearly, this deduction is subject to $p \geq 2$ and $q \geq 2$, since having either one of p or q as one results in a queue structure. The proposed control scheme can, in fact, regulate the system with settling time

$$\lambda = (\min\{p, q\} + 1)\delta.$$

Note that in a full grid structure, the minimum number of unique fastest paths that include all the plant agents is $p + q - 1$ with each fastest path along the diagonal connections confirming its uniqueness as required by Theorem 4(b). Having the fastest paths arranged in any other way — consistent with Theorems 3 and 4 — requires more than $p + q - 1$ control agents for successful targeting and growing.

As an example, consider the plant in Figure 6.3(a) with $\delta = 1$. The control scheme leads to successful targeting since there are no loops in the dependency graph shown in

Figure 6.3(b). Applying the Growing Analysis Algorithm to the plant gives:

$$\begin{aligned}
\Omega &= \{O_3, O_6, O_7, O_8, O_9\}, \\
\Omega &= \{O_3, O_6, O_7, O_8, O_9, O_5\}, \\
\Omega &= \{O_3, O_6, O_7, O_8, O_9, O_5, O_2\}, \\
\Omega &= \{O_3, O_6, O_7, O_8, O_9, O_5, O_2, O_4\}, \\
\Omega &= \{O_3, O_6, O_7, O_8, O_9, O_5, O_2, O_4, O_1\},
\end{aligned}$$

showing that growing succeeds.

6.3 Spider Structure

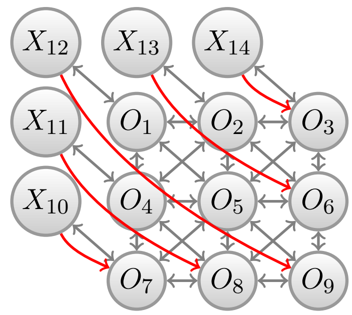
Consider a spider plant structure that has one central node and $p \geq 2$ branches of any length, where the length of branch i, denoted l_i, is the number of plant agents on that branch. An example of a spider system with $p = 4$ is shown in Figure 6.4(a). Again using Theorems 3 and 4, we can conclude that targeting and growing succeed only if at least $p - 1$ control agents are used. Hence, the minimum number of control agents needed for regulation is $m = p - 1$. In fact, a possible scheme that achieves regulation is to link the control agents to the end nodes of any $p-1$ of the p branches, with all but one targeting the branch nodes adjacent to the central node, and with the remaining control agent targeting the node at the end of the branch that is without a control agent, as in Figure 6.4(a). For this structure, the propagation time through the plant agents on the longest branch is $\max_{1 \leq i \leq p} l_i \delta$; thus the settling time is found to be

$$\lambda = \left(\max_{1 \leq i \leq p} l_i + 1 \right) \delta.$$

Using fewer than $p - 1$ control agents, we cannot satisfy the conditions of both Theorems 3 and 4 at the same time. For instance, fewer than $p - 1$ control agents can include all the plant agents on their fastest paths as required by Theorem 4(a) only if they share the central node, which contradicts Theorem 3(b).

As an example, consider the plant in Figure 6.4(a). Assume that $\delta = 1$. The dependency graph for this system has no loops as shown in Figure 6.4(b), showing that targeting works. Next, the Growing Analysis Algorithm is applied to the system which results in zeroing all

(a)



(b)

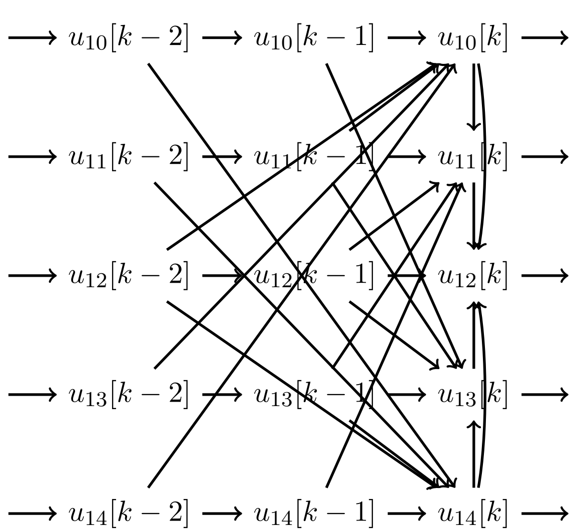


Figure 6.3: Example of a full grid plant structure with: (a) $n = 3 \times 3$ plant agents where additional diagonal edges are included between plant agents. To regulate this system, control agents are introduced as shown. (b) the corresponding dependency graph with no loops; i.e., targeting works. To avoid cluttering the graph only one set of dependencies between each pair of control signals are drawn.

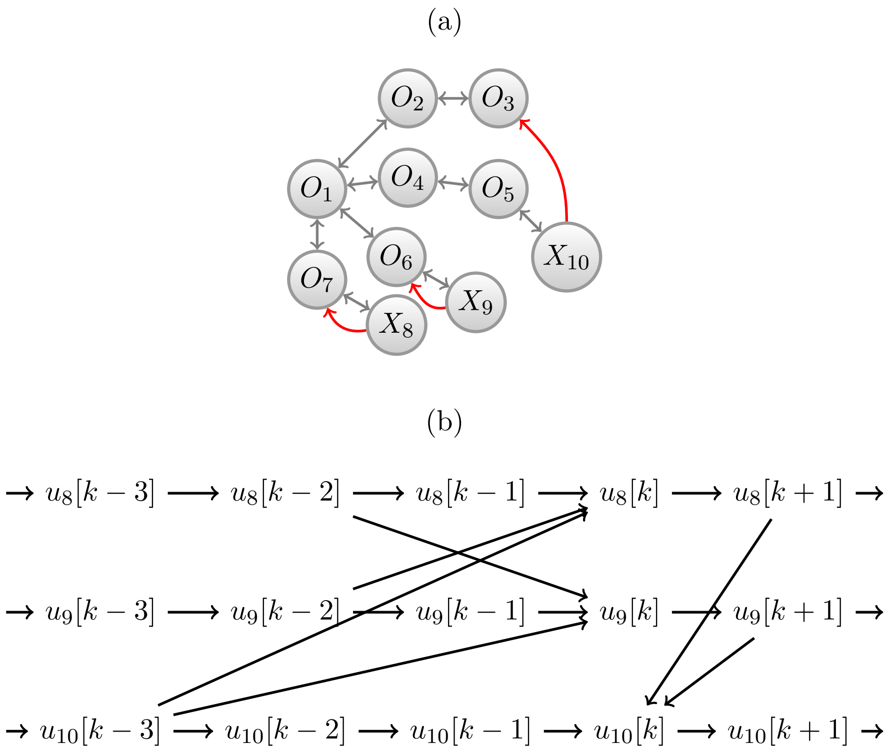


Figure 6.4: Example of spider plant structure with $p = 4$ branches: (a) the plant with the implemented control scheme, (b) the corresponding dependency graph with no loops, showing that targeting works. To avoid cluttering the graph only one set of dependencies between each pair of control signals are drawn.

the plant agents:

$$\begin{aligned}
\Omega &= \{O_3, O_6, O_7\}, \\
\Omega &= \{O_3, O_6, O_7, O_2\}, \\
\Omega &= \{O_3, O_6, O_7, O_2, O_1\}, \\
\Omega &= \{O_3, O_6, O_7, O_2, O_1, O_4\}, \\
\Omega &= \{O_3, O_6, O_7, O_2, O_1, O_4, O_5\}.
\end{aligned}$$

Thus, growing succeeds.

6.4 Ring Structure

Consider a ring plant structure, where the $n \geq 3$ plant agents are connected to each other in the shape of a ring. Figure 6.5(a) shows a ring plant structure with $n = 6$ plant agents. It is easy to see that a ring plant structure can be regulated using $m = 2$ control agents linked to two adjacent plant agents while targeting, in case of an even number of plant agents, the two furthest adjacent plant agents as in Figure 6.5(a). For an odd number of plant agents, one control agent targets the furthest plant agent, while the other control agent targets the closer of one of the two plant agents adjacent to the first target as shown in Figure 6.6(a) for a system with $n = 7$ plant agents. The settling time for a ring structure is

$$\lambda = (\lceil \frac{n}{2} \rceil + 1)\delta.$$

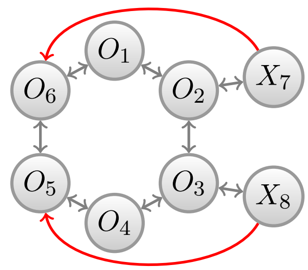
These control schemes are the only possible arrangement of two control agents and their targets which are consistent with the necessary conditions of Theorems 3 and 4. Using Theorem 4, we can deduce that it is impossible to have both targeting and growing successful for a ring plant structure with only one control agent, as one fastest path can include no more than $\lceil \frac{n}{2} \rceil$ of the plant agents.

To demonstrate that targeting and growing both work for the proposed control scheme, first consider the plant in Figure 6.5(a) with $\delta = 1$. Targeting works for the implemented control scheme as there are no loops in the dependency graph shown in Figure 6.5(b). All the plant agents will get zeroed after the Growing Analysis Algorithm is applied to the plant, showing that growing succeeds:

$$\begin{aligned}
\Omega &= \{O_5, O_6\}, \\
\Omega &= \{O_5, O_6, O_1\}, \\
\Omega &= \{O_5, O_6, O_1, O_4\}, \\
\Omega &= \{O_5, O_6, O_1, O_4, O_2\}, \\
\Omega &= \{O_5, O_6, O_1, O_4, O_2, O_3\}.
\end{aligned}$$

Next, we consider the example in Figure 6.6(a) for which targeting works as there are no loops in the dependency graph as in Figure 6.6(b) where δ is set to 1. Similarly growing works for this plant as all plant agents are included in Ω once the Growing Analysis

(a)



(b)

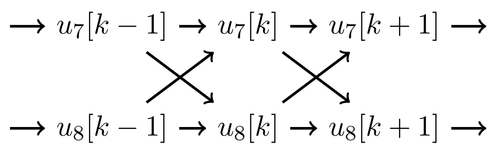


Figure 6.5: Example of a ring plant structure with $n = 6$ plant agents: (a) the plant with two control agents introduced as shown in the figure, (b) the corresponding dependency graph with no loops, showing that targeting works.

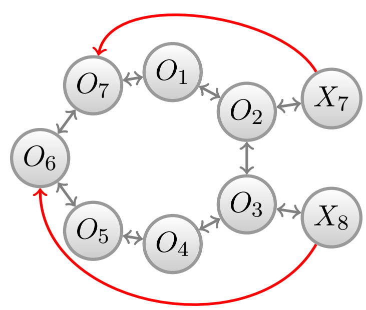
Algorithm is terminated:

$$\begin{aligned}
\Omega &= \{O_6, O_7\}, \\
\Omega &= \{O_6, O_7, O_1\}, \\
\Omega &= \{O_6, O_7, O_1, O_5\}, \\
\Omega &= \{O_6, O_7, O_1, O_5, O_2\}, \\
\Omega &= \{O_6, O_7, O_1, O_5, O_2, O_4\}, \\
\Omega &= \{O_6, O_7, O_1, O_5, O_2, O_4, O_3\}.
\end{aligned}$$

6.5 Wheel Structure

A wheel plant structure with $n \geq 4$ plant agents has $n-1$ plant agents connected in a ring and the remaining plant agent connected to all the plant agents in the ring. An example of a wheel plant structure with $n = 10$ is shown in Figure 6.7(a). Again, using Theorems 3

(a)



(b)

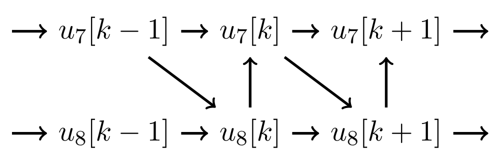


Figure 6.6: Example of a ring plant structure with $n = 7$ plant agents: (a) the controlled plant with two control agents introduced as shown in the figure, (b) the corresponding dependency graph with no loops, showing that targeting works.

and 4, we can deduce that successful targeting and growing for a wheel structure with $n \geq 7$ is achieved with at least $m = \lfloor \frac{n}{2} \rfloor$ control agents, with three control agents linked to three adjacent plant agents on the ring, where the control agent in the middle targets the furthest plant agent on the ring, while the other two control agents target their neighbouring plant agents. The remaining control agents are linked to every second plant agent targeting their neighbouring agent, unless the number of plant agents on the ring is odd which results in one control agent targeting the plant agent it is linked to as shown in Figure 6.7(a). The settling time for a wheel plant structure with $n \geq 7$ is

$$\lambda = 4\delta.$$

Including all the plant agents on fastest paths using fewer than $\lfloor \frac{n}{2} \rfloor$ control agents always leads to non-unique fastest paths, violating Theorem 4(b). To better clarify this point, consider the example shown in Figure 6.7(a). Assume X_{15} is removed and X_{14} targets O_8 rather than O_7. Obviously, this violates Theorem 4(b). Also, if the plant agent in the center is taken as a target, more control agents will be needed for successful targeting and growing.

Figure 6.7 shows the graph structure of a plant with $n = 10$ plant agents arranged in a wheel topology, and its dependency graph, assuming $\delta = 1$, which does not have any loops; i.e. targeting works for this system. Applying the Growing Analysis Algorithm results in:

$$\begin{aligned}
\Omega &= \{O_3, O_7, O_8, O_9, O_{10}\}, \\
\Omega &= \{O_3, O_7, O_8, O_9, O_{10}, O_1\}, \\
\Omega &= \{O_3, O_7, O_8, O_9, O_{10}, O_1, O_2\}, \\
\Omega &= \{O_3, O_7, O_8, O_9, O_{10}, O_1, O_2, O_4\}, \\
\Omega &= \{O_3, O_7, O_8, O_9, O_{10}, O_1, O_2, O_4, O_5\}, \\
\Omega &= \{O_3, O_7, O_8, O_9, O_{10}, O_1, O_2, O_4, O_5, O_6\}.
\end{aligned}$$

Thus, growing succeeds.

For a wheel structure with $4 \leq n \leq 6$, a different control scheme has to be used with $m = 3$ control agents to be consistent with the necessary conditions of Theorems 3 and 4. In this control scheme that successfully regulates the system, the control agent in the middle targets the node in the center. The settling time is found to be

$$\lambda = 3\delta.$$

Theorem 4 implies that growing fails for this structure with $m = 1$ or $m = 2$ control agents, since it is impossible to include all the plant agents on just one or two fastest paths that are each unique.

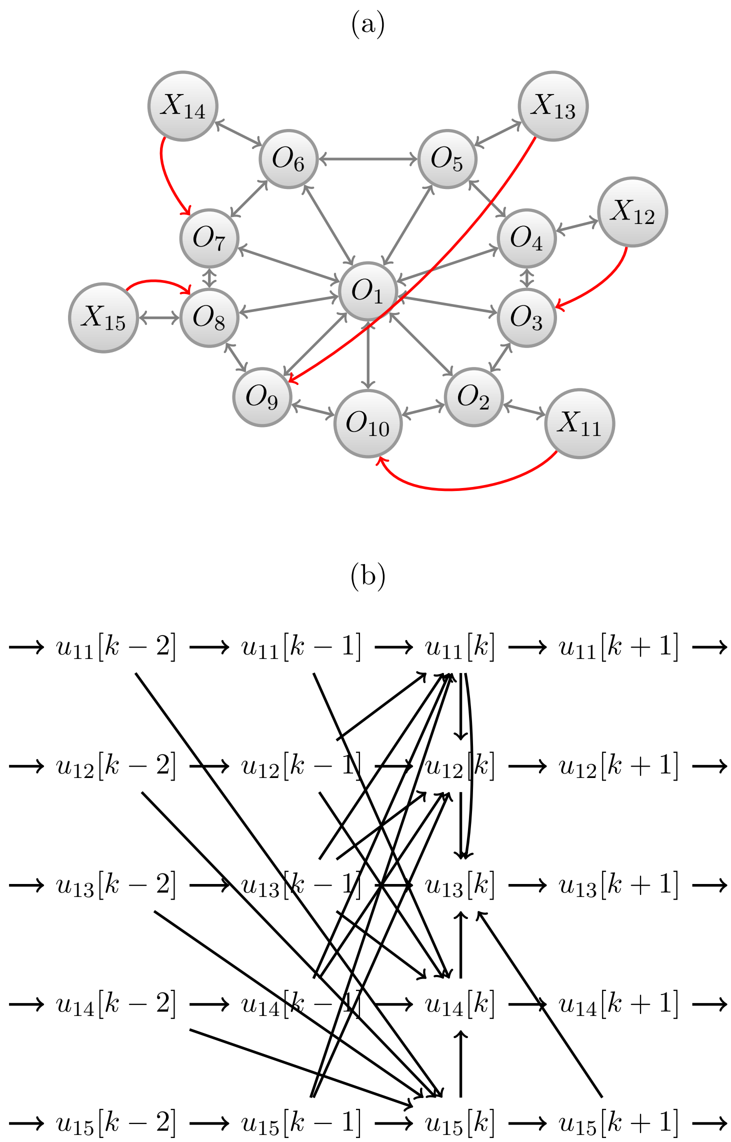


Figure 6.7: Example of a wheel plant structure with $n = 10$ plant agents: (a) the plant with the implemented control scheme, (b) the corresponding dependency graph with no loops, showing that targeting works. To avoid cluttering the graph only one set of dependencies between each pair of control signals are drawn.

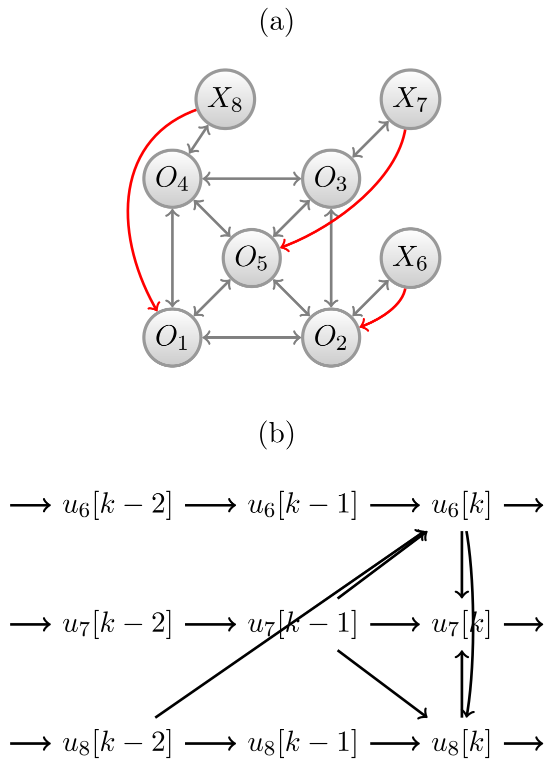


Figure 6.8: Examples of a wheel plant structure with $n = 5$ plant agents: (a) the plant with the implemented control scheme, (b) the corresponding dependency graph with no loops, showing that targeting works. To avoid cluttering the graph only one set of dependencies between each pair of control signals are drawn.

As an example, consider a wheel structured system with $n = 5$ plant agents as shown in Figure 6.8(a). Assume that $\delta = 1$. Targeting works for this system as there are no loops in the corresponding dependency graph shown in Figure 6.8(b). Growing also succeeds since all the plant agents get zeroed through the Growing Analysis Algorithm:

$$\begin{aligned}
\Omega &= \{O_1, O_2, O_5\}, \\
\Omega &= \{O_1, O_2, O_5, O_3\}, \\
\Omega &= \{O_1, O_2, O_5, O_3, O_4\}.
\end{aligned}$$

6.6 Complete Graph Structure

Consider a plant that has a complete graph structure with $n \geq 2$ plant agents where each plant agent is connected to all the other plant agents. An example of a complete graph plant structure with $n = 6$ is shown in Figure 6.9(a). As before, Theorems 3 and 4 provide us with guidelines as to which arrangements certainly result in targeting and/or growing failure, which makes it easier to find a potentially successful control scheme. For this structure, the Growing Analysis Algorithm also helps in deducing the minimum number of control agents needed for the system's regulation as each plant agent is linked to all other plant agents. Successful targeting and growing can be achieved for this example only if at least $m = n - 1$ control agents are used where each control agent is linked to one of the plant agents, with all but one targeting the same plant agent they are linked to, and with the remaining control agent targeting the only node that is without a control agent, as in Figure 6.9(a). The settling time for this structure is

$$\lambda = 3\delta.$$

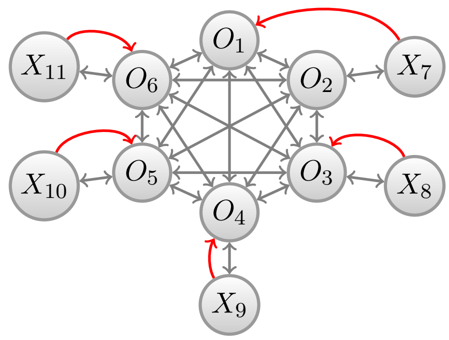
Targeting and/or growing fails for a complete graph structure with fewer than $n-1$ control agents as with this number of control agents it is impossible to have all the plant agents on fastest paths between control agents and their respective targets such that, on average, the propagation times along the paths from the control agents to their respective targets would be less than the propagation times along the paths from the control agents to all other targets (i.e., the conditions of Theorems 3 and 4 will not hold). For instance, if a control scheme similar to the scheme for a wheel structure is used in this case, we fail to satisfy condition (a) of Theorem 3. Another way to justify the number of required control agents is through the Growing Analysis Algorithm. Given that each plant agent is connected to all the other plant agents, we need to target at least $n - 1$ plant agents; otherwise growing fails right away.

Consider the example in Figure 6.9(a) and assume that $\delta = 1$. Constructing the dependency graph makes it clear that targeting succeeds (refer to Figure 6.9(b)). Applying the Growing Analysis Algorithm, it is easy to see that growing works:

$$\begin{aligned} \Omega &= \{O_1, O_3, O_4, O_5, O_6\}, \\ \Omega &= \{O_1, O_3, O_4, O_5, O_6, O_2\}. \end{aligned}$$

Note that the complete graph structure is equivalent to a queue structure when $n = 2$, a ring structure when $n = 3$, and a full grid structure (i.e., a grid structure with diagonal links) when $n = 4$. The given control scheme for the complete graph structure confirms

(a)



(b)

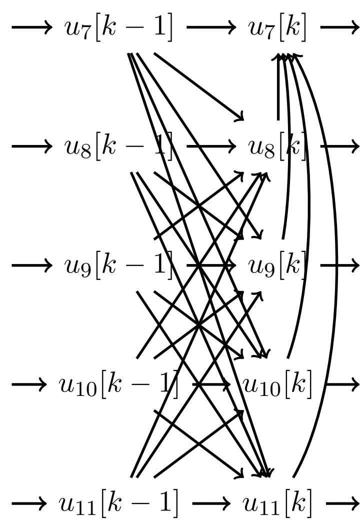


Figure 6.9: Example of a complete graph plant structure with $n = 6$ plant agents: (a) the plant with the implemented control scheme that regulates the system, (b) the corresponding dependency graph with no loops, showing that targeting works. To avoid cluttering the graph only one set of dependencies between each pair of control signals are drawn.

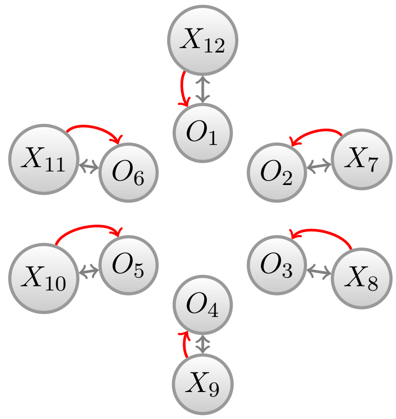


Figure 6.10: Example of a null graph plant structure with $n = 6$ plant agents. To regulate this system, six control agents are introduced as shown.

our earlier deductions on the required number of control agents for these structures. In fact, the control scheme for a complete graph with $n = 2$, $n = 3$ and $n = 4$ plant agents results in the same arrangement as the control schemes for a queue, a ring and a full grid do, respectively.

6.7 Null Graph Structure

Consider a null graph plant structure with n plant agents where the plant agents are not connected to each other; i.e., the structure is edgeless. An example of a null graph plant structure with $n = 6$ is shown in Figure 6.10. Using Theorem 4(a), one can easily deduce that regulation can be achieved only if at least $m = n$ control agents are used with each control agent linked to a plant agent and targeting that same agent as in Figure 6.10. The settling time for this structure is

$$\lambda = \delta_i + \delta_{ji} = 2\delta.$$

Note that this control scheme works for any plant with a null graph structure with any values of δ_i and δ_{ji}.

As an example, consider the system in Figure 6.10. Targeting clearly works using the proposed control scheme. Regulation is achieved once all targets are zeroed.

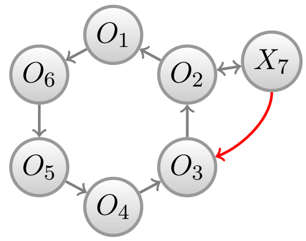


Figure 6.11: Example of a ring plant structure with $n = 6$ plant agents where all links are unidirectional and all oriented in the same way. For successful targeting and growing, one control agent is used as shown in the figure.

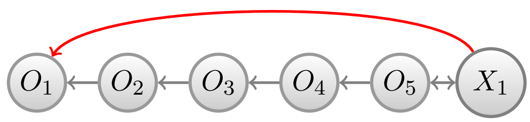


Figure 6.12: Example of a queue plant structure with unidirectional links all arranged in one direction. To regulate this system, a control agent is placed and assigned to a target as shown, such that $\Delta(X_{n+1}, T_{n+1}) < \infty$ (as required by Assumption A_1).

6.8 Variations of Previous Examples

In the above structures, all the links were assumed to be bidirectional for simplicity. Having unidirectional links in these structures sometimes simplifies and sometimes complicates the control problem. In the following, we consider three examples where assuming unidirectional links results in a smaller, the same, or a larger number of control agents needed for the regulation of a specific structure, as compared to the case where only bidirectional links were allowed.

As the first example, consider the ring structure. If the plant links are unidirectional and all oriented the same way, we can easily see that successful targeting and growing can be achieved using only one control agent, i.e., $m = 1$ as in Figure 6.11, which is consistent with Theorems 3 and 4. As the second example, consider a queue structure with unidirectional links all in the same direction. As in the bidirectional case, only one control agent is needed for regulation. When the links are unidirectional, however, the control agent can be placed at only one end of the queue as in Figure 6.12. As the final example, consider a spider plant structure with unidirectional links as shown in Figure 6.13. Again, using Theorems 3 and 4, we can conclude that successful targeting and growing can be achieved only if at least $m = p$ control agents are used in a specific arrangement as in Figure 6.13 (where p is the number of branches).

The other assumption made for all the previous examples is that the δ_i and δ_{ji} values

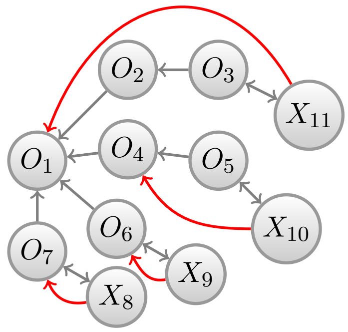


Figure 6.13: Example of a spider plant structure with $p = 4$ branches and unidirectional links as shown. For regulation of this plant, four control agents are needed as in the figure.

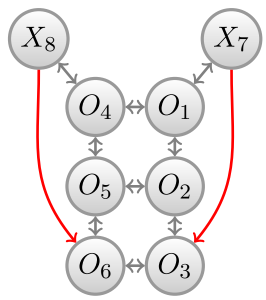


Figure 6.14: Example of a grid plant structure with $n = 3 \times 2$ plant agents where all $\delta_i = \delta_{ji} = 1$. To regulate this system, control agents are introduced as shown.

are all equal to some $\delta \geq 1$. If the propagation times between the plant agents and the control agents are different, we can still use the same idea to help design a proper control scheme. Clearly, the design problem will be more challenging. For instance, consider the example in Figure 6.14. In Section 6.2, it was shown that two control agents can achieve successful targeting and growing for this system, assuming that all $\delta_i = \delta_{ji} = 1$. As a variation of this example, assume that $\delta_{56} = \delta_{65} = 3$ while all other δ_i and δ_{ji} values are one. The previous control scheme cannot achieve successful targeting and growing as it violates condition (b) of Theorem 4; however, a modified control scheme in which control agent X_8 targets O_5 and a new control agent X_9 is linked to and targets O_6, satisfies the conditions of Theorem 3 and 4. This control scheme, indeed, results in successful targeting and growing.

6.9 Final Remarks

It should be noted that regulation of a plant with no interaction between its plant agents (i.e., null graph structure) and/or full interaction between them (i.e., the complete graph structure) requires the most number of control agents. Hence, having full interaction or lack of interaction are both considered as "challenging" plant structures from the control point of view. On the other hand, the queue and the ring structures are considered as the "easiest" plant structures as they can be regulated with the least number of control agents (i.e., $m = 1$ and $m = 2$, respectively). This observation suggests that it is possible to classify the graph structures based on the difficulty level of the control problem.

In the next chapter, a summary of the main results of this thesis is given. Also possible promising future directions are provided.

Chapter 7

Summary and Future Directions

This thesis, motivated by research in which a crowd of people is controlled by a number of control agents, focused on decentralized output regulation of nonlinear discrete-time multi-agent systems. In particular, our goal was to achieve deadbeat regulation of the outputs. For this purpose, we first provided a generalized system model along with various assumptions concerning targeting assignment, communication and sensing requirements, signal propagation time in the system, and targeting and growing processes. To achieve regulation, control agents were introduced at strategic locations among the plant agents, with each control agent trying to regulate a specific plant agent, called its target. Targeting analysis was used to determine if the control agents are capable of driving the outputs of all target plant agents to zero. Then, growing analysis was applied to determine if the same control laws can drive the outputs of non-target plant agents to zero. Two control problems were investigated in this thesis: (1) for a given plant and control scheme, find, if possible, a set of computable control laws that regulate the plant, (2) for a given plant, determine the number of control agents and a control scheme that can achieve regulation.

To deal with Problem 1, our approach is to ensure targeting and growing analyses both succeed. To verify whether or not targeting works, we used the notion of dependency graphs for the control signals. Moreover, we derived a set of necessary and sufficient algebraic conditions to determine when targeting succeeds. For growing, we used the Growing Analysis Algorithm to see if all plant agents get zeroed and regulation is achieved. To address Problem 2, a set of easily-verifiable necessary conditions were presented for targeting and/or growing to succeed. These conditions help greatly with the design problem; i.e., where to place the control agents and how to assign targets. To better show how these necessary conditions help a designer come up with a control scheme with the minimum number of control agents required, various graph structures were studied. We concluded that the plant topology is an important factor affecting the difficulty of the control problem.

The main strength of this work is the set of new easily-verifiable necessary conditions developed for successful targeting and/or growing. These conditions are very powerful as they have simple geometric interpretations, and are easily scalable to large systems. They emphasize on the graph structure and the signal propagation time through the plant agents and control agents as two important factors involved in regulation of a plant.

There are a number of different directions in which this research can go forward. In the following, we have listed down some of those ideas that are considered most promising or are of particular personal interest:

(a) Loosen up deadbeat requirement of regulation. Also, it is interesting to explore full state regulation.

(b) Extend our control approach to tracking and disturbance rejection as they arise in a large number of applications.

(c) Understand better any connections between our work and the research done within: (i) the multi-agent control area, specifically "multi-agent controllability" [16, 13], and (ii) structured systems theory, particularly the idea of "invertibility" [10, 1].

(d) Determine if it is ever possible to regulate a multi-agent system with control laws that violate Definition 2, i.e., they are not computable.

(e) Investigate communication and sensing requirements in more detail to help determine how restrictions on communications among control agents, and sensing limitations are related. For instance, it might be possible to achieve regulation even when control agents have no sensing abilities as long as there are no constraints on the communications between control agents.

(f) Introduce constraints and delays on communication and sensing. In practice, it is often not feasible to have full communications among control agents, and/or full sensing of the plant.

(g) Show that the converse of Theorem 4 holds, i.e., assuming targeting works for a given plant and control scheme, then growing succeeds if both the following hold: (a) each plant agent lies on the fastest path from some control agent to its associated target, (b) for each control agent X_i, the fastest path from X_i to T_i is unique. This conjecture is very appealing as it simplifies the control problem by giving us the option to ignore growing analysis.

(h) Find a mathematical representation to capture the growing algorithm. We have considered different approaches (e.g., finite state machines, the use of graph theory, a dependency graph for growing of the plant agents, etc.), but have not been able to develop a useful mathematical tool yet. Looking into "power dominating set" problems might be helpful for this purpose [11]. Having a mathematical representation of growing can help understand growing process better, and find more easily-verifiable geometric conditions necessary for regulation.

(i) Develop a "divide and conquer" approach to help design control schemes for very large systems.

(j) Address the critical issue of robustness. It is important to make our control approach robust in the sense of allowing for plant uncertainty, and also deal with faulty control agents and/or communication links.

(k) Consider time-varying links between plant agents. Depending on the applications, links between plant agent might appear and disappear. Thus, it is interesting to investigate this type of problem.

(l) Discover fundamental performance limitations and how they vary with plant topology. This idea has been studied briefly in Chapter 6 for only a limited number of plant topologies. Studying other plant topologies can provide us with more insight on the issue of performance limitations.

(m) Investigate optimal control problems. Two possible classes of optimal control problems are: (i) determine optimal placement of control agents and targets (e.g., to minimize the number of control agents, settling time, or communication requirements), and (ii) find optimal control signals for each control agent.

(n) Generalize our work to vector-valued outputs. Also, it may be valuable to extend our approach to a continuous-time framework.

Bibliography

[1] M. H. Amin and M. M. Hassan. Time-optimal output regulators for linear multivariable discrete-time systems; part 2. all classes of right-invertible systems. *International Journal of Control*, 46:1411–1428, 1987. 63

[2] M. Aoki. On feedback stabilizability of decentralized dynamic systems. *Automatica*, pages 163–173, 1972. 3

[3] K. Bergey, Kevin Spieser, and D. E. Davison. The psychological dynamics of students in a classroom: Modeling and control strategies based on suggestibility theory. In *Conference on Control Applications*, Singapore, October 2007. IEEE. 2, 8

[4] J. Corfmat and A. Morse. Decentralized control of linear multivariable systems. *Automatica*, 12:479–495, 1976. 3

[5] Daniel E. Davison and Kevin Spieser. Stabilization of suggestible crowds. In Pierre T. Kabamba and Jingshan Li, editors, *Advances in Systems Theory: Control, Communications, Manufacturing, and Rational Behavior*, pages 75–151. Wingspan Press, 2009. 2, 7, 8, 16, 18, 22

[6] J. Dion, C. Commault, and J. van der Woude. Generic properties and control of linear structured systems: a survey. *Automatica*, 39(7):1125–1144, 2003. 3

[7] J. Shamma (editor). Cooperative control of distributed multi-agent systems. 2007. 3

[8] J. Feddema, C. Lewis, and D. Schoenwald. Decentralized control of cooperative robotic vehicles: Theory and application. *IEEE Transactions on Robotics and Automation*, 18(5):852–864, 2002. 3

[9] T. Gustavi, D. Dimarogonas, M. Egerstedt, and X. Hu. Sufficient conditions for connectivity maintenance and rendezvous in leader-follower networks. *Automatica*, 46(1):133–139, 2010. 3

[10] M. M. Hassan and M. H. Amin. Time-optimal output regulators for linear multivariable discrete-time systems; part 1. right invertible decouplable systems. *International Journal of Control*, 46:865–879, 1987. 63

[11] T.W. Haynes, S.M. Hedetniemi, S.T. Hedetniemi, and M.A. Hennin. Power domination in graphs applied to electrical power networks. *SIAM Journal on Discrete Mathematics*, 15:519–529, 2002. 64

[12] D. Liu, L. Wang, and K.C. Tan (editors). Design and control of intelligent robotic systems. 2009. 3

[13] S. Martini, M. Egerstedt, and A. Bicchi. Controllability analysis of multi-agent systems using relaxed-equitable partitions. *International Journal of Systems, Control, and Communications*, 2(1-3), 2010. 3, 63

[14] R. Olfati-Saber, J. Fax, and R. Murray. Consensus and cooperation in networked multiagent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007. 3

[15] Z. Qu. *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. Springer, 2009. 3

[16] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimatization*, 48(1):162–186, 2009. 3, 63

[17] W. Ren and R. Beard. Distributed consensus in multi-vehicle cooperative control: Theory and applications. 2008. 3

[18] Nasim A. Shams and Daniel E. Davison. A graph-theoretic condition necessary for the stabilization of crowds using cooperative control agents. In *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, Calgary, Alberta, May 2010. 2, 8, 18, 22, 35

[19] D.D. Siljak. *Decentralized Control of Complex Systems*. Academic Press, 1991. 3

[20] K. Spieser and D. E. Davison. Multi-agent stabilization of the psychological dynamics of one-dimensional crowds. *Automatica*, 45, 2009. 2, 8

[21] Kevin Spieser. Stabilizing the psychological dynamics of people in a crowd. M.A.Sc. dissertation, University of Waterloo, 2008. 2, 8

[22] Kevin Spieser and D. E. Davison. Stabilizing the psychological dynamics of people in a queue. In *Proceedings of the American Control Conference*, Seattle, WA, June 2008. 2, 8

[23] Kevin Spieser and D. E. Davison. A cooperative multi-agent approach for stabilizing the psychological dynamics of a two-dimensional crowd. In *Proceedings of the American Control Conference*, Seattle, WA, June 2009. AACC. 2, 8

[24] S. Sundaram and C. N. Hadjicostis. Distributed function calculation via linear iterations in the presence of malicious agentsparts i and ii. In *American Control Conference*, Seattle, WA, June 2008. 3

[25] S. Wang and E.J. Davison. On the stabilization of decentralized control systems. *IEEE Transactions on Automatic Control*, AC-18:473–478, 1973. 3