2018

Multi-agent persistent monitoring of a finite set of targets

BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

MULTI-AGENT PERSISTENT MONITORING OF A FINITE SET OF TARGETS

by

XI YU

B.Sc., Karlsruhe Institute of Technology, 2010
Dipl.-Ing., Karlsruhe Institute of Technology, 2011

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2018

Approved by

First Reader

Sean B. Andersson, PhD
Associate Professor of Mechanical Engineering
Associate Professor of Systems Engineering

Second Reader

Christos G. Cassandras, PhD
Distinguished Professor of Engineering
Professor and Division Head of Systems Engineering
Professor of Electrical and Computer Engineering

Third Reader

John B. Baillieul, PhD
Distinguished Professor of Mechanical Engineering
Distinguished Professor of Systems Engineering
Distinguished Professor of Electrical and Computer Engineering

Fourth Reader

Roberto Tron, PhD
Assistant Professor of Mechanical Engineering
Assistant Professor of Systems Engineering

...obwohl es wohl nie zu einer praktischen Anwendung kommen wird.
(I do not think that the radio waves I have discovered will have any practical application.)

Heinrich Rudolf Hertz

Acknowledgments

I cannot express enough thanks to my supervisor, Dr. Sean Andersson, who guided me to the research path, trained me to be a researcher and a teacher, and also shaped me into an optimistic and confident person. I feel extremely lucky to have met Sean, who has always been the strongest supporter when I encounter difficulties, the most patient and reliable supervisor to my academic performance, and a mentor in my life.

My very special gratitude goes to my committee, Dr. Christos Cassandras, Dr. John Baillieul, and Dr. Roberto Tron, for their time to review my work and all their comments and suggestions to improve it. I would also like to thank my defense chair, Dr. Hua Wang, for taking the time to help me organize the defense.

My completion of this work could not have been accomplished without the support of my classmates, colleagues, and collaborators. Peng, Trevor, Brian, and Catherine, the time with you was the most beautiful beginning of my PhD life. Yufan, Brett, Sean, Sheila, Yuhe, Jessie, and Spencer, I will give my best wishes to you while leaving this lovely family. My gratitude to Alyssa, Cristi, Kayhan, Kevin, Bowen, Xinmiao and Sepideh, who graduated before me and your success encouraged me to keep going. Thank you, Nan, for the works we collaborated. Thank you, my friends in robotics lab, for the food and laughter we shared.

I am also grateful to the ME department staff and CISE staff, who always provided me with help and warm friendship. I would like to thank the ISSO staff, who helped me to stay in United States to complete my degree, and Brendan from Mugar Library, who provided the last round of review to my thesis.

To Minjie: Thank you for always having faith in me. To Hao: Thank you for encouraging me to pursue a PhD in US. To Jingwen, Jing, Mujun, and Kuo: Thank you for being there whenever I need you.

Finally, to my parents, my grand parents, and the extended family: Your encouragement, support, and everlasting love are essential to every tiny piece of my achievement. To Mo, who accompanied me while most of the results in this thesis were derived: My heartfelt thanks.

Xi Yu

December, 2017

MULTI-AGENT PERSISTENT MONITORING OF A FINITE SET OF TARGETS

XI YU

Boston University, College of Engineering, 2018

Major Professor: Sean B. Andersson, PhD
Associate Professor of Mechanical Engineering
Associate Professor of Systems Engineering

ABSTRACT

The general problem of multi-agent persistent monitoring finds applications in a variety of domains ranging from meter to kilometer-scale systems, such as surveillance or environmental monitoring, down to nano-scale systems such as tracking biological macromolecules for studying basic biology and disease. The problem can be cast as moving the agents between targets, acquiring information from or in some fashion controlling the states of the targets. Under this formulation, at least two questions need to be addressed. The first is the design of motion trajectories for the agents as they move among the spatially distributed targets and jointly optimize a given cost function that describes some desired application. The second is the design of the controller that an agent will use at a target to steer the target's state as desired.

The first question can be viewed in at least two ways: first, as an optimal control problem with the domain of the targets described as a continuous space, and second as a discrete scheduling task. In this work we focus on the second approach, which formulates the target dynamics as a hybrid automaton, and the geometry of the targets as a graph. We show how to find solutions by translating the scheduling problem

into a search for the optimal route. With a route specifying the visiting sequence in place, we derive the optimal time the agent spends at each target analytically.

The second question, namely that of steering the target's state, can be formulated from the perspective of the target, rather than the agent. The mobile nature of the agents leads to intermittent control, such that the controller is assumed to be disconnected when no agent is at the target. The design of the visiting schedule of agents to one target can affect the reachability (controllability) of this target's control system and the design of any specific controller. Existing test techniques for reachability are combined with the idea of lifting to provide conditions on systems such that reachability is maintained in the presence of periodic disconnections from the controller. While considering an intermittently connected control with constraints on the control authority and in the presence of a disturbance, the concept of *degree of controllability* is introduced. The degree is measured by a region of states that can be brought back to the origin in a given finite time. The size of this region is estimated to evaluate the performance of a given sequence.

Contents

List of Figures

List of Abbreviations

2-D	.....	Two-Dimensional
3-D	.....	Three-Dimensional
AFM	.....	Atomic Force Microscope
IPA	.....	Infinitesimal Perturbation Analysis
LQG	.....	Linear Quadratic Gaussian (Control)
LTI	.....	Linear, Time-Invariant
MIMO	.....	Multiple-Input, Multiple-Output
MIP	.....	Mixed Integer Programming
NCS	.....	Networked Control System
NP hard(ness)	.....	Non-deterministic Polynomial-time hardness
PMP	.....	Pontryagin Minimum Principle
$\mathbb{R}^2$	.....	the Real plane
$\mathbb{R}^3$	.....	the Real space
SISO	.....	Single-Input, Single-Output
SPT	.....	Single Particle Tracking
TPBVP	.....	Two-Point-Boundary-Value Problem
TSP	.....	Traveling Salesma(e)n Problem
VRP	.....	Vehicle Routing Problem

Chapter 1

Multi-agent Multi-target persistent monitoring problems

Using a team of cooperating robots to perform tasks in a finite target space, such as monitoring a changing environment, detecting hazardous spots at a disaster scene, or tracking moving objectives, is an important problem of ongoing interest. Such problems have been studied in (Stone and Veloso, 2000; Li and Jilkov, 2003; Shoham et al., 2003; Horling and Lesser, 2004; Ren et al., 2005; Busoniu et al., 2008; Yang et al., 2008; Smith et al., 2011a; Stump and Michael, 2011; Mathew et al., 2013). Unlike the type of problems that falls in the concept of coverage or exploration problems (Zhong and Cassandras, 2011; Sun and Cassandras, 2015; Cassandras et al., 2013; Lin and Cassandras, 2013), such that every point in the target space is of interest, there are a myriad of scenarios in which we are particularly interested in *some* rather than all points in the target space. The points could be data sources that we need to collect information from, or locations with potential risks that need our frequent attention. These points, which could be fixed or moving, are considered as 'targets' in this thesis, while the robots need to move around the space and are referred to as mobile 'agents' in this thesis.

The problem of using multiple agents to track and monitor targets is a class of problems that have been under investigation for decades (see, e.g. the survey articles (Li and Jilkov, 2003; Li and Jilkov, 2005)). The fundamental paradigm in this class of problems is to use the collection of agents to jointly optimize some cost func-

tion that captures the desired features of the tracking and monitoring problem (Ren et al., 2005; Horling and Lesser, 2004). The potential applications cover a wide range of spatial and temporal scales, from macro systems, such as battlefield surveillance, ocean environment monitoring, or air traffic management(Spengler and Schiele, 2003; Mao, 2009; Koller et al., 1993), down to systems with nanometer-scale features, such as methods based on the tracking of fluorescent or magnetic particles used in bio-molecular and nano-medicine research (Ashley et al., 2012; Cromer Berman et al., 2011). This wealth of application domains has generated interest and efforts from a variety of fields, such as systems and control, machine learning, and artificial intelligence (see, e.g., (Li and Jilkov, 2003; Li and Jilkov, 2005; Stone and Veloso, 2000; Busoniu et al., 2008; Shoham et al., 2003)) and covers a variety of sub-fields, including persistent coverage (Smith et al., 2012).

In many, if not most, applications, it is of primary interest to develop decentralized schemes so that each agent can act either independently or based only on information shared among neighboring agents (De Gennaro and Jadbabaie, 2006; Yang et al., 2008). However, when feasible to use, a centralized controller has significant advantages, such as avoiding communication or synchronization difficulties, analyzing information from the entire collection of agents at all times, and the ability to design control policies that are aware of the state of the entire system. Beyond understanding fundamental limits in performance, centralized control is relevant in applications with nano-scale systems where the agents are often controlled by a single computing platform.

The multi-agent multi-target persistent monitoring problem can be decomposed into two parts: a centralized scheduling algorithm that dispatches each of the agents in a coordinated manner to different targets, and a control policy that agents execute to collect information from the target assigned to it or to drive state of target as

desired. Our measure of performance is a function of the targets' states, which can be timeliness of data at a target, measurements of a process that stochastically evolves in time, or a variety of other situations. While a target is being attended by an agent, its state is driven towards the desired state through the agent acquiring information or executing control on it. Targets that are not being attended have states that continue to evolve, drifting the state away from the desired value. The problem can be viewed from at least two perspectives. From the agents view, the motion trajectories for the agents need to be designed as they move among the spatially distributed targets. From the targets view, we care about whether the overall system remains "controllable" in an appropriate sense (such as whether the state for each target can be driven arbitrarily close to a desired point) despite the need to switch agents between targets and the time it takes to do so.

1.1 Applications at different spatial scales

There is a surprisingly rich number of applications where this problem may find use. They may be found across a broad range of spatial scales, from as large as the surveillance in modern smart city, environment monitoring, energy management (Michael et al., 2011; Smith et al., 2011a), down to as tiny as studies of the dynamics of biological macromolecules.

One example of a nano-scale application is single particle tracking (SPT). This is a class of experimental techniques used study the motion of bio-molecular particles to understand their interaction with a specific environment. The method has been used to determine the moving mechanisms of a myosin VI along an actin molecule (Yildiz et al., 2004), or the spreading feature of certain type of virus (Lakadamyali et al., 2003). These objects cannot be observed directly through normal optical microscopes, as they are smaller than the diffraction limit of visible light.

To overcome this limitation, super-resolution methods can be applied (Moerner and Fromm, 2003). Small fluorescent particles are attached to the objects we are interested in. When excited by lasers with a certain range of wavelength, the fluorescent particles will emit a fixed wavelength light, which produces a point spread function that is visible under optical microscopes.

Now that the nano-scaled particles become 'visible', different tracking methods can be applied. One approach is to take a series of wide-field images taken through microscopes. Various algorithms can be used to localize the particles in each frame and then link locations across frames to make trajectories. This class of methods uses the microscope passively, that is, independent of the position or motion of any of the particles being tracked. A microscope can only observe an object clearly when it is within a very limited axial distance. To observe objects distributed in different axial depth, the focus of the optical system needs to be adjusted. Obviously, only objects within a small range of axial depth can be 'seen' in the same photo. Thus, taking a series of 2-D photos and comparing them, we may extract a clear trajectory of a particle only when it is moving in a single plane.

As an alternative, there are active methods that use feedback to follow a particle in three dimensions (Ashley et al., 2012). Based on this idea, we are able to study how to bring this method to an efficient multiple-agent multiple-target tracking paradigm. This method takes the focus point in a 3-D sample as a moving agent. By counting the photon density the agent can acquire at its current position, it estimates the possible position of the particle and 'chases' it. Since the photon density is higher closer to the particle, linear or non-linear extremum seeking control laws can be applied to drive the agent approaching the particle. Existing work of (Shen and Andersson, 2011) shows how an Linear-Quadratic-Gaussian (LQG) control can be applied to realize the tracking of one particle using a single agent. (Ashley et al., 2016) shows the

realization of such function using an extreme seeking control scheme.

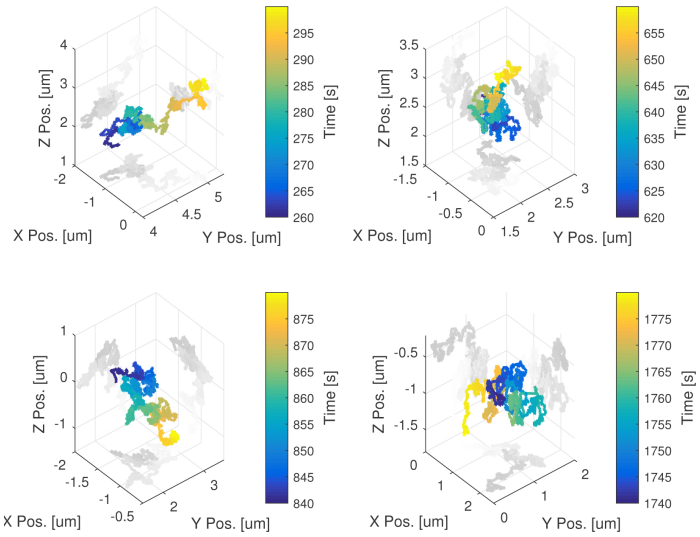


Figure 1·1: Three-dimensional trajectory of a quantum dot diffusing in a hydrogel tracked in a confocal microscope using the method presented in (Ashley et al., 2016).

When applying multiple-agent multiple-target tracking on fluorescent particles, we assume that a pre-knowledge of an approximate position of the particles are estimated based on the information gained through the wide-field camera. This information is not exact, which we consider as an 'uncertainty' in our knowledge of their positions. The focus points are seen as agents. While an agent is sent to the estimated position of the target assigned to it, it begins searching for the extreme of the photon density. With time passing, there will be more information collected and we will establish a better knowledge of the position of the particle. This process is viewed as a decrease in the uncertainty relate to the target particle. The focus point is controlled by shifting a piezo sample stage, and therefore a holonomic motion of the agents is possible. In addition, all control is done through a single, centralized computer, motivating centralized solutions.

1.2 Optimized dispatch of agents towards targets

As mentioned before, the class of problems we consider is composed of two levels of design. At the level of the agents, the problem we care about is the design of the agents' motion, defining how and when the agents are dispatched to the targets.

There is a significant body of work in the literature on the scheduling problem, following (at least) two approaches. Under the first approach to the scheduling problem, the target space is described as a continuous domain. One class of schemes uses parameterized curves to define the agent trajectories, transforming the problem into one of minimizing a given cost function over the target space, which allows gradient based solutions.

Under the second approach, the targets are viewed as discrete tasks that are assigned to agents according to a carefully designed schedule. The schedule is composed by the sequence specifying which targets and in what order one agent will go to visit, and a timeline that determines how long the agent spends at each target in the sequence.

1.2.1 Gradient based optimization in a continuous domain

In this subsection we will discuss a scheme that is not pursued in this thesis, but will be used in Sec.2.1.2 as a comparison. While modeling the target space as a continuous domain, as in existing works such as (Lin and Cassandras, 2013), a value is assigned to every point in the target space. Depending on the agents' sensing radius, the points within a certain distance from a target carry such values associated with the dynamics of the target's states, which evolve with time. The dispatch of agents may be accomplished by designing closed loop motion trajectories that lead the agents around the space to visit the targets periodically and possibly linger around the targets to perform the control tasks. The trajectories can be modeled as time

variables representing the agents' velocity and directions. The problem then turns into an optimal control problem, with the objective to minimize some cost function that captures the desired features of the target space (Horling and Lesser, 2004).

Existing work (Zhou et al., 2016) showed that the optimal control problem can be reduced to a parametric optimization problem in a one dimensional target space. The work proved that the agents should either move at their top speed or stay still in their optimized trajectories. Therefore, an optimal agent trajectory can be characterized by a finite number of *events*. An event refers to the agent switching motion direction or switching between moving and stay still. The agent's motion can therefore be modeled as a hybrid control problem, and event-driven methods, such as Infinitesimal Perturbation Analysis (IPA) can be introduced to determine the gradient of the optimization objective over the event locations (Cassandras et al., 2010; Wardi et al., 2010; Cassandras et al., 2013). A straightforward gradient based algorithm can then be used to find out the (local) optimum. Such an approach inherits many advantages from event-based methods. For example, it has a great tolerance to stochastic effects in the targets' dynamics. The most important property superior to other approaches is that this approach is scalable in the number of events, rather than the number of agents and targets. On the other hand, event-based methods uses the gradient of the objective function only when an event is triggered by a trajectory that an agent chooses. Notice that an event usually occurs while the agent passes by a point associated with a changing value, which is within the target's neighborhood. When the target space is sparse, there would be a large portion of the target space that carries zero value. It is highly possible that a random trajectory never travels through any of target's neighborhood, and therefore triggers no event. The lack of any event may cause serious problems in such optimization tasks (Schwager et al., 2009; Cao et al., 2011; Oh and Ahn, 2014), and can be solved, as shown in (Khazaeni and Cassandras,

2016), by introducing potential fields related to the target's dynamic states to fill in the sparse area in the target space, such that the gradient of the objective function can exist even no event was triggered. The agents can then be driven to explore the whole target space and finally approach some target and trigger an event.

1.2.2 Graph based task scheduling problem

The other approach is to view the targets as tasks to be accomplished according to a pre-designed visiting sequence and a timeline. The problem is then formulated using finite automata to describe target dynamics (Smith et al., 2011a; Lahijanian et al., 2010; Smith et al., 2011b; Mathew et al., 2013) with the geometry of the target locations described by a graph where each vertex is a target (or task) and each edge carries a weight which is a function of the distance between the connected targets (Lan and Schwager, 2013). While a visiting sequence is selected for an agent, the problem is reduced to optimizing a given cost function of the targets states as a function of the amount of time to spend at each target (and the controller to apply).

When considering an infinite horizon problem, it makes intuitive sense to focus on periodic schedules as these are simpler to implement than aperiodic ones. It has been shown that under certain conditions, periodic schedules can be designed that ensure the entire system, from the point of view of managing the uncertainty levels of the targets, remains controllable despite the delays incurred by traveling from one target to the next (as shown in Sec.3.2 and (Yu and Andersson, 2013; Yu and Andersson, 2014)). In addition, given a particular sequence it is clear that the uncertainty levels at the targets are a function of the dwell times. The dwell time can therefore be viewed as a control input that can be optimized to determine the best cost of the given sequence. This can then be followed by an evaluation of a collection of such sequences to find one that achieves the best overall performance.

To solve such problems is computationally challenging in at least two aspects: the

evaluation of the cost of a sequence involves determining the optimal dwell times, requiring a nonlinear optimization over multiple variables, while finding the globally optimal visiting sequences for the agents requires enumeration of all such sequences, which scales up rapidly with the number of agents and targets (Yu et al., 2015). When viewed in this graph theoretic way, the problem is clearly related to Traveling Salesmen Problems (TSPs) and Vehicle Routing Problems (VRPs). TSPs and VRPs have a long and rich literature and, while they are NP hard, there are many sub-optimal approximations that have been developed that permit solutions to be found rapidly (Laporte et al., 2000; Bektas, 2006). However, these problems have well-defined edge costs that do not depend on the visiting sequence; approximate solutions rely on this property. In our persistent monitoring problem, a change at any part of the sequence alters the optimization problem that must be solved to find the dwell times at each target and thus the cost of the sequence. As a result, the approximation procedures from TSPs and VRPs cannot be used.

Computation of the dwell-time is challenging due in part to the non-linearity of the problem and is made even more complicated by the interaction between the different agents. In one-dimensional spaces (where the targets are distributed on a line), it has been shown that the global optimum consists of a partition where any two agents share no more than a single target (Cassandras et al., 2013). The same is not necessarily true in higher dimensional spaces (Lin and Cassandras, 2015). To simplify the problem, a commonly used approach is to partition the workspace into non-overlapping sets of targets with each set assigned to a single agent. This is often done through Voronoi partitions (Breitenmoser et al., 2010; Gusrialdi et al., 2008). In Sec.2.2.1.4 we also discuss the selection of a sequence based on maximum matching algorithms.

Sec.2.2.2.3 and Sec. 2.2.3.3 focus on the selection of sequence for the single agent

case. To reduce the computational load we study the possibility of analytically solving for the optimal dwell time. Sec.2.2.2.3 starts by assuming the agent moves through a periodic cycle, visiting each target in its cycle exactly once (as shown in (Yu et al., 2017)), showing first that under reasonable choices of the dwell times, the dynamics of the target will always converge to a steady state, periodic cycle. We then go on to show that if the dwell times are chosen by the rule that the agent remains with its current target until the state is driven to zero and then transitions immediately to the next target in its sequence, then many of the cost functions of interest, such as the peak value of the information state and the time length of the cycle, are minimized. Imposing a homogeneity assumption on the targets results in the average information state also being minimized and translated the entire problem to a 'simple' TSP.

Sec. 2.2.3.3 continues the work by relaxing the constraint of requiring exactly one visit for each target in a given scheduling sequence for a single agent. While this is a small change, it has a major complicating consequence. By relaxing the single visit requirement, the number of sequences to consider becomes infinite. To overcome this challenge, we establish that the switching policy of staying with the current target until its information state is driven to zero and then immediately moving to the next target remains optimal under certain assumptions. With this choice of dwell time, the average state value of the targets in the cycle can then be calculated analytically. This allows us to compute an upper bound on the number of visits to any given target in a sequence for it to be optimal. This bound implies that the search of the optimal sequences can be reduced from infinite to a very large but finite number. Finally, an algorithm similar to edge substitution algorithms for TSPs is designed to adjust a given sequence towards a better one to reduce the average state of the targets.

1.3 Design of controller to regulate the targets' dynamic systems

The second part of the problem is the design of the controller that an agent will use at a particular target to steer the target's state as desired.

Most existing methods under either the discrete or continuous formulation assume very simple dynamics for the targets' states. For example, in (Zhou et al., 2016; Yu et al., 2017), the state of each target is described using a single scalar variable whose evolution is described by a simple linear increase (in the absence of an agent) or decrease (when being attended by at least one agent). The methods are typically agnostic to the details of the dynamics so long as some reasonable assumptions are satisfied, such as a monotonic decrease towards zero in the presence of an agent.

More generally, however, the dynamics of the targets may be described by a multivariable state and the effect of an agent may not be to generate a simple decrease in the target state. In this work, the (multidimensional) state of each target evolves according to a linear, time-invariant (LTI) system. In this setting, the design of both the sequence schedule for the agents and the controller to be used when visiting a given target become interesting and coupled problems.

From the point of view of the targets, the mobile nature of the agents leads to intermittent control. When no agent is at the target, the controller is assumed to be disconnected such that the input remains zero; when an agent arrives the controller is connected, giving the agent the ability to affect the target dynamics. This intermittent connection can be modeled as the control of a large number of plants (the targets) using a small number of sensors and actuators over a limited bandwidth network (the agents). Formulating the model in this way, namely as a networked control system (NCS) problem, implies that the design of the visiting schedule of the agents to the target can affect the controllability (or reachability) of the combined control system

as well as influence the design of any specific controller.

The question of reachability for LTI systems is well studied in the literature. A variety of tests have been developed, such as the Hautus test (Hautus, 1970) and the Kalman rank condition, that use the controllability Gramian to provide a simple binary result as to whether the system is controllable or not. Such tests have also been developed for more complex systems, such as those with switched modes or more general hybrid models (Blondel and Tsitsiklis, 1999; Lygeros et al., 1999; Sun et al., 2002). This basic question has also been studied under the networked control setting where the controller must be switched between systems (Zhang and Hristu-Varsakelis, 2006). One major assumption in (Zhang and Hristu-Varsakelis, 2006) was that the communication policy had the property that at least one actuator and one sensor were connected to the controller at each time. In practice, however, the process of switching a network channel between accesses may take a non-trivial amount of time. Physical examples of this include any setting in which a sensor must be physically moved between two locations, such as when a group of unmanned aerial vehicles is tasked with investigating and tracking multiple ground vehicles spread throughout an area that is large with respect to the sensing range of the robot. In this thesis we refer to the time it takes for a switch to occur, during which no information can be passed through the channel, as a *switching delay*.

The switching delays considered here should not be confused with time delays or a packet loss. These delays do not occur randomly but rather occur with a timing determined by the communication sequence and thus, ultimately, by the control designer. In addition, during the delay, no information at all can be passed through the channel. A simplified version of this setting is considered in Sec. 3.2, where the plant is connected to the network for q steps and disconnected for r steps. We used the techniques from (Zhang and Hristu-Varsakelis, 2006), combined with the idea

of *lifting* (Bamieh et al., 1991) to show that, for the case of a single-input, single output (SISO) system, the existence of these blind periods could in fact cause the system to lose the properties of reachability and observability. A test was then derived which could determine, for a given system matrix, whether a given delay time would cause loss of reachability and observability, and a scheme suggested in which the delay could be extended to ensure those properties were maintained. Later we extend the results of the SISO setting to the more relevant case of multiple-input, multiple-output (MIMO) systems. Similar ideas were also developed in (Baillieul and Kong, 2014) where the connection to the controller was random rather than periodic.

Most of the work in this area, including our own previous efforts, assumed there were no constraints on the control inputs. In practice, of course, systems always face limitations on their control authority as well as the presence of disturbances to the dynamics. Including disturbances places the problem into the category of robust control and previous work has focused on computing an *invariant set* for the control system by computing the Minkowski sum of both the input and disturbance signals (Blanchini, 1999; Raković et al., 2007). These results assume a continuous connection between the controller and the plant. In our problem, while the disturbance signal drives the target dynamics at every time step, independent of the presence of absence of any agent, control is available only intermittently due to the presence of one or more agents at any given target. The question of *degree of controllability*, that is, of how much control authority is needed, can be traced back at least to the 1970s where (Friedland, 1975) proposed a measure based on the condition number of the controllability Gramian. The Hautus test was used in (Hamdan and Nayfeh, 1989) to develop a measure based on the angles between the eigenvectors of the state and input matrices. In (Viswanathan et al., 1984) and similar works (Viswanathan and Longman, 1983; Klein et al., 1982), the degree of controllability was measured based

on a *recovery region* defined as the collection of states that can be brought back to the origin in a given finite time under limited control authority and methods to estimate both upper and lower bounds on the size of this region were developed. This was further extended in (Junkins and Kim, 1991; Gawronski and Lim, 1996) and later in (Kang et al., 2009) when disturbance rejection was introduced into the story.

Chapter 3 build upon the results of our earlier work in (Yu and Andersson, 2013; Yu and Andersson, 2014) to study the question of reachability of a linear dynamic system under a periodically-connected control and with constraints on the control authority and in the presence of a disturbance. After formulating the problem in Sec. 3.1, we focus in Sec. 3.2 on the unconstrained case and establish conditions on the periodic schedules that cause a loss of reachability for a given system and show how to alter such a schedule to regain reachability. In Sec. 3.3 we bring back the control constraints and adopt the notion of a recovery region introduced in (Viswanathan et al., 1984), extending it to the discrete-time setting considered here. We develop a method for calculating the size of this recovery region. By taking the ratio of the size of the recovery region to the size of the "escape" region , we establish a measure of the degree of reachability. Finally, we use this to determine the optimal number of connected periods for a periodic sequence.

1.4 Contribution of the thesis

There are two main contributions within this thesis. The first is the design of a periodic schedule for one or more agents moving around a finite number of targets, repeatedly visiting them to collect information and reduce uncertainty about the target. We consider and compare a graph-based and a gradient-based approach to model this problem. Following the graph-based approach, we present a method to enumerate the agents' visiting sequence, and show that with a given visiting sequence, the

problem is translated into a discrete-time dynamic system with the targets' sampled uncertainty level as the state vector and the dwell time at each target as the input vector. In the one-agent case we show that under a mild assumption and with a constant input this discrete-time system converges to an asymptotically stable steady state and that the underlying continuous dynamics converge to a periodic cycle with a fixed period. We show further that the sampled uncertainty, the peak uncertainty, and the period are all minimized under the policy that the agent switches to the next target in its sequence as soon as the uncertainty of the current target is reduced to zero. Additionally, we show that if the average uncertainty over the steady state period is taken as a measure of performance, then the same policy is optimal under the additional assumption that the targets are homogeneous. By extending the results on the optimal switching condition into a general case, such that multiple visits to one target within one period is allowed, we use the optimized condition to narrow the search for an optimal sequence from an infinite number of possible sequences to a finite (though possibly very large) number by determining an upper bound on the number of visits at any given target. Finally, we develop an algorithm for modifying any given sequence to reduce the average information level summed across all the targets.

The second main contribution of this thesis is to analyze the target's controllability / reachability under the frame of multi-agent multi-target persistent monitoring problem, such that the states of each of the individual targets are only controlled intermittently. We assume that the state dynamics of each of the targets are given by a linear, time-invariant, controllable system and develop conditions on the visiting schedules of the agents to ensure that the property of controllability is maintained in the face of the intermittent control. We then introduce constraints on the control authority and a bounded disturbance into the target dynamics and develop a method

to evaluate system performance under this scenario. Finally, we use this method to determine how the amount of time the agents spend at a given target before switching to the next in its sequence influences the control of the states of the entire collection of targets.

Chapter 2

Optimal Scheduling of Multiple Agents Monitoring Multiple Targets in a Finite Space

In this chapter we introduce both the gradient-based and graph-based approaches for generating the optimized schedules for agents monitoring multiple targets in a finite time horizon. Sec. 2.1 compares both approaches while in Sec. 2.2 we present a detailed analysis of the graph-based one. In Sec. 2.2 we study how to find a good set of sequences for multiple agents with a finite time horizon. We then consider the infinite horizon case and search for periodic schedules. In the special case of a single agent, we show how to calculate the optimized dwell time and how this result can be used to filter available sequences.

2.1 Optimized Schedules Generated Under a Continuous and a Discrete Model

We consider N targets in a finite target space. Let the position of the targets be $s_i \in \mathbb{R}^1$, $\mathbb{R}^2$, or $\mathbb{R}^3$, $i = 1, \ldots, N$. In the same target space there are M mobile agents. Let the position of them at time t be $x_j \in \mathbb{R}^1$, $\mathbb{R}^2$, or $\mathbb{R}^3$, $j = 1, \ldots, M$. Each of the agents can move at the speed of

$$\dot{x}_j = v_j, \tag{2.1}$$

and the speed is constrained by $|v_j(t)| \leq 1$, $j = 1, \ldots, N$.

The ability of an agent j to sense target i is modeled by a function $p_{ij}(x_j, s_i)$, defined as $p_{ij}(x_j, s_i) = 1$ if $x = s_j$, and such that $p_{ij}(x_j, s_i)$ is monotonically non-increasing in the distance $|x_j - s_i|$, thus capturing the reduced effectiveness of a sensor over its range. This range is denoted by γ_j. Therefore, we set $p_{ij}(x_j, s_i) = 0$ when $|x_j - s_i| > \gamma_j$. That is

$$p_{ij}(x_j, s_i) = \max\{1 - \frac{|s_j - x|}{\gamma_j}, 0\}. \tag{2.2}$$

Accordingly, the joint probability that target i is sensed by any one of the M agents is

$$P_i(t) = 1 - \prod_{j=1}^{N}[1 - p_{ij}(x_j(t), s_i)]. \tag{2.3}$$

Next, we define uncertainty functions $r_i(t)$ associated with targets $i = 1, \ldots, N$, so that they have the following properties: (i) $r_i(t)$ increases with a pre-specified rate a_i if $P_i(t) = 0$, (ii) $r_i(t)$ decreases with a fixed rate b_i if $P_i(t) = 1$ and (iii) $r_i(t) \geq 0$ for all t. It is then natural to model uncertainty dynamics associated with each target as follows:

$$\dot{r}_i(t) = \begin{cases} 0, & \text{if } r_i(t) = 0,\ a_i \leq b_i P_i(t), \\ a_i - b_i P_i(t), & \text{otherwise,} \end{cases} \tag{2.4}$$

where we assume that initial conditions $r_i(0)$, $i = 1, \ldots, N$, are given and that $b_i > a_i > 0$ (thus, the uncertainty strictly decreases when there is perfect sensing $P_i(t) = 1$).

2.1.1 Graph-based approach for agents scheduling

In this subsection we follow the graph-based approach and provide a method to numerically determine the schedule for multiple agents in a finite time window. By this approach we view the targets as tasks, which lead naturally to a graph-based description of the problem (Lan and Schwager, 2013; Smith et al., 2011a; Lahijanian

et al., 2010; Smith et al., 2011b; Mathew et al., 2013). This high level of abstraction allows one to guarantee an optimal solution, though at the cost of a significant computational load. It is worth highlighting, however, that the complexity of such schemes is driven by the size of the graph and they are thus essentially invariant to the underlying dimensionality of the target space.

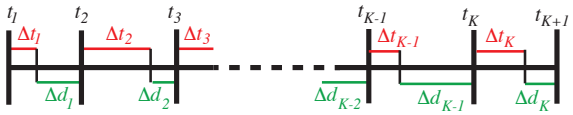


Figure 2·1: Time sequence of a single agent on a given trajectory. The t_j are the time points where the agent begins to move to the next target in the sequence. Each move takes Δt_j units of time followed by a dwell period of Δd_j units of time during which information is collected from the target.

As illustrated in Fig. 2·1, our approach to the discrete setting is to divide the overall planning time horizon T for agent j into a sum of K_j consecutive time steps $\{t_j^1, t_j^2, ..., t_j^{K_j}\}$, $j = 1, \ldots, M$, with $t_j^1 = 0$. The dependence on j indicates that each agent may have a different discretization. We denote the end of the K-th step as $t_j^{K+1} = T$. Each step $k \in \{1, ..., K_j\}$ begins with a travel stage where the agent moves to a particular target i. Under the assumption that during the transition between targets each agent moves at its maximum speed of $|v_j| = 1$, the travel time is

$$\Delta t_j^k = |x_j^k(t_j^k) - s_i|. \tag{2.5}$$

Upon arriving at a target, the agent dwells for a time Δd_j^k. Note that due to the range-based nature of the sensing, the uncertainty actually begins to decrease before the arrival of the agent at the target and continues to decrease after the agent has departed until the target is out of the sensing range.

The problem of optimizing the v_j to minimize the average uncertainty over all the targets can be translated into a mixed integer programming (MIP) problem to select the sequence of targets and the dwell time at each target. Letting α_{ji}^k be a binary

variable denoting whether agent j is assigned to target i at time step k, this MIP is

$$\min_{\alpha_{ji}^k, \Delta d_j^k} J = \frac{1}{T} \sum_{i=1}^{N} \int_0^T r_i(t) dt \tag{2.6}$$

$$\textit{subj.to.} \quad \alpha_{ji}^k \in \{1, 0\}, \quad \sum_{i=1}^{N} \alpha_{ji}^k = 1, \quad \forall j, k \tag{2.7}$$

$$\sum_{k=1}^{K} \Delta t_j^k + \Delta d_j^k \leq T, \quad \forall M. \tag{2.8}$$

Note that we assume each agent is assigned to a maximum of only one target at any one time. The event-driven approach (as described in Sec.2.1.2) has no such restriction. We break the solution of this problem into three parts: enumeration of all feasible trajectories, calculation of the cost of the feasible trajectories, and then selection of the optimal trajectory based on those costs. We focus on the case of a single agent for simplicity of description before generalizing to the multiple agent case.

The first part, namely determining feasible trajectories, is straightforward. Given the fixed time horizon T, the target locations, the locations of the agent at the start of the time horizon, and the maximum speed of the agent, a feasible trajectory is one where the sequence of targets can all be visited within the time horizon. Similarly, the third part simply involves comparing the trajectories and selecting the one with the minimal cost.

In the second part, the cost of each feasible trajectory must be determined. Suppose we have a given feasible trajectory with K targets in its sequence. Note that because a trajectory may include multiple visits to the same target, K may be larger than n (and may be much larger for large time horizons and small n). Let $\{i_1, i_2, \ldots, i_K\}$ denote the indices of the targets in the sequence. From (2.6), the cost

of this trajectory is given by the optimization problem

$$\min_{\Delta d_j^k} J = \frac{1}{T} \sum_{i=1}^{N} \int_0^T r_i(t) dt$$
$$s.t. \quad \sum_{k=1}^{K} \Delta t^k + \Delta d^k \leq T.$$

Our approach to solving this optimization problem is to setup a recursive calculation. As illustrated in Fig. 2·1, since the travel times Δt_j are completely determined by the sequence alone, optimizing over the dwell times is equivalent to optimizing the switching times t_j. Assume for the moment that the switching times through t_{K-1} have been determined (and thus the first $K-2$ dwell times, $\Delta d^1, \ldots, \Delta d^{K-2}$ are known). The two final dwell times are completely determined by selecting the time t_K at which to switch the agent from target i_{K-1} to target i_K. This then gives us a simple single variable optimization problem

$$\min_{\Delta T_K} J = \frac{1}{\Delta T} \int_{t^{K-1}}^{T} \left(r_{i_{K-1}}(t) + r_{i_K}(t) \right) dt$$

where $\Delta T = T - t_{K-1}$. This allows the final switching time to be expressed as a function of the previous time $t_K = t_K(t_{K-1})$. Repeating this leads to an expression of the optimal switching times as a nested sequence of optimization functions which can be solved numerically.

This same optimization procedure can be generalized to the case of multiple agents. The primary challenge is that the set of feasible trajectories, and the calculation of the cost of those trajectories, quickly becomes intractable since all possible combinations of assignments of multiple agents must be considered. The computational complexity can be mitigated somewhat by taking advantage of known properties of optimal solutions (as described in (Zhou et al., 2016)).

Since the computationally complexity is exponential in the length of the time horizon, this approach is most feasible over short horizons. In prior work on linear systems, it was shown that an appropriately defined periodic schedule is sufficient to ensure the entire system remains controllable (Yu and Andersson, 2013; Yu and Andersson, 2014). In the current context, this translates to being able to keep the uncertainty of each of the targets arbitrarily close to zero. Motivated by this, we typically apply our discrete approach over a relatively short time horizon. If the resulting optimal trajectory is periodic, we extend it to longer horizons by simply repeating it.

2.1.2 Gradient-based approach and how it compares with the Graph-based solution

In this subsection we introduced the gradient-based method described in (Zhou et al., 2016) to solve the problem in a one dimensional case, and compare the results with the global optimal results acquired by applying the graph-based approach.

We limit the position of agents and targets to lie on a finite line, such that $s_i, x_j \in [0, L] \subset \mathbb{R}$. We assume that the agents are initially located so that $s_j(0) < s_{j+1}(0)$, $j = 1, \ldots, N-1$, and we wish to prevent them from subsequently crossing each other over all t:

$$s_j(t) - s_{j+1}(t) \leq 0. \tag{2.9}$$

This assumption is non-restrictive as it was shown in (Zhou et al., 2016) that it is a necessary condition for the optimality.

The goal of this approach is to control the movement of the M agents through $v_j(t)$ so that the cumulative uncertainty over all targets $i = 1, \ldots, N$ is minimized over a fixed time horizon T. Thus, setting $\mathbf{v}(t) = [v_1(t), \ldots, v_M(t)]$ we aim to solve

the following optimal control problem:

$$\min_{\mathbf{v}(t)} \quad J = \frac{1}{T} \int_0^T \sum_{i=1}^{N} r_i(t) dt \tag{2.10}$$

subject to the agent dynamics (2.1), uncertainty dynamics (2.4), control constraint $|v_j(t)| \leq 1$, $t \in [0, T]$, and state constraints (2.9). Imagine that each target is associated with a "virtual queue" where uncertainty accumulates with inflow rate a_i. The service rate of this queue is time-varying and given by $b_i P_i(t)$, controllable through the agent position at time t. This interpretation is convenient for characterizing the *stability* of such a system over a time horizon T: For each queue, we require that $\int_0^T a_i < \int_0^T b_i P_i(t) dt$. Equivalently, we may require that each queue becomes empty at least once over $[0, T]$. Note that this analogy readily extends to two or three-dimensional settings.

This problem is formulated as an optimal control problem in (Zhou et al., 2016). By applying the Pontryagin Minimum Principle (PMP) it can be shown that the optimal control $v_j^*(t)$ could only take value from $\{-1, 0, 1\}$. As a complete solution generally involves the solution of a two-point-boundary-value problem (TPBVP), it is not feasible to derive an analytical solution. However, there are a couple of structural properties of the optimal trajectories that can be proved, based on which the optimal control problem can be captured by a set of parameters, namely the time points when the agents switch directions or switch between still or moving modes. Event-driven methods such as IPA can be used to develop a gradient-descent algorithm that yields a local optimal solution.

The numerical examples are presented here to demonstrate the performance of the gradient-based algorithm using the IPA scheme described in this subsection. The results are compared against the global optimal found by the discrete scheduling algorithm of Sec. 2.1.1.

The first simulation consists of a single agent performing a persistent monitoring task on three targets over a time horizon of 100 seconds. The targets are located at positions $s_1 = 5$, $s_2 = 10$, $s_3 = 15$ and their uncertainty dynamics in (2.4) are defined by the parameters $a_i = 1$, $b_i = 5$, and $r_i(0) = 1$ for all i. The agent has a sensing range of 2 and is initialized with $x(0) = 0$, $v(0) = 1$. The results from the IPA-driven gradient descent approach are shown in Fig. 2·2. The top image shows the trajectory of the agent as determined by 1000 iterations of the IPA-driven gradient-based approach while the bottom shows the evolution of the overall cost as a function of iteration number. The agent is clearly moving through a periodic cycle of $s_1 \to s_2 \to s_3 \to s_2 \to s_1 \cdots$, dwelling for a short time at each target before moving to the next. Notice that the agent dwells for a shorter time at the center target since it visits that location twice per cycle. The second image in the figure shows that gradient descent converges within the first 100 iterations. This simulation aims to test the event driven IPA scheme with the discrete scheduling algorithm which yields a global optimal but suffers from a high computational complexity. Thus, we start with a short time horizon $T = 100$ s. Event-driven IPA optimizes the trajectory fast but the convergence is somewhat unstable due to the lack of events within a short time horizon. The final cost is 26.11. The bottom images in Fig. 2·2 shows the evolution of the target uncertainties.

The corresponding result based on the discrete setting of Sec. 2.1.1 is shown in Fig. 2·3. The optimal trajectory is essentially the same as in the IPA-based approach with the agent moving through the three targets in a periodic fashion. The final cost was 25.07, matching that of the IPA approach and thus verifying the approximate optimality of the solution found in Fig. 2·2. Unlike the IPA setting which relies on events to drive the agent trajectories towards optimality, the discrete approach ensures the uncertainties of all three targets consistently brought back to zero.

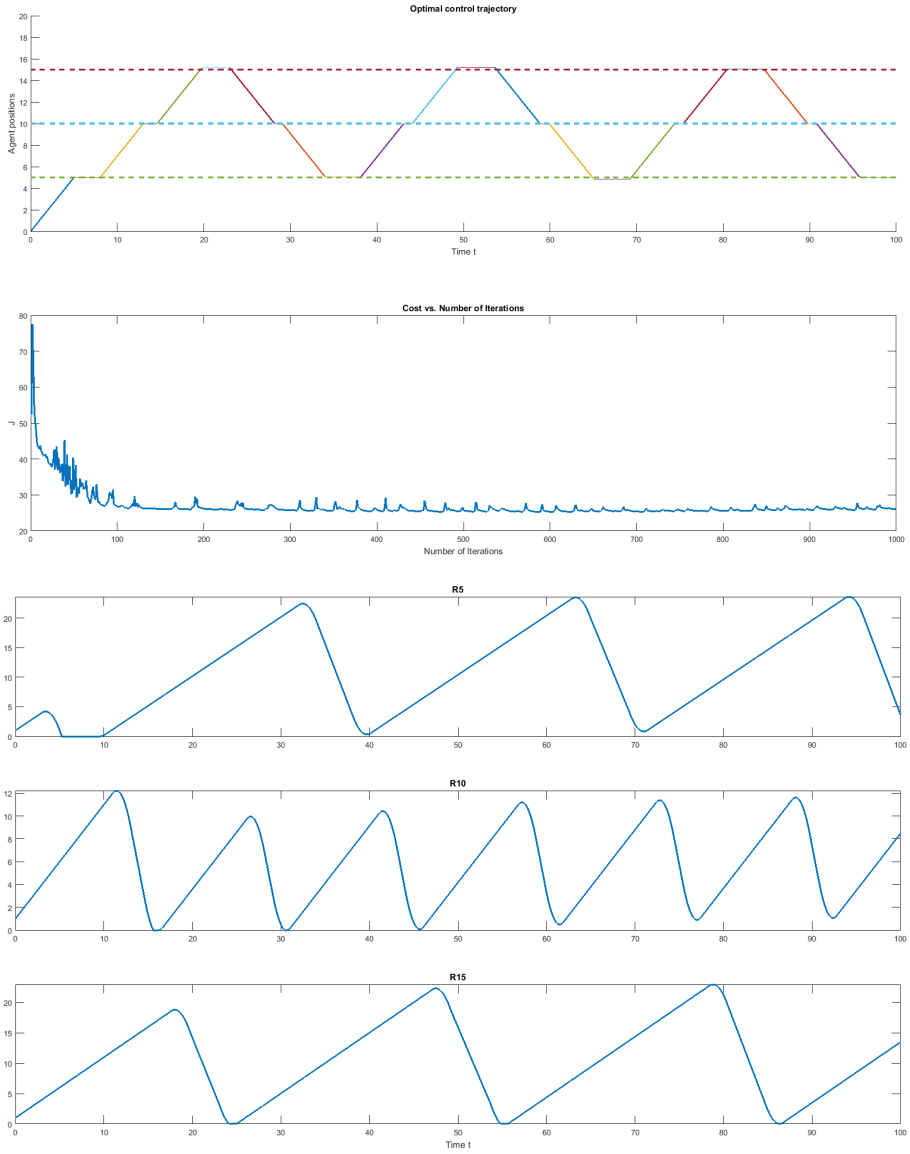


Figure 2·2: A single agent monitoring three targets using the IPA-driven gradient descent algorithm. (top image) Agent trajectory. (second image) Calculated cost as a function of iteration in the gradient descent. The final cost is 26.11. (bottom images) Target uncertainties along the trajectory.

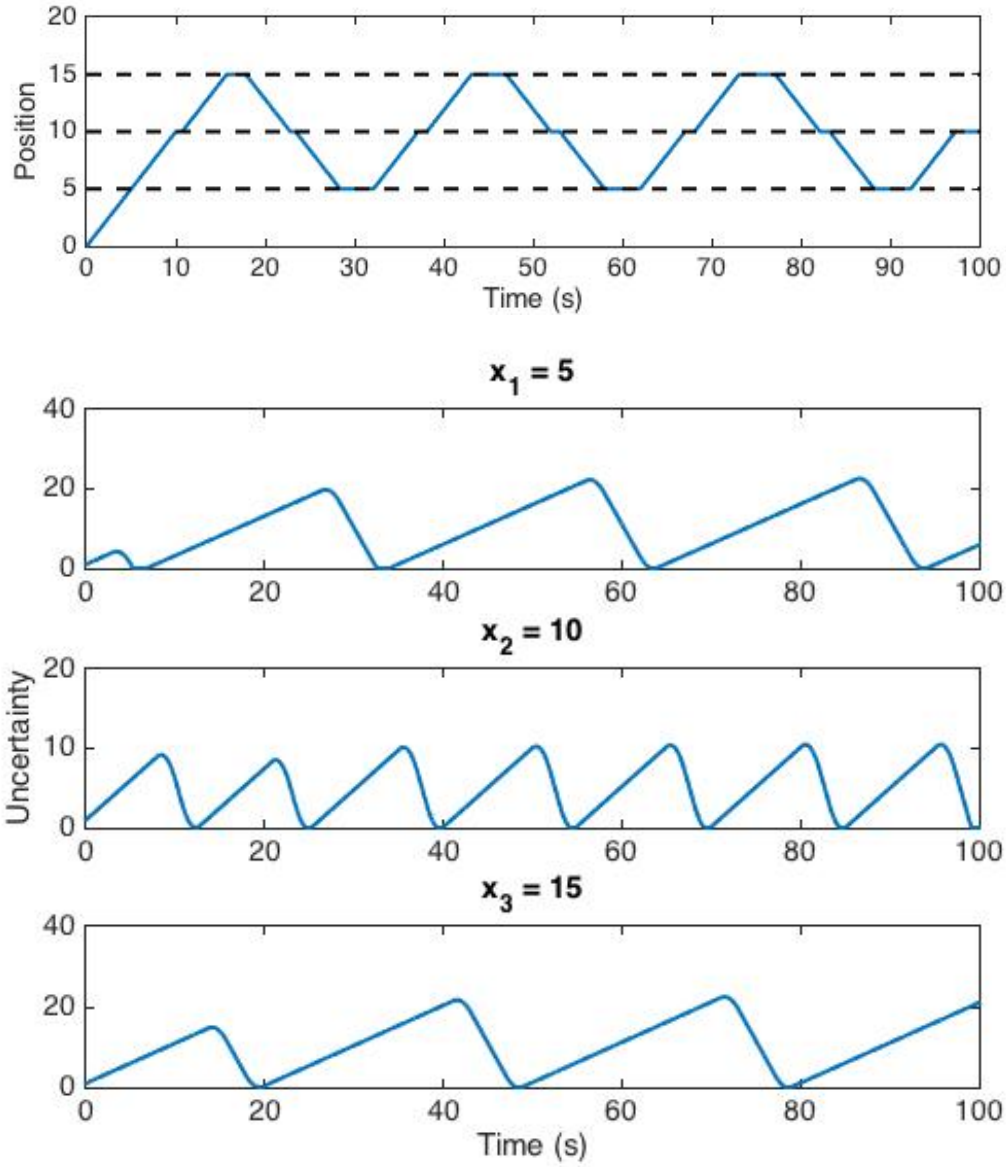


Figure 2·3: A single agent monitoring three targets using the optimal discrete assignment and dwelling time. The final calculated cost was 25.07. (top image) The agent trajectory is the same as in Fig. 2·2. (bottom images) The target uncertainties along the trajectory.

The next simulation involves two agents and five targets over a 500 second time horizon. The targets are located at $s_1 = 5$, $s_2 = 7$, $s_3 = 9$, $s_4 = 13$, $s_5 = 15$. The uncertain dynamics were the same as in the single agent, three target case. As before, the agents have a sensing range of 2 and are initialized at $x_1(0) = x_2(0) = 0$, with $v_1(0) = v_2(0) = 1$. The results from the event-driven IPA gradient descent approach are shown in Fig. 2·4. The solution is again periodic with the agents dividing the targets into two groups. Notice in particular that the single agent on targets s_4 and s_5 is able to keep the uncertainties very near to zero since the targets are quite close relative to the sensing range of the agent. The other agent is able to hold its middle target (s_2) near to zero since it is constantly near that target. Notice also a slight rise in the uncertainties of targets s_1 and s_3 at the end of the planning horizon.

The corresponding result based on the discrete setting is shown in Fig. 2·5. Rather than solving over the full horizon, the problem was solved over a 60 second horizon and then the periodic trajectory repeated to fill the 500 second horizon. The results are again very close to the event-driven IPA method.

2.2 Sequences Enumeration and Optimization of Dwell Time

2.2.1 Multiple sequences assignment

In this section we demonstrate our method inspired by maximal matching to enumerate the agents' visiting sequences. We assume that every target is fixed and carries an uncertainty state that is monotonically increasing unless the target is actively being monitored by an agent. A simulation example with a collection of fixed targets is given in Sec. 2.2.1.4. Our approach uses a receding horizon algorithm and is straightforward to apply to the case of moving targets so long as the motion of the agents is sufficiently fast relative to that of the targets.

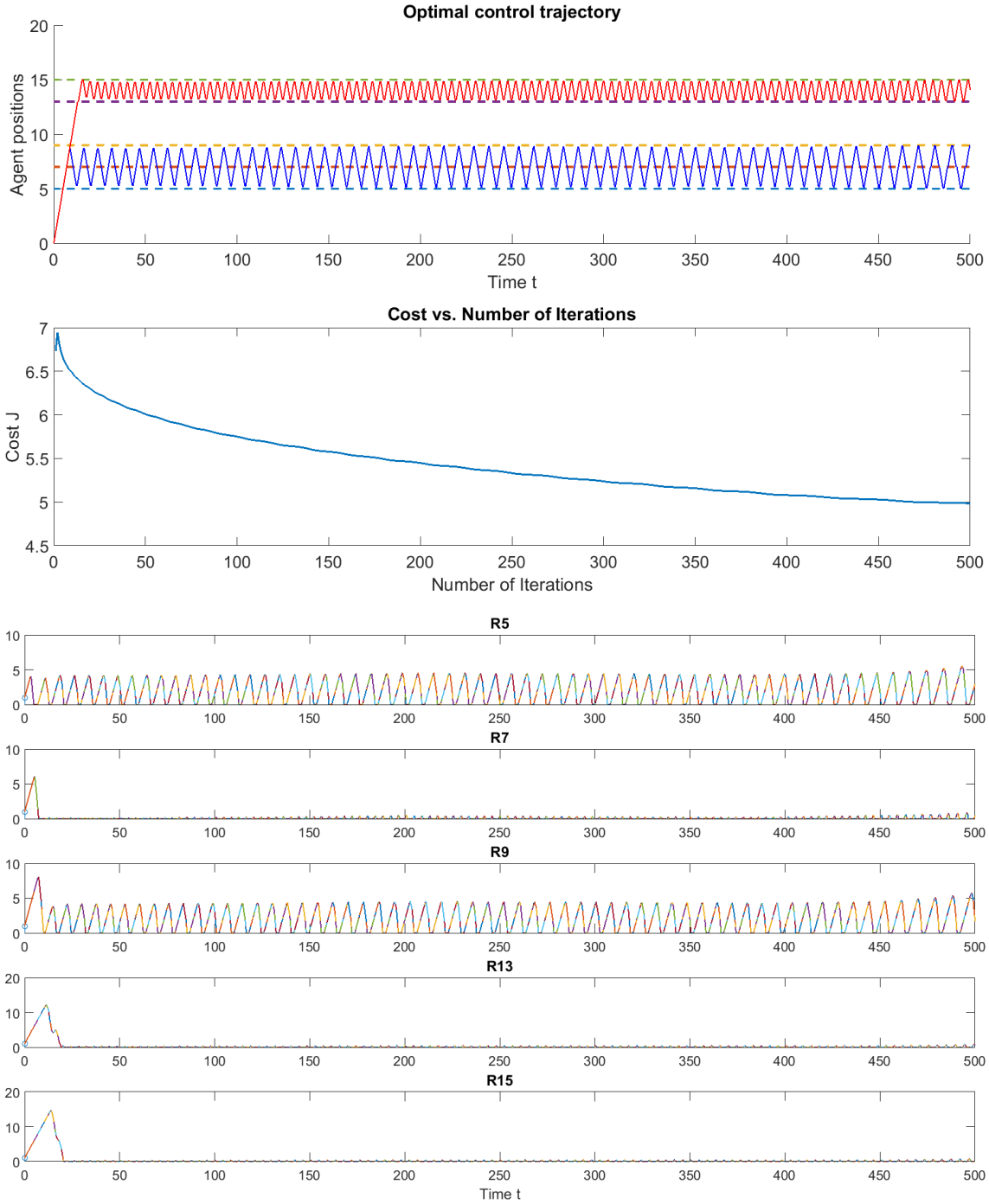


Figure 2·4: Two agents monitoring five targets using the event-driven IPA gradient descent algorithm. (top image) Agent trajectories. (second image) Calculated cost as a function of iteration. The final cost is 4.99. (bottom images) Target uncertainty values along the above trajectories.

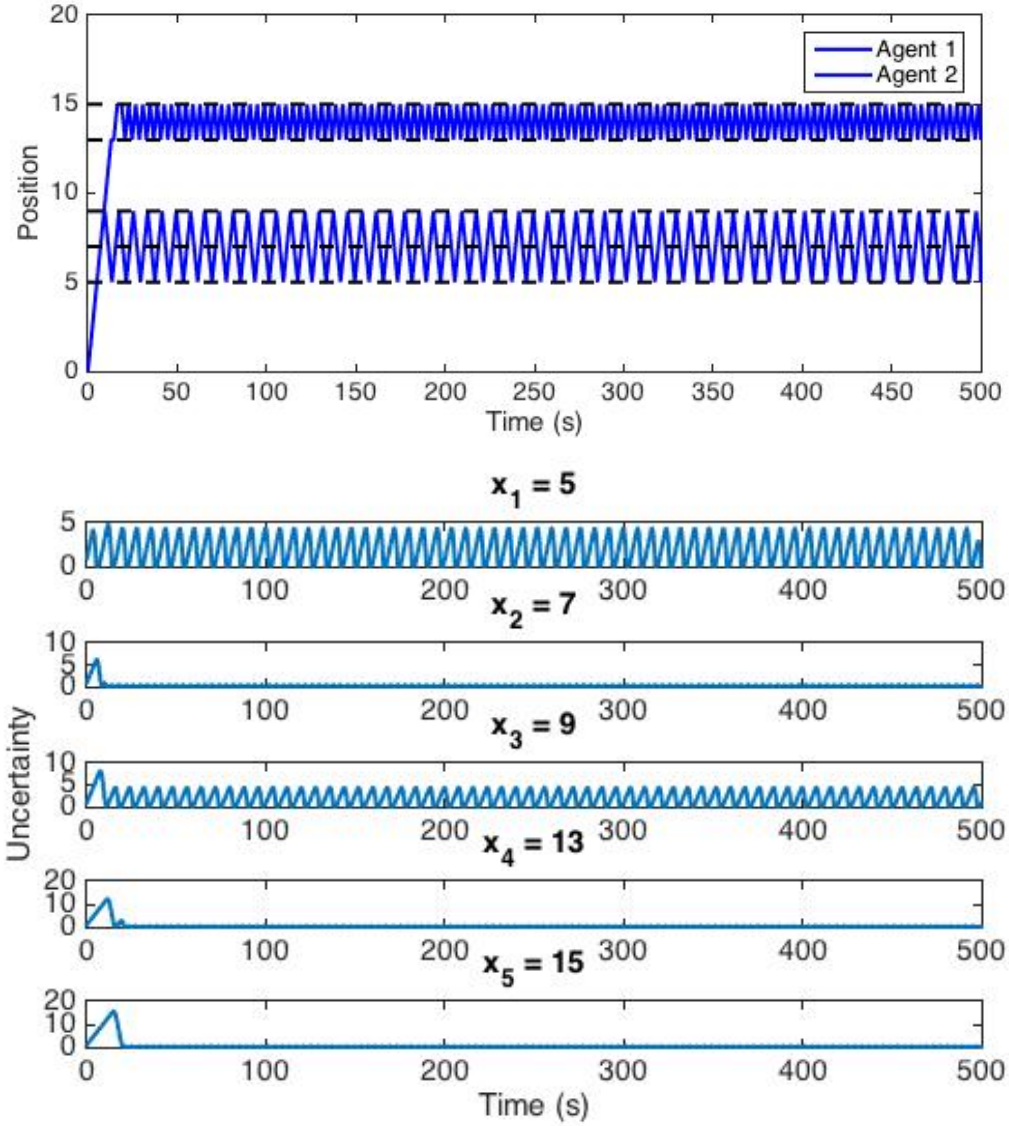


Figure 2·5: Two agents monitoring five targets using the discrete assignment and dwelling time. The final cost was 4.92. (top image) Agent trajectories. (bottom images) Target uncertainty values along the above trajectories.

2.2.1.1 Problem description

We consider a collection of M agents moving in $\mathbb{R}^d$ (with d typically two or three) with positions $x_1, \ldots, x_M$. Agents move according to a kinematic model with maximum speed $\bar{v}_j$. Assignment planning is done at discrete times with intervals Δt and the positions of the agents at these times are denoted $x_1^k, \ldots, x_M^k$. It is shown in (Cassandras et al., 2013; Zhou et al., 2016) that, in general, optimal solutions involve agents moving at maximum speed between targets and as a result we assume that agents always move at $\bar{v}_j$ when transiting between targets and then spend the remainder of the time interval at their assigned target. Note that this implies a finite range for each agent in a time step and a measurement time that depends on the relative positions of the target and agent.

Similar to before, the agents are tasked to monitor N targets moving in $\mathbb{R}^d$ with known, fixed locations $s_1, s_2, \ldots, s_N$ with, in general, $N \gg M$. (While there are, of course, many challenges with discovering the locations of the targets to be monitored, our focus here is only on the dynamic assignment problem.) Each target has a corresponding uncertainty state, r_i, $i = 1, \ldots, N$, representing the state of knowledge about that target. This state evolves in continuous time according to a known function

$$\dot{r}_i(t) = w_i(r_i, x_1, \ldots, x_M, s_i), r_i(0) = r_i^o. \tag{2.11}$$

As with the agent positions, the locations of the targets and the uncertainty values at the planning time t_k are denoted s_i^k and r_i^k, respectively. Note that the dynamics of the uncertainty corresponding to target i depend upon the current uncertainty, the position of all the agents, and the position of the target.

The goal of a control policy is to determine an assignment of each of the agents to a target at each time step with the primary goal of minimizing (or at least reducing)

the overall uncertainty across the entire collection of targets over a given time window of k_s time steps. The assignments at time step k are described using an assignment matrix A_k such that a_{ij}^k is equal to one if target i is assigned to agent j at time step k.

As stated previously, while the assignment problem is described in discrete time, it is important not to lose sight of the fact that the agents must physically move from their current location to their target node. As noted above, this implies that there is a maximum range for each target; we refer to this range as the *effective detection radius* γ_d^j. Further, since the total time of each step is fixed at Δt, the farther an agent needs to move to get to its target, the less time is available for measuring and thus the less the uncertainty state can be reduced. This can be captured in the uncertainty dynamics (2.11) by allowing the dynamics to switch between two different evolution functions based on whether an agent has reached the target and is collecting measurements or not. For example,

$$\dot{s} = \begin{cases} w_i^1(r_i), & \min\limits_{i\in\{1,\ldots,M\}} \|s_i - x_j\| \geq \epsilon, \\ \\ w_i^2(r_i, x_1, \ldots, x_M, s_i), & \min\limits_{i\in\{1,\ldots,M\}} \|s_i - x_j\| < \epsilon \end{cases} \tag{2.12}$$

where ϵ is a small parameter defining how close an agent must be to a target to begin collecting information. The particular form of these expressions can be selected to describe the specifics of a variety of applications. For example, in the previous section, we took a constant increasing and decreasing model to describe how the information uncertainty carried by one target evolves.

For a given assignment of an agent to a target at time k, we can associate a *reward* as the net change in the uncertainty of the corresponding target. That is, if agent j is assigned to target i, the reward is given by

$$W_{ij}^k = r_i^{k-1} - r_i^k, \tag{2.13}$$

where r_i^k is determined from r_i^{k-1} (and from the assignment) by integrating (2.12) over one Δt, starting from r_i^{k-1}.

In solving the assignment problem one seeks to minimize or maximize an overall measurement of all targets. Here we require the agent to maximize the total reward over the planning horizon. Because motion from one target to another essentially incurs a cost due to a reduced time to acquire information, an optimal solution is likely to favor keeping agents at their current target rather than moving them to an unvisited one. In order to encourage greater exploration among the targets, one could take one of two approaches. The first is to explicitly express a decreasing gain in reward over time by, for example, changing the rate of decay of the uncertainty function the longer an agent stays with a target. The second is to explicitly enforce a constraint that says once a target has been visited, it cannot be re-visited for at least a fixed number k_v of time steps. In practice, one would likely use some combination of both approaches.

The problem as outlined here can be formally described by the following binary integer programming problem.

$$\max \sum_{j=1}^{M} \sum_{i=1}^{N} \sum_{k=1}^{k_s} W_{ij}^k \alpha_{ij}^k, \tag{2.14}$$

$$\text{subject to} \quad \alpha_{ij}^k \in \{1, 0\} \tag{2.15}$$

$$\alpha_{ij}^k \cdot \|x_j^k - s_i\| \leq \bar{v^j} \cdot \Delta t, \tag{2.16}$$

$$\sum_{i=1}^{N} \alpha_{ij}^k \leq 1, \quad \forall k, j, \tag{2.17}$$

$$\sum_{j=1}^{M} \alpha_{ij}^k \leq 1, \quad \forall k, i, \tag{2.18}$$

$$\sum_{\bar{k}=k}^{k+k_v-1} \alpha_{ij}^{\bar{k}} \leq 1, \quad \forall k, i, j. \tag{2.19}$$

Here the first constraint ensures agents are only assigned to targets within their effective detection radius, the second ensures that no more than one agent is assigned to each target, the third that no more than one target is assigned to each agent, and the final one ensures that targets are not re-visited within k_v time steps.

Our approach to solving this binary integer programming problem is to recast it as a matching problem on a graph.

2.2.1.2 Matching problem

In graph theory, a 'matching' is the act of connecting each vertex to another with no more than one edge; a connected vertex is said to be "matched". In our setting, we define the agents as one set of vertices and the targets (or sequence of targets) as a second and then seek to match them by connecting between these sets (see Fig. 2·6). By constraining that each agent can visit no more than one target and that each target can be visited by no more than one agent, the problem becomes a standard bipartite matching.

The two classes of matching on bipartite graphs that are relevant to our problem are those of *maximal matching*, where no more edges can be added to the current matching, and *maximum matching* where there is a maximum number of connecting edges. Note that every maximum matching is maximal but not every maximal is maximum. In the monitoring problem, it makes intuitive sense to use every agent if possible and thus in general we seek a maximum matching. However, there are cases where the optimal solution is only maximal, leaving some agents without an assignment.

Finding a maximal matching on a given bipartite graph is relatively straightforward. Essentially, one adds a single edge at a time, trimming all other connections to the affiliated vertex from the list of possible edges. This continues until the list of remaining possible edges is empty, yielding a maximal matching. This is illustrated

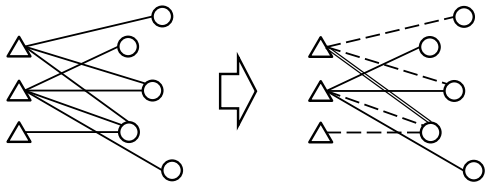


Figure 2·6: Selection of one edge affects other edges in a matching problem. The left side shows all possible edge connections; in the right an edge has been connected (bold line) leading to the deletion of several other possibilities (dashed lines).

in Fig. 2·6 where the left side indicates all the possible edge connections and the right shows a choice of one from the top, left vertex to the fourth, right vertex (indicated by a bold line) and the corresponding deletions from the list of possible edges (dashed lines).

To formally formulate our monitoring problem as a matching problem, we define a bipartite graph $G = (U, V, E)$ as follows. First, all agents are placed in one collection of vertices of the graph, termed U. The second collection V is then constructed from all possible sequences of assignments. Each vertex consists of a single sequence of k_s steps. The cardinality of V is then m^{k_s}.

The edges E contains all feasible edges $e_{j,I}$ linking each agent j to exactly one sequence of assignments $I = [i^1, ..., i^{k_s}] \subset V$. An edge is included in E only if the following conditions on the corresponding agent and targets in the (possible) assignment are satisfied, ensuring each target in the sequence is within the effective detection radius of the agent at each step:

$$\begin{aligned} \|x_j - s_{i^1}\| &\leq \bar{v}_j \Delta t, \\ \|s_{i^1} - s_{i^2}\| &\leq \bar{v}_j \Delta t, \\ &\vdots \\ \|s_{i^{k_s-1}} - s_{i^{k_s}}\| &\leq \bar{v}_j \Delta t. \end{aligned}$$

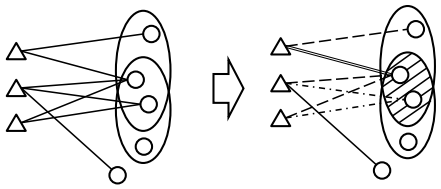


Figure 2·7: Unlike a standard maximal matching problem, an edge selection deletes edges to vertex that share targets in their sequences that overlap within the "no-revisit" window; these edges (shown as dashed-dot lines) are not directly connected to either the original agent or target vertex. The solid line is the selected edge and the dashed lines are edges that are discarded in a standard matching problem (as well as in our formulation).

Each edge carries a reward W_{Ij} that is the sum of the reward from each step of the assignment sequence in the vertex in V connected to by the agent from vertex set U,

$$W_{j,I} = W_{ij} + \sum_{k=1}^{k_s-1} W_{i^k,i^{k+1}}, \tag{2.20}$$

where each term is defined by (2.13) based on the position of the agent at the beginning of step k in the assignment sequence.

Since we are looking for a maximal matching that satisfies all constraints, when an edge is selected, we trim all the other edges connected to the same agent and edges to all the sequences that have at least one assignment in common with the connected sequence within k_v steps. As shown in Fig. 2·7, this is different from a standard matching problem since there are discarded edges that are not directly connected to either the agent vertex or the assigned target sequence vertex. In the figure, the selected edge is the bold line, the dashed lines are the standard discarded lines, and the dash-dot lines are those that are discarded because they share a target in their sequence with the sequence of the selected vertex.

The optimal solution we seek should provide every agent a series of assignments in the scheduling time window of k_s steps while respecting the constraints described

in the binary integer problem (2.19). Our solution is presented in the next section.

2.2.1.3 Algorithms

We break our solution into two parts. We first optimize over a finite horizon and then use this in a receding horizon scheduler.

Finite horizon planning

We consider first a single planning stage over a horizon of k_s steps (keeping in mind that each step lasts a physical duration of time of Δt). The essential idea is to build up a multidimensional reward matrix $W_{1:k_s}(j, i^1, \ldots, i^{k_s})$ that takes an agent j and a sequence of target assignments $i^1, \ldots, i^{k_s}$ and maps it to the corresponding reward received by that agent. The total reward, then, is the sum of the rewards to each agent. The challenge in the construction of $W_{1:k_s}$ is that each choice of a possible assignment by an agent affects the rewards of the assignments of all the other agents due to the constraints of only one agent per target per step and no re-visits in k_v steps.

As described in Sec. 2.2.1.2, a possible assignment $[j, I] = [j, i^1, i^2, ..., i^{k_s}]$ of sequence I to agent j is included as a vertex in the graph only if the first assignment, i^1 is within the effective detection radius of agent j, and every following target i^{k+1} is within ${r_d}^j$ of its prior target i^k, since the agent will be at the spot of i^1 after visiting its first assignment. However, rather than completely disallowing such sequences, we assign them a reward of zero; the assignment algorithm (described below) then ignores such sequences. If all ranges are within the effective detection radius, the reward of an assignment is given by the sum of the per-stage rewards, as in (2.20). Since $W_{1:k_s}(j, i^1, \ldots, i^{k_s})$ is indexed by the agent number and the target assignment at each step, it has dimension $M \times \underbrace{N \times ... \times N}_{k_s}$.

Because the reward of an assignment sequence depends on the other assignments, $W_{1:k_s}$ must be built up anew for each possible sequence. Thus, the calculation of a reward for a collection of assignments begins by assigning to the first agent a sequence with non-zero reward (thus implying it can reach each target in the list). All the related sequences for all other agents are then assigned a value of zero. The second agent is then assigned a sequence of non-zero reward, the related sequences of the remaining agents are set to zero, and so on until an assignment is made for each agent. If at any time an agent only has zero-reward sequences available, it is assigned to stay still. The total reward of this set of assignments is then recorded.

The overall optimal solution is found by enumerating all possible assignments and comparing their rewards sum; this is done by systematically moving through all possible combinations. The computational complexity of this, however, is mitigated by the fact that the reward of many sequences is set to zero and are omitted from consideration. Despite this, it is clear that the number of possible sequences grows exponentially in the number of time steps in the planning window. Thus this method is effective only for a short horizon, motivating the use of a receding horizon controller as detailed in Sec. 2.2.1.3.

Note that the algorithm assumes that if an agent has only zero reward sequences available, it is assigned to not move. In a real application, these agents present an opportunity to explore the space and search for new targets.

Receding horizon planning

A receding horizon approach provides two major benefits at the cost of a sub-optimal solution. First, it allows for persistent monitoring over essentially infinite horizons by constantly recalculating over a short, computationally feasible, horizon. Second, this recalculation offers the opportunity to update the model; for example, if the targets are slowly moving then they can be treated as fixed over a short horizon and their

positions then updated as the next control is calculated.

Our receding horizon planner follows a standard approach, leveraging the finite horizon planner in Sec. 2.2.1.3. That is, we plan using the finite horizon scheme to produce an agent-target assignment over a (short) k_s look-ahead planning window. The first step of the sequence is then executed by each of the agents, and a new sequence then determined by again planning over the next k_s window. Even if the targets are assumed to be fixed, the agents are changing positions at each time step and thus the rewards in $W_{1:k_s}$ need to be recalculated at each step. We can account for limited knowledge of the future (or the relative importance of the first step over subsequent steps) by multiplying a step's reward by a discount factor ω_k with

$$\omega_1 = 1 \geq \omega_2 \geq \cdots \geq \omega_{k_s}.$$

2.2.1.4 Simulation Results and Discussion

To demonstrate the algorithms, we present a simulation result involving two agents monitoring ten fixed targets in a cube of dimension ten units per side. The effective detection radii of the agents were randomly selected (so that each differed somewhat) from a Gaussian distribution with mean 3 and standard deviation 0.2. The original locations of the agents and targets were generated randomly, uniformly distributed in the cube. For illustration purposes, a simple cost function was defined where the reward of assigning agent j to target i at step k was given by

$$W_{ij}^k = \frac{1}{\|x^n - s_i\|}.$$

Note that the reward depends upon the inverse distance between the position of the agent at the beginning of the step and the position of the target, thus reflecting the decrease in reward due to the time it takes the agent to move to the target. Note also that the reward structure does not need to explicitly account for the effective

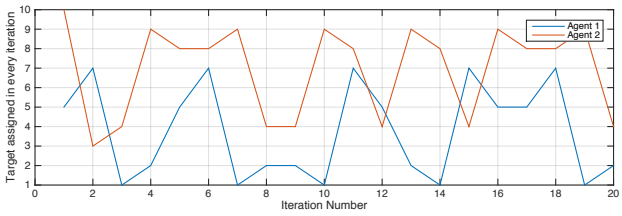


Figure 2·8: Executed target assignments for two agents monitoring ten targets over 20 iterations using a 3-stage look-ahead receding horizon planner.

detection radius as the scheduling algorithms introduced in Sec.2.2.1.3 separately enforce that constraint.

We used the receding horizon scheduling scheme with a three-step look-ahead planning stage followed by a one step execution at each iteration. A discounted reward structure was used where the first step was weighted with $\omega_1 = 1$, $\omega_2 = 0.6$, and $\omega_3 = 0.3$. The no-revisit window was set to $k_s = 3$.

The receding horizon scheduling algorithm was simulated for 20 iterations. The optimal assignments are shown in Fig. 2·8. Note that each agent visits multiple targets with one converging to a three-target loop and the other to a four-target loop. One of the targets was far away from the initial condition of both agents and was not visited in the 20 iteration simulation; since both agents had settled into loops, it is likely this target would never be visited. The other nine targets were visited with seven being visited multiple times on one of the two loops. The actual trajectories in the cubic workspace are shown in Fig.2·9. Notice that the agents divided the targets not only according to their distance to each agent; a group of several targets that can form a tight loop were more likely to be assigned together to the same agent.

Intuitively we would expect an agent to perform periodic trajectories once a loop is formed. The simulation results, however, revealed some other possibilities, despite the simplicity of the setup. An agent may reverse its direction on the loop, or even stay with one target for a while, despite the fact that this yields no reward for that

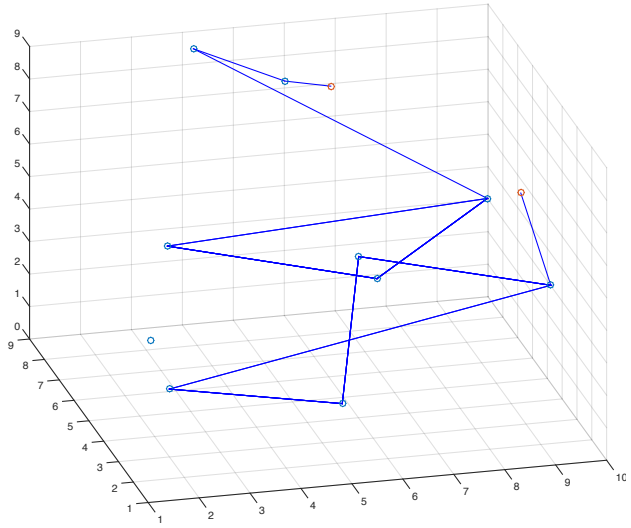


Figure 2·9: Trajectories of two agents monitoring ten targets in a 3-D space. The assignments were determined using a three-stage look-ahead planner.

step. There are at least two factors that contribute to the diverse behavior of the agents. The first is that the decision dynamics are non-Markovian; that is, the choice of what to do next depends not only on where the agent is now but also on where it has recently visited. Second, the assignments of the two agents interact to affect the future choices made.

Other simulations with different constraints (different detection radii, different not re-visiting constraints, and so on) were also performed, though specifics on the simulations and detailed results are omitted for brevity. For example, imposing a constraint of not revisiting a target within two steps can lead to an agent cycling back and forth between two targets.

There are several factors that determine the computational complexity of the receding horizon algorithm. As expected, the cost is exponential in the number of look-ahead steps or the number of agents. More interesting is the dependence on the number of assignment sequences that need to be considered. In general, the cardinality of all sequences in the reward matrix is MN^{k_s}. However, not all of

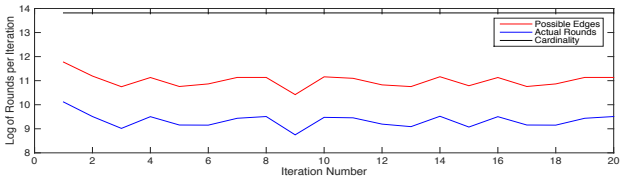


Figure 2·10: Comparing the numbers of all sequences combinations, all possible solutions, and the actual rounds needed to find the optimal one in every iteration.

these sequences are feasible since many targets are outside of the effective detection range of the agents. This is particularly important to note when the size of the workspace is increased to accommodate more targets since many targets would be too far away to fall into the planning horizon. This is illustrated in Fig. 2·10 which compares the number of all sequences (cardinality), the possible number of solutions, and the number we actually enumerated at each iteration to find the optimal one. The figure shows that we typically discard about 80% of the possible solutions before enumeration.

2.2.2 Dwell time optimization

Sec. 2.2.1.4 has shown part of the complexity of the approach while generating multiple visiting schedules together and allowing multiple agents to visit the same target, even not at the same time. Furthermore, while considering persistent monitoring in an infinite time horizon, an intuitive approach is to search for periodic schedules. Sharing targets may cause extra problems in different agents synchronizing their periods. It is a natural simplification to seek solutions where the target space is partitioned, and each target is only assigned to one agent.

In this subsection we assume that the N targets have been partitioned into M sets, each of which is assigned a single agent. Within each set, the sequence of target visits for the agent can be prescribed in at least two ways. First, the time frame can

be fixed and then all possible sequences that can be completed in this time window are considered; this approach is described in (Zhou et al., 2016). The second is to consider all sequences with a fixed number of visits. It is this latter one we focus on here. With the idea that the sequence will be repeated for persistent monitoring over the infinite time horizon, we consider all sequences where each target is visited exactly once per cycle with the agent returning to the first target at the end of the cycle to prepare for the next.

The main contribution of this subsection is the solution of the optimization problem defining the dwell time at each of the targets in a sequence. We show that a simple policy, namely leaving the target as soon as the uncertainty has been brought as low as possible, optimizes several related measures of performance. This involves three steps. First, in Sec. 2.2.2.1, the continuous scheduling problem is translated into a discrete one where the control input is the switching condition defining the uncertainty value at which an agent should depart the current target. Second, in Sec. 2.2.2.2 we show that under mild assumptions, any constant policy stabilizes the system to a steady state depending on that policy. Third, in Sec. 2.2.2.3 we determine the optimal policy with respect to the specified performance measures. With this policy in place for a given sequence, the cost of that sequence can be easily calculated, eliminating a computationally expensive step in the evaluation of the visiting sequences.

2.2.2.1 Discrete Dynamics

Similar as in Sec.2.1, we consider a collection of n targets and a single agent. A sequence is given such that every target is visited exactly once. We label the targets as $i = 1, ..., n$; a loop through the targets is a cycle. The location of target i is denoted as s_i and to each target there is an associated uncertainty state, defined below in (2.24) and denoted as $r_i(t)$ at time t. The agent can move between targets

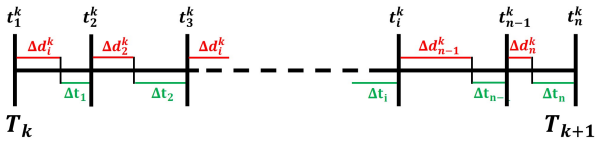


Figure 2·11: The time line of an agent. Δt_i is the transit time to get to the next target while Δd_i is the actual dwell time at that target.

with an average speed $\bar{v}$. We construct a discrete-time system by abstracting a cycle of visits completed by the agent into one step of a discrete-time system. The time when the agent begins its k^{th} visit at the first target is defined to be the beginning of the k^{th} step and is denoted as T_k. The sampled information is denoted as $R_i(k) = r_i(T_k)$. The duration of the k^{th} step is given by

$$T_{k+1} - T_k = \sum_{i=1}^{n} (\Delta d_i^k + \Delta t_i), \tag{2.21}$$

where Δd_i^k is the dwell time at the i^{th} target and Δt_i is the travel time from target i to $i+1$ with Δt_n the time to travel from target n back to target 1. These travel times are given by

$$\Delta t_i = \frac{\|s_{i+1} - s_i\|}{\bar{v}}, \quad \forall i = 1, ..., n, \quad s_{n+1} = s_1, \tag{2.22}$$

and the time of arrival of the agent to the i^{th} target in the k^{th} step is given by

$$t_i^k = T_k + \sum_{q=1}^{i-1} (\Delta d_q^k + \Delta t_q). \tag{2.23}$$

A graphical depiction of times in a cycle is shown in Fig.2·11.

The information state of target i is bounded below by 0, increases monotonically when it is not being visited, and decreases monotonically when the target is being

attended to by an agent. A natural model for this is

$$\dot{r_i}(t) = \begin{cases} -b_i & \text{if } \exists k \text{ s.t. } t - t_i^k \in [0, \Delta d_i^k); r_i(t) > 0 \\ 0 & \text{if } \exists k \text{ s.t. } t - t_i^k \in [0, \Delta d_i^k); r_i(t) = 0 \\ a_i & \text{otherwise.} \end{cases} \tag{2.24}$$

where a_i and b_i are both positive scalars. Notice that this model is related to (2.4), though here we assume a constant decrease rate of r_i rather than a changing rate over the distance between the target and the agent.

The dwell time Δd_i^k is determined by $r_i(t_i^k)$ and the switching condition provided for target i. The condition is defined such that the switching to the next target should be made at a specific time relative to the point at which the uncertainty drops to zero. This condition for target i in the k^{th} step is denoted as u_i^k, and its range is defined as follows:

$$u_i^k \in \left[-\frac{r_i(t_i^k)}{b_i}, \infty \right). \tag{2.25}$$

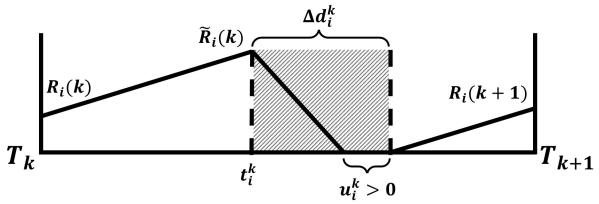
When u_i^k is negative, the physical meaning is that the switching should be made when $r_i = (-b_i) \cdot u_i^k$. Fig.2·12 illustrates the dynamics of r_i in a step k for both positive and negative vales of u_i^k.

Consider the uncertainty related to target i in this subsystem. From the beginning of step k it starts growing with rate a_i. Its value when the agent arrives at the target is given by

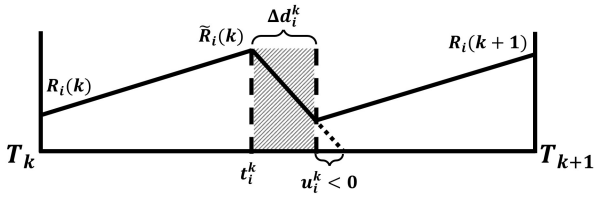
$$r_i(t_i^k) = R_i(k) + a_i \sum_{q=1}^{i-1} (\Delta d_q^k + \Delta t_q) \triangleq \tilde{R}_i(k) \tag{2.26}$$

where $R_i(k)$ is the initial value at the beginning of the period. We denote this value as $\tilde{R}_i(k)$. Note that it is the peak value for r_i during the agent's k^{th} visit of agent i.

Once the agent arrives, the uncertainty of the target decreases with a rate of $-b_i$ till the switching condition u_i^k is reached. The agent's actual dwell time at target i



(a) $u_i^k > 0$.



(b) $u_i^k < 0$.

Figure 2·12: Evolution of the uncertainty of target i in period k. The shaded section indicates the time an agent is visiting the target. (a) A positive value of u_i^k implies the agent remains with the target even after the uncertainty has been driven to zero while (b) a negative value implies the agent departs while the uncertainty is still positive.

can be written as

$$\Delta d_i^k = \frac{\tilde{R}_i(k)}{b_i} + u_i^k. \tag{2.27}$$

Δd_i^k is guaranteed to be non-negative though (2.25).

When the agent departs the target, the uncertainty is

$$r_i(t_i^k + \Delta d_i^k) = (-b_i)\bar{u}_i^k, \tag{2.28}$$

where we define $\bar{u}_i^k = \min(u_i^k, 0)$, and $\underline{u}_i^k = \max(u_i^k, 0)$. Note that this uncertainty level depends only on u_i^k and in particular is independent of both the initial value $R_i(k)$ and the peak value $\tilde{R}_i(k)$.

The uncertainty of target i at the beginning of next step is

$$\begin{aligned}
R_i(k+1) =&(-b_i)\bar{u}_i^k + a_i\Delta t_i + a_i \sum_{q=i+1}^{n} \left(\Delta t_q + \Delta d_q^k\right) \\
=&(-b_i)\bar{u}_i^k + a_i\Delta t_i \\
&+ a_i \sum_{q=i+1}^{n} \left(\Delta t_q + u_q^k + \frac{R_q(k)}{b_q}\right) \\
&+ a_i \sum_{q=i+1}^{n} \frac{a_q}{b_q} \sum_{l=1}^{q-1} \left(\Delta t_l + u_l^k + \frac{R_l(k)}{b_l}\right) \\
&+ a_i \sum_{q=i+1}^{n} \frac{a_q}{b_q} \sum_{l=1}^{q-1} \frac{a_l}{b_l} \sum_{m=1}^{l-1} \left(\Delta t_m + u_m^k + \frac{R_m(k)}{b_m}\right) \\
&+ \cdots .
\end{aligned} \tag{2.29}$$

Combining this for all the targets, the discrete dynamics of the uncertainty can be expressed as

$$\begin{aligned}
R(k+1) = BHB^{-1}R(k) + BHU(k) - B\bar{U}(k) + \\
(BH + A)D
\end{aligned} \tag{2.30}$$

where

$$\begin{aligned}
&B = \mathbf{diag}[b_1, ...b_n], \quad A = \mathbf{diag}[a_1, ...a_n], \\
&D = [\Delta t_1, ...\Delta t_n]^T, \quad R(k) = [R_1(k), ..., R_n(k)]^T, \\
&U(k) = [u_1^k, ..., u_n^k]^T,
\end{aligned}$$

and $H \in \mathbb{R}^{n\times n}$ is defined by

$$h_{ij} = \frac{a_i}{b_i}\left(\prod_{q=j+1}^{n}(\frac{a_q}{b_q}+1) - \prod_{q=j+1}^{i}(\frac{a_q}{b_q}+1)\right),$$

where, with a slight abuse of notation, $\prod_q^p(\cdot)$ is defined to be 1 when $p = q-1$ and 0

when $p < q - 1$. As an illustration, when $n = 2$, we have

$$H_2 = \begin{bmatrix} \frac{a_1 a_2}{b_1 b_2} & \frac{a_1}{b_1} \\ 0 & 0 \end{bmatrix},$$

and when $n = 3$,

$$H_3 = \begin{bmatrix} \frac{a_1 a_2}{b_1 b_2} + \frac{a_1 a_3}{b_1 b_3} + \frac{a_1 a_2 a_3}{b_1 b_2 b_3} & \frac{a_1}{b_1} + \frac{a_1 a_3}{b_1 b_3} & \frac{a_1}{b_1} \\ \frac{a_2 a_3}{b_2 b_3} + \frac{a_2^2 a_3}{b_2^2 b_3} & \frac{a_2 a_3}{b_2 b_3} & \frac{a_2}{b_2} \\ 0 & 0 & 0 \end{bmatrix}.$$

2.2.2.2 Steady state analysis

Sec. 2.2.2.1 defined the discrete dynamics of the uncertainty of the targets. We now analyze the system in (2.30) and show that for any given choice of constant control, there is an asymptotically stable equilibrium, beginning with the existence of the equilibrium.

Proposition 1. *Consider the discrete system* (2.30) *with a constant input* $U(k) \equiv [u_1, u_2, ..., u_n]$. *If* $b_i > (n-1)a_i$, $\forall i$, *then there exists a steady state solution* $\bar{R} = [\bar{R}_1, \bar{R}_2, ...\bar{R}_n]$ *that is independent of the initial states. Furthermore, every entry in* $B^{-1}\bar{R} + U$ *is non-negative.*

Proof. Let $R(k+1) = R(k) = \bar{R}$. From (2.30) we have

$$B^{-1}\bar{R} + U = (I - H)^{-1}\left(\underline{U} + (H + B^{-1}A)D\right). \tag{2.31}$$

Prop.1 holds if $(I - H)$ is invertible and $(I - H)^{-1}$ contains no negative entry. It can be shown after some algebra that the proposition holds if

$$\sum_{p=2}^{n-l+1} (p-1)\mathcal{S}_{l,n}^p < \prod_{i=l}^{n} b_i, \qquad \forall l = 1, ...n-1, \tag{2.32}$$

with $\mathcal{S}_{l,n}^p$ defined as

$$\mathcal{S}_{l,n}^p = \left(\prod_{i=l}^{n} b_i\right) \sum_{j_1 \neq ... \neq j_p} \left(\prod_{q=1}^{p} \frac{a_{j_q}}{b_{j_q}}\right). \tag{2.33}$$

Here $j_{q,\ q=1,...p}$ are p different indexes with $l \leq j_q \leq n$. For example, when $n = 2$, Prop.1 holds if

$$\sum_{p=2}^{2}(p-1)\mathcal{S}_{1,2}^{2} < \prod_{i=l}^{2} b_i, \quad \mathcal{S}_{1,2}^{2} = \left(\prod_{i=1}^{2} b_i\right)\frac{a_1 a_2}{b_1 b_2},$$

which simplifies to $a_1 a_2 < b_1 b_2$. For $n = 3$ Prop.1 holds if

$$\begin{aligned} l = 1: \quad & \mathcal{S}_{1,3}^{2} + 2\mathcal{S}_{1,3}^{3} < b_1 b_2 b_3, \text{ where} \\ & \mathcal{S}_{1,3}^{2} = a_1 a_2 b_3 + a_1 b_2 a_3 + b_1 a_2 a_3, \\ & \mathcal{S}_{1,3}^{3} = a_1 a_2 a_3; \\ l = 2: \quad & \mathcal{S}_{2,3}^{2} < b_2 b_3 \quad \text{with } \mathcal{S}_{2,3}^{2} = a_2 a_3. \end{aligned}$$

By assumption $b_i > (n-1)a_i,\ \forall i$. Thus

$$\mathcal{S}_{l,n}^{p} < \binom{n-l+1}{p}\frac{1}{(n-1)^p}\left(\prod_{i=l}^{n} b_i\right)$$

and therefore

$$\sum_{p=2}^{n-l+1}(p-1)\mathcal{S}_{l,n}^{p} < \left(\prod_{i=l}^{n} b_i\right)\sum_{p=2}^{n-l+1}\binom{n-l+1}{p}\frac{p-1}{(n-1)^p}.$$

We have

$$\sum_{p=2}^{n-l+1}\binom{n-l+1}{p}\frac{p-1}{(n-1)^p} \leq \sum_{p=2}^{n-l+1}\binom{n-l+1}{p}\frac{p-1}{(n-l)^p}.$$

The final expression equals to 1 and therefore (2.32) holds. □

Notice that the stability condition $b_i > (n-1)a_i$ in this proposition echoes a basic stability condition in queuing theory that the service rate should be greater than the arrival rate at each target.

The next proposition establishes the asymptotic stability of the equilibrium.

Proposition 2. *Consider the discrete system* (2.30) *with a constant input* $U(k) \equiv [u_1, u_2, ..., u_n]$. *If* $b_i > (n-1)a_i,\ \forall i$, *then* $\lim_{k\to\infty} R = \bar{R}$.

Proof. Let $E(k) = R(k) - \bar{R}$ be the error of $R(k)$ from the steady state $\bar{R}$ at step k. Then, from (2.30),

$$\begin{aligned} B^{-1}E(k+1) &= R(k+1) - \bar{R} \\ &= HB^{-1}(\bar{R} + E(k)) + HU(k) - \bar{U}(k) + (H + B^{-1}A)D \\ &= HB^{-1}E(k). \end{aligned}$$

Since B^{-1} is a diagonal matrix with its diagonal entries positive, $E(k)$ converges if and only if the eigenvalues of H satisfy $|\lambda_{H,i}| < 1$, $\forall i$.

Establishing this bound on the eigenvalues is similar to the analysis in the proof of Prop.1 and is omitted. □

Next we establish that the peak values of the uncertainties also converge to a steady state.

Proposition 3. *Consider the discrete system* (2.30) *with a constant input* $U(k) \equiv [u_1, u_2, ..., u_n]$. *If* $b_i > (n-1)a_i$, $\forall i$, *then the peak value of* $\tilde{R}(k)$ *converges to a fixed set of value* $\tilde{R}$ *that is independent of the initial states.*

Proof. According to (2.26) we have

$$\tilde{R}_i(k) = R_i(k) + a_i \sum_{q=1}^{i-1} \left(\Delta t_q + u_q + \frac{\tilde{R}_q(k)}{b_q} \right) \tag{2.34}$$

$$\begin{aligned} =& R_i(k) + a_i \sum_{q=1}^{i-1} \left(\Delta t_q + u_q + \frac{R_q(k)}{b_q} \right) \\ &+ a_i \sum_{q=1}^{i-1} \frac{a_q}{b_q} \sum_{l=1}^{q-1} \left(\Delta t_l + u_l + \frac{R_l(k)}{b_l} \right) \\ &+ \cdots . \end{aligned} \tag{2.35}$$

Prop.2 states that $R_i(k) \to \bar{R}_i$ with $k \to \infty$. Thus $\tilde{R}_i(k)$ converges to a constant value $\tilde{R}_i$. □

The value of the steady state for the peak values can be calculated as follows.

First, from (2.29) we have

$$\frac{\tilde{R}_i}{b_i} = -\left(\frac{a_i\tilde{R}_i}{b_i^2} + \frac{a_i u_i}{b_i} + \bar{u}_i\right) + \frac{a_i}{b_i}\sum_{q=1}^{n}\left(\Delta t_q + u_q^k + \frac{\bar{R}_q}{b_q}\right) + \frac{a_i}{b_i}\sum_{q=1}^{n}\frac{a_q}{b_q}\sum_{l=1}^{q-1}\left(\Delta t_l + u_l^k + \frac{\bar{R}_l}{b_l}\right) + \cdots .$$

Then

$$\tilde{R}_i = -\frac{b_i(a_i u_i + b_i\bar{u}_i)}{a_i + b_i} + \frac{a_i}{a_i + b_i}\sum_{q=1}^{n}\left(\Delta t_q + u_q^k + \frac{\bar{R}_q}{b_q}\right) + \frac{a_i}{a_i + b_i}\sum_{q=1}^{n}\frac{a_q}{b_q}\sum_{l=1}^{q-1}\left(\Delta t_l + u_l^k + \frac{\bar{R}_l}{b_l}\right) + \cdots .$$

In addition to the uncertainty states and their peak values, the duration of the time steps also converges.

Proposition 4. *Consider the discrete system* (2.30) *with a constant input* $U(k) \equiv [u_1, u_2, ..., u_n]$. *If* $b_i > (n-1)a_i$, $\forall i$, *then the period length* $T_{k+1} - T_k$ *converges to a fixed amount of time which is independent of the initial states.*

Proof. The period is given by

$$T_{k+1} - T_k = \sum_{i=1}^{n}\left(\frac{\tilde{R}_i}{b_i} + u_i + \Delta t_i\right). \tag{2.36}$$

Since u_i and Δt_i are constant values and $\frac{\tilde{R}_i}{b_i}$ converges $\forall i$, $T_{k+1} - T_k$ converges to a constant value. □

The final proposition in this section establishes that the average value of the uncertainty also converges. While this is perhaps clear from the previous results, the proof established the relationship between this average value and several other parameters.

Proposition 5. *Consider the discrete system* (2.30) *with a constant input* $U(k) \equiv [u_1, u_2, ..., u_n]$. *If* $b_i > (n-1)a_i$, $\forall i$, *then the average value* $\langle R_i \rangle = \frac{1}{T_{k+1}-T_k}\int_{T_k}^{T_{k+1}} r_i^k(t)dt$ *converges to a fixed value that is independent of the initial states.*

Proof. From (2.24), we have

$$\int_{T_k}^{T_{k+1}} r_i^k(t)dt = \frac{\tilde{R}_i^2(k)}{2a_ib_i} - (T_{k+1} - T_k)\, b_i\bar{u}_i. \tag{2.37}$$

The proposition then follows from Props. 3 and 4. □

2.2.2.3 Optimized Switching Conditions

In this section we search for the optimal switching policy U^* for the given sequence. There are several natural choices of performance, including the uncertainty values at the start of each period, the peak value during the period, the average value over the period, and the length of the period. Since our primary concern is with the infinite time horizon, we consider the steady state values of these terms. The previous section established that each of these converges to a steady value if the control is constant.

When $U \equiv 0$ is optimal.

The next proposition shows that for all but one of these performance measures, the control $U = 0$ is optimal.

Proposition 6. *Consider the discrete system* (2.30) *with a constant input* $U(k) \equiv [u_1, u_2, ..., u_n]$. *If* $b_i > (n-1)a_i$, $\forall i$, *then the sum of the steady state uncertainties,* $\sum_{i=1}^n \bar{R}_i$, *the sum of the peak steady state uncertainties,* $\sum_{i=1}^n \tilde{R}_i$, *and the period length* $T_{k+1} - T_k$ *are all minimized when* $u_i \equiv 0$.

Proof. From the proofs of Props 1, 3 and 4, the values of $\bar{R}_i$ and $\tilde{R}_i$ are monotonic functions of $|u_i|$. Therefore all these values reach their lowest point when $U(k) \equiv [0]^{n\times 1}$. Thus these are all minimized using the zero control.

Now consider the period length $T_{k+1} - T_k$ given in (2.36). Its value depends on u_i both explicitly and through $\tilde{R}_i$. Clearly, for positive u_i, the period length increases. For negative u_i, the contribution from the explicit term decreases but the contribution from $\tilde{R}_i$ increases by the same amount. Thus the period is minimized when $u_i \leq 0$ for all i. In particular it is minimized with the zero control. □

Thus the optimal choice of the switching condition is for the agent to leave its current target as soon as the uncertainty reaches zero. This result relies only on the fact that the uncertainty converges, as established in Sec. 2.2.2.2. In particular, it does not depend on the distance between the targets, their layout, the sequence of visiting them, or even their particular dynamics (outside of the convergence criterion). In the context of the larger problem of finding the best sequence, this result establishes that evaluating the cost of a given sequence is the calculation of a straightforward analytical expression and no longer requires the solution of a nonlinear optimization problem.

Optimal switching condition for $\langle R_i \rangle$.

The switching condition that optimizes the average value of the uncertainty is not as simple to determine. In general, the optimizing choice depends on the details of the targets. However, the next proposition establishes that under the special condition of a homogeneous set of targets, the zero control is once again optimal.

Proposition 7. *Consider the discrete system* (2.30) *with a constant input* $U(k) \equiv [u_1, u_2, ..., u_n]$ *and assume that* $a_i = a,\ b_i = b\ \forall\, i$. *If* $b > (n-1)a$, *then the sum of the average values over the steady state condition,* $\sum_{i=1}^{n} \langle R_i \rangle$, *is minimized when* $u_i \equiv 0$.

Proof. From Prop. 3 we have that

$$\tilde{R}_i = a_i \left(\sum_{q=1}^{n} (\Delta t_q + u_q + \frac{\tilde{R}_q}{b_q}) - u_i - \frac{\tilde{R}_i}{b_i} \right) - b_i \bar{u}_i.$$

Rearranging we get

$$\begin{aligned} \sum_{i=1}^{n} \frac{\tilde{R}_i}{b_i} = & \sum_{i=1}^{n} \frac{a_i}{b_i} \cdot \sum_{i=1}^{n} (\Delta t_i + u_i + \frac{\tilde{R}_i}{b_i}) \\ & - \sum_{i=1}^{n} \frac{a_i u_i}{b_i} - \sum_{i=1}^{n} \frac{a_i \tilde{R}_i}{b_i^2} - \sum_{i=1}^{n} \bar{u}_i. \end{aligned} \tag{2.38}$$

Since $a_i = a,\ b_i = b$, (2.38) can be simplified to

$$\sum_{i=1}^{n} \frac{\tilde{R}_i}{b_i} = \frac{na \sum_{i=1}^{n} \Delta t_i + (n-1)a \sum_{i=1}^{n} u_i - \sum_{i=1}^{n} \bar{u}_i}{b - (n-1)a}.$$

Therefore,

$$\begin{aligned}\tilde{R}_i =& a \left(\sum_{q=1}^{n} (\Delta t_q + u_q) - u_i - \frac{\tilde{R}_i}{b} \right) - b\bar{u}_i \\ &+ \frac{a^2 \sum_{q=1}^{n} (n\Delta t_q + (n-1)u_q) - ab \sum_{q=1}^{n} \bar{u}_q}{b - (n-1)a}.\end{aligned}$$

From this we get

$$\begin{aligned}\frac{\tilde{R}_i}{b} =& \frac{a}{a + b - na} \sum_{q=1}^{n} \left(\Delta t_q + \frac{b}{(a+b)} \underline{u}_q \right) \\ &- \frac{a u_i + b\bar{u}_i}{a+b}.\end{aligned} \tag{2.39}$$

We denote the peak uncertainty when the inputs are set to zero for all targets as $\tilde{R}_i^0$. According to (2.39) we have

$$\frac{\tilde{R}_i^0}{b} = \frac{a}{a + b - na} \sum_{q=1}^{n} \Delta t_q, \tag{2.40}$$

and according to (2.37) the average uncertainty over time subject to this set of inputs is

$$\langle R_i^0 \rangle = \frac{1}{2} \tilde{R}_i^0 = \frac{ab}{2(a + b - na)} \sum_{q=1}^{n} \Delta t_q. \tag{2.41}$$

Now consider an input sequence that is not identically zero. We start with analyzing one target among the group. For this selected target i, if we have $u_i = 0$, then regardless of the other inputs, we will always have

$$\langle R_i \rangle = \frac{1}{2} \tilde{R}_i = \frac{ab}{2(a + b - na)} \sum_{q=1}^{n} \left(\Delta t_q + \frac{b}{a+b} \underline{u}_q \right)$$

so that

$$\langle R_i\rangle - \langle R_i^0\rangle = \frac{ab^2}{2(a+b)(a+b-na)}\sum_{q=1}^{n}\underline{u}_q \geq 0. \tag{2.42}$$

Similarly, if $u_i < 0$, we have $\bar{u}_i = u_i$ and

$$\begin{aligned}\langle R_i\rangle =& \frac{1}{2}\tilde{R}_i - b\bar{u}_i \\ =& \frac{ab}{2(a+b-na)}\sum_{q=1}^{n}\left(\Delta t_q + \frac{b}{(a+b)}\underline{u}_q\right) - \frac{3bu_i}{2} \\ \geq& \langle R_i^0\rangle + \frac{ab^2}{2(a+b)(a+b-na)}\sum_{q=1}^{n}\underline{u}_q \geq \langle R_i^0\rangle\end{aligned} \tag{2.43}$$

regardless of the other inputs. The above results indicate that, for any target with a non-positive switching condition, no matter what value the other inputs shall be, the agent's performance with this specific target is no better than with the all-zero inputs. Furthermore, if a set of inputs contains no positive entries, we will have $\langle R_i\rangle \geq \langle R_i^0\rangle$ for every target, and thus $\sum_{i=1}^{n}\langle R_i\rangle \geq \sum_{i=1}^{n}\langle R_i^0\rangle$.

When an input set contains at least one positive entry, the situation becomes more complicated. We first consider a set of inputs with exactly one positive input, that is $u_i > 0$. For this specific target i we have

$$\begin{aligned}\langle R_i\rangle - \langle R_i^0\rangle &= \frac{(a+b)\tilde{R}_i^2}{2(a+b)\tilde{R}_i + 2abu_i} - \frac{1}{2}\tilde{R}_i^0 \\ &= \frac{(a+b)\tilde{R}_i(\tilde{R}_i - \tilde{R}_i^0) - abu_i\tilde{R}_i^0}{2(a+b)\tilde{R}_i + 2abu_i}.\end{aligned} \tag{2.44}$$

Consider now a different target, l. Since by assumption the i^{th} target is the only one with a positive value of the switching condition, we have $u_l \leq 0$. Then

$$\langle R_l\rangle = \langle R_l^0\rangle + \frac{ab^2}{2(a+b)(a+b-na)}\sum_{q=1}^{n}\underline{u}_q.$$

Under this scenario, every u_q with $q = 1, ..., n$ except for u_i is zero. Therefore

$\sum_{q=1}^{n} \underline{u}_q = u_i$. Then

$$\begin{aligned}
&\left(\langle R_i\rangle - \langle R_i^0\rangle\right) + \left(\langle R_l\rangle - \langle R_l^0\rangle\right) > 0 \\
&\Leftrightarrow ab^2u_i\left((a+b)\sum_{q=1}^{n}\Delta t_q + bu_i\right) \\
&\quad + ab\left(bu_i - (a+b-na)u_i\right)^2 \\
&\quad + ab(a+b)\sum_{q=1}^{n}\Delta t_q\left(bu_i - 2(a+b-na)u_i\right) > 0 \\
&\Leftrightarrow b^2u_i^2 + \left(bu_i - (a+b-na)u_i\right)^2 + (n-1)au_i > 0 \qquad (2.45)
\end{aligned}$$

and this last holds for $n > 1$. As stated above, (2.42) and (2.43) guarantee that $\langle R_q\rangle \geq \langle R_q^0\rangle$ for all other targets, $q \neq i, l$. Thus when $u_i > 0$ and $u_q \leq 0$ for all $q \neq i$ we have

$$\sum_{q=1}^{n}\langle R_q\rangle > \sum_{q=1}^{n}\langle R_q^0\rangle, \text{ s.t. } u_i > 0, \text{ and } u_q \leq 0, \ \forall q \neq i.$$

Finally we consider an input set with $z \geq 2$ positive entries. Without loss of generality we let $u_1, ..., u_z > 0$ and $u_{z+1}, ..., u_n \leq 0$. According to (2.44) we have

$$\begin{aligned}
&\sum_{i=1}^{z}\left(\langle R_i\rangle - \langle R_i^0\rangle\right) \geq 0 \qquad \Leftrightarrow \\
&(a+b)\sum_{q=1}^{n}\Delta t_q\sum_{i=1}^{z}\left(b\sum_{q=1}^{n}\underline{u}_q - 2(a+b-na)u_i\right) \geq 0. \qquad (2.46)
\end{aligned}$$

Since

$$\sum_{q=1}^{n}\underline{u}_q = \sum_{i=1}^{z}u_i,$$

(2.46) turns into

$$(a+b)\sum_{q=1}^{n}\Delta t_q\left(zb\sum_{i=1}^{z}u_i - 2(a+b-na)\sum_{i=1}^{z}u_i\right) \geq 0$$

which holds for $z \geq 2$. The remaining targets all have non-positive switching conditions so that their average uncertainty exceeds the value under an all-zero input set. Now we have shown that Prop. 7 holds for a set of inputs that contains multiple positive entries. Together with the conclusions above we have established Prop. 7.

□

Notice that under the zero policy, (2.41) leads to

$$\sum_{i=1}^{n} \langle R_i \rangle^* = \frac{nab}{2(a+b-na)} \sum_{q=1}^{n} \Delta t_q. \tag{2.47}$$

This in turn implies that for a single agent assigned to a homogeneous set of targets, the average uncertainty is minimized with the total traveling time, $\sum_{q=1}^{n} \Delta t_q$. The problem is then reduced exactly to a TSP. It is important to note that this does not hold when the targets are heterogeneous. It also does not imply anything about the more general multi-agent problem. In fact, (2.47) suggests that the number of targets assigned to an agent affects the cost of the sequences and therefore the weight of the edges in the graph formulation. However, in the multi-agent case with a homogeneous set of targets, the result in (2.47) can be used to determine the optimal partitioning of the targets to the agents.

Finally, we note that the constraint of a homogeneous set of targets is sufficient but not necessary. Following the proof of Prop.7 it can be shown that the same result holds if the homogeneity constraint is replaced by $\frac{b_i}{a_i} = k > n-1, \ \forall i$. However, the optimality of the zero switching condition may not hold if the targets are strongly heterogeneous. The optimal switching conditions can still be determined on a case by case basis but at the cost of numerical solution of the optimization problem.

2.2.3 Optimal schedule search

In this subsection we continue the work in Sec.2.2.2.3 by relaxing the constraint of requiring exactly one visit for each target in a given scheduling sequence for a single

agent. While this is a small change, it has a major complicating consequence. By relaxing the single visit requirement, even enumeration becomes impossible as the number of sequences to consider becomes infinite. To overcome this challenge, we first establish in Sec. 2.2.3.1 that the policy of staying with the current target until its information state is driven to zero and then immediately moving to the next target remains optimal under certain assumptions. With this choice of dwell time, the average state value of the targets in the cycle can then be calculated analytically. This allows us to compute in Sec. 2.2.3.2 an upper bound on the number of visits to any given target in a sequence for it to be optimal. This bound implies that the search of the optimal sequences can be reduced from infinite to a very large but finite number. Finally, in Sec. 2.2.3.3, we develop an algorithm similar to edge substitution algorithms for TSPs that adjusts a given sequence towards a better one to reduce the average state of the targets.

2.2.3.1 Problem formulation and Dwell time selection

Let us again consider a collection of n targets and a single agent in either $\mathbb{R}^2$ or $\mathbb{R}^3$. The targets are labeled by an index $i = 1, ..., n$. The visiting sequence of the agent is given by a sequence $\mathfrak{S}$ of $M_{\mathfrak{S}}$ elements, $\mathfrak{S} = [i_1, i_2, ...i_j, ..., i_{M_{\mathfrak{S}}}]$ with $i_j \in \{1, ..., n\}$, $i_j \neq i_{j+1}$ $j = 1, ..., M_{\mathfrak{S}} - 1$, $i_1 \neq i_{M_{\mathfrak{S}}}$. According to this sequence, each target i is visited m_i times. We require every target to be visited at least once, such that $m_i \geq 1$, and $M_{\mathfrak{S}} \geq n$. After completing one sequence, the agent returns to the first target listed to form a loop. If each of the targets is visited exactly once (as in Sec.2.2.2.3), the sequence becomes a solution candidate to a classical TSP problem. Here we refer to it as a *TSP sequence* and denoted as $\mathfrak{S}^{TSP}$. A $\mathfrak{S}^{TSP}$ contains $M_{\mathfrak{S}^{TSP}} = n$ entries.

Similar to Sec.2.2.2.3, each of the n targets carries some information that we are interested in. The information state r_i evolves as defined in (2.24) in Sec.2.2.2.3.

As stated in Sec.2.2.2.3, the agent moves between targets with a uniform speed.

Upon arriving at the j^{th} target in its sequence, it dwells for a duration Δd_j before leaving and traveling to its next scheduled target. The time line is as shown in Fig.2·11. For each visit in the sequence, the agent dwells at the scheduled target until a *switching condition* is met. The switching condition is denoted as $U = [u_1, ...u_j, ..., u_{M_{\mathfrak{S}}}]$ with each entry specifying an amount of time. If $u_j \leq 0$, the agent should stop its j^{th} visit and depart the current target $|u_j|$ units of time *before* the target state reaches zero. If $u_j > 0$, then the agent should linger at the target for an additional u_j units of time *after* the state reaches zero. Similar to Sec. 2.2.2.3, we set $\bar{u}_j = \min(u_j, 0)$, and $\underline{u}_j = \max(u_j, 0)$.

We again define $\tilde{R}_j(k) = r_{i,j}(t_j^k)$ to be the value of the information state of the j^{th} target in the sequence at the beginning of the visit in the k^{th} round of the cycle. Notice that the j^{th} target in a given visiting sequence $\mathfrak{S}$ is denoted by i_j. The dwell time at this target is denoted as Δd_j with the actual value depending on the value of the information state r_j and the switching condition u_j. In this section we also use T_k to be the beginning of the k^{th} cycle such that the period of the k^{th} cycle is $T_{k+1} - T_k$.

It was shown in Sec.2.2.2.3 that for TSP sequences, if the target dynamics satisfy $b > (n-1)a$, then for any given switching condition U, the $\tilde{R}_j(k)$ and period of the cycle converge to steady state values. Further, it was shown that the switching condition $U \equiv 0$, that is a policy that requires the agent to leave the current target as soon as the information state reaches 0, minimizes the steady state value of $\tilde{R}_j$, the period length, and the average information status of all targets $\langle R \rangle = \sum_{i=1}^{n} \langle R_i \rangle$. In the remainder of this section we examine whether the same results hold for a broader class of sequences that allow for multiple visits to the same target within a cycle.

We consider first the existence of a steady state. Clearly, for a steady state to exist in the information states, the agent must be able to effectively remove the increases in the information incurred while not attending to each of the targets on the cycle.

Assume target i is visited multiple times in a cycle and let $j_1, \ldots, j_{m_i}$ denote the indices in the cycle where this target is attended to by the agent. Given a policy U, the total dwell time at the target is given by

$$\sum_{q=1}^{m_i} \left(\frac{\tilde{R}_{j_q}}{b} + u_{j_q} \right).$$

During this time, the other $n-1$ targets are not being attended to and thus each receives an increase in its information of

$$\frac{a}{b} \sum_{q=1}^{m_i} \left(\tilde{R}_{j_q} + b u_{j_q} \right).$$

To remove all these increases requires a total of

$$\frac{(n-1)a}{b^2} \sum_{q=1}^{m_i} \left(\tilde{R}_{j_q} + b u_{j_q} \right)$$

time units. Of course, because the agent can only visit one target at a time, during the extra time to remove the extra information of the multiple visits, the information states of the non-visited targets will then increase by

$$\frac{(n-1)^2 a^2}{b^2} \sum_{q=1}^{m_i} \left(\tilde{R}_{j_q} + b u_{j_q} \right).$$

Continuing this calculation reveals that extra time needed is given by

$$\sum_{q=1}^{m_i} (\frac{\tilde{R}_{j_q}}{b} + u_{j_q}) \sum_{p=1}^{\infty} \left(\frac{(n-1)a}{b} \right)^p. \tag{2.48}$$

A steady cycle exists if and only if the increase and the decrease of the information states in a period are balanced. Calculating this balance at the beginning of the first

visit to target i in the k^{th} cycle, $t_{j_1}^k$, leads to the condition

$$\sum_{q=1}^{m_i}(\tilde{R}_{j_q}+b\bar{u}_{j_q}) = a\sum_{j=1}^{M_{\mathfrak{S}}}(\Delta t_j+u_j)+a\sum_{i=1}^{n}\frac{r_i(t_{j_1}^k)}{b}$$
$$+a\sum_{q=1}^{m_i}(\frac{\tilde{R}_{j_q}}{b}+u_{j_q})\left(\sum_{p=1}^{\infty}\left(\frac{(n-1)a}{b}\right)^p-1\right). \tag{2.49}$$

Satisfying this equation requires that (2.48) converges to a finite number. This is guaranteed if

$$-1<\frac{(n-1)a}{b}<1.$$

Given that $(n-1)a$ and b are both positive, we finally have that a steady state will exist if

$$b>(n-1)a.$$

Note that this is the same condition as in the single-visit sequences in the previous section.

We now consider the steady state period length. Since the target dynamics are homogeneous, we can simply consider the sum $\sum_{i=1}^{n} r_i$. As shown in Fig. 2·13, the sum of the states evolves under only three modes. When the agent is in transit between two targets, the overall state increases with the rate na. When the agent arrives at the target, the total state first decreases at a rate $b-(n-1)a$ until that state reaches zero. At that point the state increases at a rate $(n-1)a$ until the switching condition is met. In steady state, the summed state must not have a net change over one period. Therefore,

$$na\sum_{j=1}^{M_{\mathfrak{S}}}\Delta t_j+(n-1)a\sum_{j=1}^{M_{\mathfrak{S}}}\underline{u}_j=(a+b-na)\sum_{j=1}^{M_{\mathfrak{S}}}(\Delta d_j-\underline{u}_j).$$

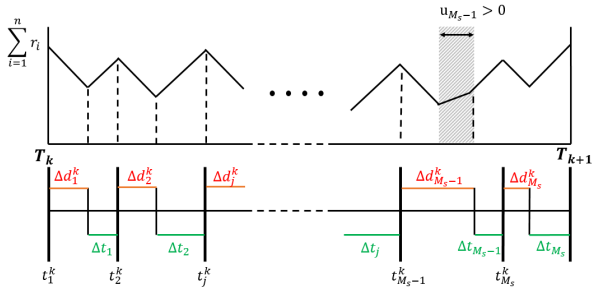


Figure 2·13: The dynamics of the sum of the information states together with the time line of the agent. During Δt_i the agent is in transit and the state increases. During Δd_i^k the state decreases until one of the target's state reaches zero. The state then increases more slowly until the switching condition is met and the agent leaves the current target.

Rearranging yields

$$\sum_{j=1}^{M_{\mathfrak{S}}} \Delta d_j = \sum_{j=1}^{M_{\mathfrak{S}}} \frac{na\Delta t_j + b\underline{u}_j}{a+b-na}.$$

Using this we calculate that the steady state period is given by

$$T_{k+1} - T_k = \sum_{j=1}^{M_{\mathfrak{S}}} (\Delta t_j + \Delta d_j) = \sum_{j=1}^{M_{\mathfrak{S}}} \frac{(a+b)\Delta t_j + b\underline{u}_j}{a+b-na}. \tag{2.50}$$

Recall that the Δt_j are the travel times between targets and are thus fixed for a given sequence $\mathfrak{S}$. By inspection, then, the period length is minimized by minimizing the $\underline{u}_j$ which can be achieved with $U \equiv 0$.

We now turn to the steady state value of $\tilde{R}_j$. From (2.49), we see that any $u_j < 0$ will result in a direct increase in $\tilde{R}_j$, while any $u_j > 0$ will allow other targets' state to increase over extra time, and finally increase the $\tilde{R}_j$ we are concerned about. Therefore $U \equiv 0$ minimizes this as well.

Finally, we consider the average state value $\langle R_i \rangle$. Assume that the target i is visited m_i times in one period. According to (2.50), the agent completes one period in $\sum_{j=1}^{M_{\mathfrak{S}}} \frac{(a+b)\Delta t_j}{a+b-na}$ units of time. Let the peak state value in each of the m_i visits be

denoted by $\tilde{R}_{j_q}, q = 1, ..., m_i$. Then, the dwell time of each visit is

$$\Delta d_{j_q} = \frac{\tilde{R}_{j_q}}{b}, \quad q = 1, ..., m_i.$$

The sum of all of these dwell times at target i takes a fraction of the total given by $\frac{a}{a+b}$. Under the zero policy, then, we can equate the total dwell time at that target with the fraction of the total time, leading to

$$\sum_{q=1}^{m_i} \frac{\tilde{R}_{j_q}}{b} = \sum_{j=1}^{M_{\mathfrak{S}}} \frac{a\Delta t_j}{a+b-na},$$

where we have used (2.50). Thus the average state value across all m_i visits is

$$\langle R_i \rangle = \frac{\sum_{q=1}^{m_i} (\tilde{R}_{j_q})^2}{2\sum_{q=1}^{m_i} \tilde{R}_{j_q}} \geq \frac{1}{2m_i} \sum_{q=1}^{m_i} \tilde{R}_{j_q}. \tag{2.51}$$

Hence the average information status of one target depends not only on the overall period time, but also the distribution of time among the multiple visits. Unfortunately, this implies that the zero switching policy is not guaranteed to optimize the overall average state as adding additional dwell times at other targets may shift the dwell times at target i, leading to a change in the average.

Despite this limitation, the switching policy $U \equiv 0$ still optimizes the value of the target states at the beginning of each of their visits, $\tilde{R}_j$ (which is equivalent to the peak value of the target state). It is also a simple policy and therefore, in the remainder of this paper we fix the switching policy to this one and then search for optimal sequences.

2.2.3.2 Bounding the number of visits

Our primary goal here is to show that under the switching policy $U \equiv 0$, there is an upper bound on the number of visits allowed for a sequence to be optimal.

Let us consider the average information state over all n targets being visited in a given sequence. The lower bound of the overall average state is a direct sum of (2.51) from every target which results in

$$\sum_{i=1}^{n} \langle R_i \rangle_{\mathfrak{S}} \geq \frac{\sum_{i=1}^{n} \frac{1}{m_i} ab}{2(a+b-na)} \sum_{q=1}^{M_{\mathfrak{S}}} \Delta t_q. \tag{2.52}$$

In this equation, the parameters a,b and n are fixed, while the sum of Δt_q is a function of the particular sequence. Increasing the number of visits at one of the targets increases the corresponding m_i (which counts the number of visits), and thus decreases $\frac{1}{m_i}$. However, it may also lead to an increase in the overall traveling time. There is thus a trade-off between these two terms. Based on this we can establish some simple rules for evaluating sequences.

Perhaps the simplest case is the one shown in Fig. 2·14 where an agent is assigned to move from i_2 to i_3 and, while making the move between these two targets, passes directly over the location of i_1. Clearly, incorporating i_1 into the sequence will not increase the transit time but will decrease the lower bound on the average by decreasing $\frac{1}{m_{i-1}}$ due to the increased number of visits to target i_1. Thus, adding a stop at i_1 *may* produce a lower overall average and thus cannot be ruled out.

Of course, when the targets are distributed in either $\mathbb{R}^2$ or $\mathbb{R}^3$, it is rare that three or more targets will lie on a line. A more practical assumption is that there is a target, i_q, located within a certain region around the agent's trajectory from i_a to i_{a+1}. This case will be discussed in Sec 2.2.3.3.

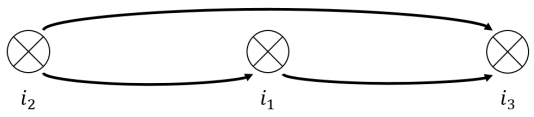


Figure 2·14: Simple example: adding visit to i_1 as the agent moves from i_2 to i_3 will not increase the traveling time.

In the remainder of this section, we establish several conditions for optimality (in

terms of the overall average) of a sequence by comparing it to another, simpler one, namely a TSP sequence. There are a variety of approximation algorithms to quickly find a suboptimal but still reasonable TSP solution; we therefore assume we have one such sequence and denote it as $\mathfrak{S}^{TSP}$. In order to easily compare other sequences to our reference, we introduce the notion of *average edge length*, denoted as $\langle \Delta t \rangle_{\mathfrak{S}}$, and defined by

$$\langle \Delta t \rangle_{\mathfrak{S}} = \frac{1}{M_{\mathfrak{S}}} \sum_{q=1}^{M_{\mathfrak{S}}} \Delta t_q. \tag{2.53}$$

Using this, the lower bound on the overall average value becomes

$$\begin{aligned} \sum_{i=1}^{n} \langle R_i \rangle_{\mathfrak{S}} &\geq \frac{\sum_{i=1}^{n} \frac{1}{m_i} ab}{2(a+b-na)} \left(M_{\mathfrak{S}} \langle \Delta t \rangle_{\mathfrak{S}} \right) \\ &= \left(\sum_{i=1}^{n} \frac{1}{m_i} \right) \left(\sum_{i=1}^{n} m_i \right) \frac{\langle \Delta t \rangle_{\mathfrak{S}} ab}{2(a+b-na)}. \end{aligned} \tag{2.54}$$

From here we can establish the following condition.

Lemma 8. *For any given sequence $\mathfrak{S}$, if there exists a TSP sequence $\mathfrak{S}^{TSP}$ such that*

$$\langle \Delta t \rangle_{\mathfrak{S}^{TSP}} \leq \langle \Delta t \rangle_{\mathfrak{S}},$$

then the sequence $\mathfrak{S}$ is not optimal.

Proof. The total number of visits in the sequence $\mathfrak{S}$ is given by $M_{\mathfrak{S}} = \sum_{i=1}^{n} m_i$. It is easy to show that

$$\left(\sum_{i=1}^{n} \frac{1}{m_i} \right) \left(\sum_{i=1}^{n} m_i \right) \geq \left(\sum_{i=1}^{n} \frac{n}{M_{\mathfrak{S}}} \right) \left(\sum_{i=1}^{n} \frac{M_{\mathfrak{S}}}{n} \right) = n^2.$$

Therefore

$$\begin{aligned}\sum_{i=1}^{n}\langle R_i\rangle_{\mathfrak{S}} &\geq \left(\sum_{i=1}^{n}\frac{n}{M_{\mathfrak{S}}}\right)\left(\sum_{i=1}^{n}\frac{M_{\mathfrak{S}}}{n}\right)\frac{\langle\Delta t\rangle_{\mathfrak{S}}ab}{2(a+b-na)}\\ &\geq \frac{n^2\langle\Delta t\rangle_{\mathfrak{S}^{TSP}}ab}{2(a+b-na)}\\ &= \frac{nab}{2(a+b-na)}\sum_{q=1}^{n}\Delta t_q\\ &= \sum_{i=1}^{n}\langle R_i\rangle_{\mathfrak{S}^{TSP}}.\end{aligned}$$

Thus, if the average edge length of $\mathfrak{S}$ is no smaller than the average edge length of a TSP sequence, then the overall average uncertainty under $\mathfrak{S}$ is greater than or equal to the overall uncertainty under the TSP sequence with equality only when $M_{\mathfrak{S}}$ is an integer multiple of n. □

Introducing an additional target visit into a sequence may reduce the average edge length. From (2.54), reducing the average edge length may reduce the overall average cost. However, it must reduce the average edge length enough to overcome the increase generated from the additional visit. With this in mind, we now establish a lower bound on the average edge length to help calculate an upper bound on the total number of visits. With a bound on the total number of visits, we will have reduced the number of sequences from an infinite set to a (possibly very large) finite set.

It is clear from (2.53) that for any physical configuration of targets, the average edge length, $\langle\Delta t\rangle_{\mathfrak{S}}$, of any sequence is larger than the shortest edge in the graph of that configuration. Let us denote the length of this shortest edge as Δt^* and define the ratio

$$K = \frac{\langle\Delta t\rangle_{\mathfrak{S}^{TSP}}}{\Delta t^*},$$

where $\mathfrak{S}^{TSP}$ is again our (simple) reference sequence. With this, we have

$$\langle \Delta t \rangle_{\mathfrak{S}} > \Delta t^* = \frac{1}{K} \langle \Delta t \rangle_{\mathfrak{S}^{TSP}}. \tag{2.55}$$

We will use this ratio to determine a bound on the total number of visits. Clearly, since the value of K depends on the reference sequence, the better the performance of the reference TSP sequence, the tighter the bound will be. We now establish several lemmas to lead up to the main result in Thm. 11.

Lemma 9. *Consider a collection of n targets and a sequence $\mathfrak{S}$ where the targets are visited $m_1, m_2, ..., m_n$ times. If*

$$\left(\sum_{i=1}^{n} \frac{1}{m_i} \right) \left(\sum_{i=1}^{n} m_i \right) > K n^2, \tag{2.56}$$

then this sequence is not optimal.

Proof. The proof is a straightforward calculation.

$$\begin{aligned} \sum_{i=1}^{n} \langle R_i \rangle_{\mathfrak{S}} &\geq \left(\sum_{i=1}^{n} \frac{1}{m_i} \right) \left(\sum_{i=1}^{n} m_i \right) \frac{\langle \Delta t \rangle_{\mathfrak{S}} ab}{2(a+b-na)} \\ &> K n^2 \frac{\frac{1}{K} \langle \Delta t \rangle_{\mathfrak{S}^{TSP}} ab}{2(a+b-na)} = \sum_{i=1}^{n} \langle R_i \rangle_{\mathfrak{S}^{TSP}} \end{aligned}$$

□

In the next lemma, we establish a lower bound of the overall average state generated by a sequence, under the condition that some of the targets have a pre-determined and fixed number of visits.

Lemma 10. *Consider a collection of n targets and any sequence $\mathfrak{S}$ with n_1 of the n targets visited exactly $m_1, ..., m_{n_1}$ times. Then the overall average uncertainty is bounded by*

$$\sum_{i=1}^{n} \langle R_i \rangle_{\mathfrak{S}} > \left((n - n_1) + \sqrt{\sum_{i=1}^{n_1} \frac{1}{m_i} \sum_{i=1}^{n_1} m_i} \right)^2 \frac{\Delta t^* ab}{2(a+b-na)}. \tag{2.57}$$

Proof. Let the total number of visits for the other $n - n_1$ targets be denoted as

$$M_0 = \sum_{i=n_1+1}^{n} m_i.$$

We then have

$$\begin{aligned}
&\left(\sum_{i=1}^{n} \frac{1}{m_i}\right)\left(\sum_{i=1}^{n} m_i\right) \\
&= \left(\sum_{i=1}^{n_1} \frac{1}{m_i} + \sum_{i=n_1+1}^{n} \frac{1}{m_i}\right)\left(\sum_{i=1}^{n_1} m_i + M_0\right) \\
&\geq \left(\sum_{i=1}^{n_1} \frac{1}{m_i} + \frac{(n-n_1)^2}{M_0}\right)\left(\sum_{i=1}^{n_1} m_i + M_0\right). \qquad (2.58)
\end{aligned}$$

The expression in (2.58) reaches its minimum at

$$M_0 = (n - n_1)\sqrt{\frac{\sum_{i=1}^{n_1} m_i}{\sum_{i=1}^{n_1} \frac{1}{m_i}}}. \qquad (2.59)$$

Therefore we have

$$\begin{aligned}
&\left(\sum_{i=1}^{n} \frac{1}{m_i}\right)\left(\sum_{i=1}^{n} m_i\right) \\
&\geq \left((n-n_1) + \sqrt{\sum_{i=1}^{n_1} \frac{1}{m_i} \sum_{i=1}^{n_1} m_i}\right)^2.
\end{aligned}$$

Using this expression in (2.54) together with (2.55) yields the desired result. □

Finally, we want to exclude the "trivial" modification to a given sequence defined by simply repeating the sequence and defining the combined repetitions as a single cycle. One way to avoid such a condition is to require that at least one of the targets be visited only once in the sequence. With this assumption, we can combine the previous lemmas to determine an upper bound on the number of visits to any other target in the optimal sequence.

Theorem 11. *Consider a collection of n targets and a sequence $\mathfrak{S}$ with at least one target visited exactly once. Then, a necessary condition for the sequence to be optimal is that each of the other $n-1$ targets should receive strictly less than $\left(\left(\sqrt{K}-1\right)n+2\right)^2$ visits.*

Proof. Without loss of generality, we assume the first target in the sequence is visited exactly once. Consider now a different target in the sequence that is visited m times. According to (2.57), the overall average uncertainty over the n targets is lower bounded by

$$\sum_{i=1}^{n}\langle R_i\rangle_{\mathfrak{S}} > \left((n-2)+\sqrt{m}+\frac{1}{\sqrt{m}}\right)^2 \frac{\Delta t^* ab}{2(a+b-na)}.$$

Using (2.55) to compare the overall average uncertainty of the sequence with the reference TSP sequence yields

$$\begin{aligned}\sum_{i=1}^{n}\langle R_i\rangle_{\mathfrak{S}} &> \left(n+\sqrt{m}+\frac{1}{\sqrt{m}}-2\right)^2 \frac{\langle\Delta t\rangle^*_{\mathfrak{S}^{TSP}} ab}{2K(a+b-na)} \\ &= \frac{\left(n+\sqrt{m}+\frac{1}{\sqrt{m}}-2\right)^2}{Kn^2}\sum_{i=1}^{n}\langle R_i\rangle_{\mathfrak{S}^{TSP}}. \end{aligned} \tag{2.60}$$

Now, the given sequence is clearly not optimal if the reference TSP sequence produces a lower average uncertainty. Thus, from (2.60), the given sequence cannot be optimal if

$$n+\sqrt{m}+\frac{1}{\sqrt{m}}-2 < \sqrt{K}n. \tag{2.61}$$

Since (2.61) holds if $m \geq \left(\left(\sqrt{K}-1\right)n+2\right)^2$, the theorem is established. □

2.2.3.3 Sequence refinement

Thm. 11 established that the optimal sequence can be found in a finite set. In practice, of course, this pool of candidates may still be so large that enumeration remains impractical except in very small settings. Thus, in this section we develop an algorithm for refining a given sequence by considering the insertion of a single

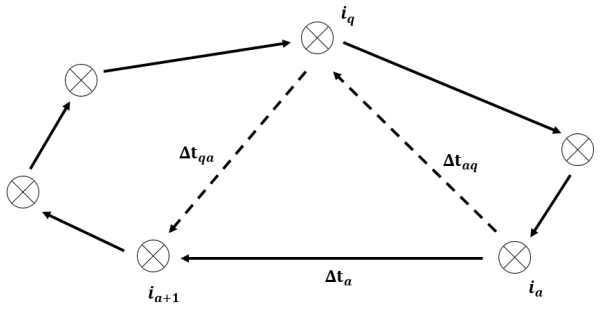


Figure 2·15: Introducing a single additional visit between two existing ones.

additional visit. Iterative application of the refinement can then move the sequence to increasingly better performance, at least until a local optimal is found. To avoid needing to consider all possible single insertions, we develop a quick search based on the physical layout of the targets.

Let the initial sequence be $\mathfrak{S} = \{i_1, i_2, \ldots, i_{M_\mathfrak{S}}\}$. The edges connecting the targets are denoted as $\Delta t_1, \Delta t_2, \ldots, \Delta t_{M_\mathfrak{S}}$. In this sequence, every target is visited $[m_1, \ldots, m_n]$ times. Inserting a visit to target q between targets i_a and i_{a+1} creates a new sequence where target q is now visited $m_q + 1$ times. Let the new edges introduced by the insertion be denoted as Δt_{aq} and Δt_{qa} and the new sequence as $\mathfrak{S}_q$; this setup is shown in Fig. 2·15. Note that inserting this visit implies the previous visit from i_a to i_{a+1} is removed.

From (2.51), the average state value of target i is given by

$$\langle R_i \rangle_\mathfrak{S} = \frac{\alpha_i ab}{2(a+b-na)} \sum_{j=1}^{M_\mathfrak{S}} \Delta t_j,$$

where α_i is a coefficient determined by the number of visits to target i, m_i, and the time lag between the visits with $\frac{1}{m_i} \leq \alpha_i \leq 1$. In particular, if $m_i = 1$, then $\alpha_i = 1$.

After inserting the additional visit, from (2.52) the average value for target q

(which is given the additional visit) becomes

$$\langle R_q \rangle_{\mathfrak{S}_\mathfrak{q}} \geq \frac{\frac{1}{m_q+1}ab}{2(a+b-na)} \sum_{j=1}^{M_{\mathfrak{S}_\mathfrak{q}}} \Delta t_j. \tag{2.62}$$

The overall average becomes

$$\sum_{i=1}^{n} \langle R_i \rangle_{\mathfrak{S}_\mathfrak{q}} \geq \frac{\left(\sum_{i=1}^{n} \alpha_i - \alpha_q + \frac{1}{m_q+1}\right) ab}{2(a+b-na)} \sum_{j=1}^{M_{\mathfrak{S}_\mathfrak{q}}} \Delta t_j.$$

The total traveling time around the new sequence is given by

$$\sum_{j=1}^{M_{\mathfrak{S}_\mathfrak{q}}} \Delta t_j = \sum_{j=1}^{M_{\mathfrak{S}}} \Delta t_j + \Delta t_{aq} + \Delta t_{qa} - \Delta t_a. \tag{2.63}$$

Therefore, in order for the new sequence to outperform the current sequence, we must have

$$\Delta t_{aq} + \Delta t_{qa} < \Delta t_a + \frac{(m_q+1)\alpha_q - 1}{(m_q+1)\sum_{i=1, i\neq q}^{n} \alpha_i + 1} \sum_{j=1}^{n} \Delta t_j. \tag{2.64}$$

The relationship in (2.64) provides a constraint that can be understood as an ellipse with i_a and i_{a+1} as the two foci. As shown in Fig. 2·16, if target q lies outside this ellipse, or the number of visits to this target exceeds $\left(\left(\sqrt{K}-1\right)n+2\right)^2 - 1$ (according to Thm. 11), then adding a visit to that target is guaranteed to *increase* the average cost and thus should not be considered. If the target lies inside the ellipse (target q' in Fig. 2·16), and the number of visits does not exceed the upper bound, then it *may* reduce the average cost and it should be tested.

Note that this test is based on the lower bound of the new sequence's average state value, and therefore should be considered as a conservative method. Consider, for example, the configuration shown in Fig. 2·17. Here, the ellipses generated by

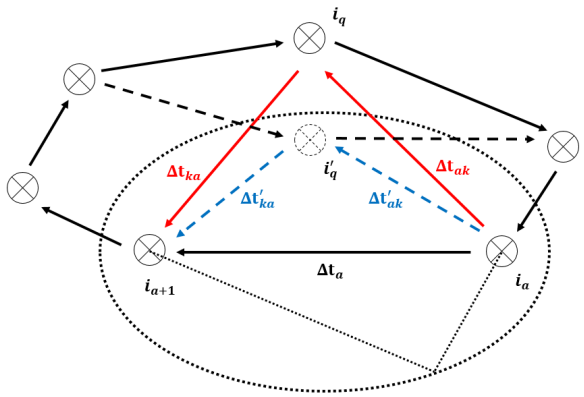


Figure 2·16: Evaluating possible targets for an additional visit between i_a and i_{a+1}. Target q lies outside the ellipse and a visit should not be added while target q' lies inside the ellipse and thus adding a visit *may* lead to a reduction in the total average information state.

(2.64) around each pair of targets do not intersect any other targets. Thus, adding any single visit will only increase the average information state. However, because this result is local, this does not imply that the sequence is optimal.

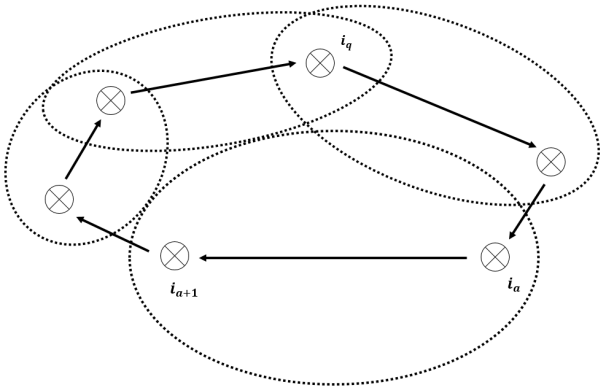


Figure 2·17: In this configuration, the ellipses from (2.64) at each pair of targets do not contain any other targets and thus adding any single additional visit will increase the average information state.

Chapter 3

Scheduling of Multiple Agents in a Persistent Monitoring Task Using a Reachability Analysis

In this chapter, we build upon the results of our earlier work in (Yu and Andersson, 2013; Yu and Andersson, 2014) to study the question of reachability of a linear dynamic system under a periodically-connected control and with constraints on the control authority and in the presence of a disturbance. After formulating the problem in Sec. 3.1, we focus in Sec. 3.2 on the unconstrained case and establish conditions on the periodic schedules that cause a loss of reachability for a given system and show how to alter such a schedule to regain reachability. In Sec. 3.3 we bring back the control constraints and adopt the notion of a recovery region introduced in (Viswanathan et al., 1984), extending it to the discrete-time setting considered here. We develop a method for calculating the size of this recovery region. By taking the ratio of the size of the recovery region to the size of the "escape" region , we establish a measure of the degree of reachability. Finally, we use this to determine the optimal number of connected periods for a periodic sequence.

3.1 Problem Formulation

Consider a group of $\mathcal{N}$ targets, each located in physical (2-D or 3-D) space and with a state evolving in $\mathbb{R}^n$ according to

$$x(k+1) = Ax(k) + Bu(k) + d(k), \tag{3.1}$$

where the input $u(k) \in \mathbb{R}^m$ and disturbance $d(k) \in \mathbb{R}^n$ satisfy

$$u(k) \in \mathfrak{U},\ \mathfrak{U} = \left\{ u \in \mathbb{R}^m \middle| |u_i| \leq 1, i = 1, \ldots, m \right\},$$
$$\|d(k)\|_2 \leq \delta,$$

with $\|\cdot\|_2$ indicating the standard Euclidean norm and δ a positive constant. We assume that A is invertible and that the pair $[A, B]$ is reachable. As described below, the control is applied to the target by an agent. The need for the invertibility of A is related to the intermittent nature of the control arising from the limited duration of a visit by an agent and is elaborated on briefly later in this section. Note that the targets may each have different system matrices and bounds. In the sequel, we develop our analysis from the point of view of one of the $\mathcal{N}$ targets and thus omit any target index on the dynamics in (3.1) to avoid cluttering the notation.

There are $\mathcal{M}$ homogeneous agents that move in the physical space to visit the targets. When visiting a target, an agent dwells for some time to apply control to the system before departing. We assume that the agents move in such a way that each target sees a periodic sequence of visits of period p. Inside this period, the target sees interleaved "disconnected" and "connected" stages defined by whether an agent is present and applying control or not. The i^{th} disconnected and connected stages have a duration of r_i and q_i time steps, respectively. We define $\bar{r} = (r_1, r_2 \ldots)$, $\bar{q} = (q_1, q_2, \ldots)$, and refer to this visiting sequence as a $(p, \bar{q}, \bar{r})$ policy. With $r =$

$r_1 + r_2 + \cdots$ and $q = q_1 + q_2 + \cdots$, we clearly have $p = q + r$. For concreteness and without loss of generality, we assume that the period starts with a disconnected stage. Note that our definition of a $(p, \bar{q}, \bar{r})$ policy is analogous to a communication sequence in (Zhang and Hristu-Varsakelis, 2006). The need for invertibility of A follows from this view of a $(p, \bar{q}, \bar{r})$ policy as it is shown in (Zhang and Hristu-Varsakelis, 2006) that lack of this property can lead to loss of reachability.

We will consider two problems under this general scenario. In each case we begin with a simple $(p, \bar{q}, \bar{r})$ policy with a single disconnected stage of length r followed by a single connected stage of length q before extending results to more general policies with multiple disconnected and connected stages.

In the first problem, addressed in Sec. 3.2, we explore under what conditions the reachability of the target is maintained under a $(p, \bar{q}, \bar{r})$ policy. To do so, we will drop the bound on the control input and initially assume no disturbance. With these results in hand, we reintroduce the disturbance. Because the disturbance is *a priori* unknown, reachability under the fully connected scenario (where there is always an agent at the target) implies only the ability to bring the state to within a ball of radius δ of the desired target state. Under a $(p, \bar{q}, \bar{r})$ policy, there are portions of the period where the system is uncontrolled and we explore the effect of this setting on the size of the final ball around the desired target state, using the notion of *lifting* (Bamieh et al., 1991) to address the periodic nature of the $(p, \bar{q}, \bar{r})$ policy.

In the second problem, developed in Sec. 3.3, we reintroduce the bounds on the control signal and focus on bringing the target state back to the origin. To ensure control is needed, we assume A is unstable. Because the system is unstable, both the drift term and the disturbance tend to drive the system away from the origin. Over time and in the absence of control, the possible location of the system's state lies within an expanding domain. Of course the control can counteract this expansion

and work to hold the system near the origin. In our setting, however, its effectiveness is limited by two factors: the bounds on the control signal and the $(p, \bar{q}, \bar{r})$ policy which forces the control actions to be applied for only a portion of the period p. This combination leads to a limited domain that can be brought back to the origin within the q connected steps. Reachability is now defined as the expanding domain of possible state locations lying within the domain that can be recovered to the origin. Using lifting once again, we show that it is possible to design controls for a reachable system that keep it near the origin and explore the effect of the control bounds and the $(p, \bar{q}, \bar{r})$ policy on the size of region the system can be stabilized to.

3.2 Degree of Reachability

We now address the first problem described in Sec. 3.1, namely the effect of a $(p, \bar{q}, \bar{r})$ policy on the reachability of the target system (3.1). Throughout this section we relax the constraint on the control, allowing $u(k) \in \mathbb{R}^m$. We begin by ignoring the disturbance, focusing on the impact of the intermittent control. After establishing when the target system retains the property of reachability (and how it can be regained when it is lost), we extend the results to include the disturbance.

While the model for the target dynamics in (3.1) are linear and time-invariant (LTI), the introduction of intermittent but periodically-applied control through the $(p, \bar{q}, \bar{r})$ policy leads to a linear, time-varying system. Because the policy is periodic, we apply the idea of "lifting" (Bamieh et al., 1991) to transform the system back into an LTI one that incorporates the $(p, \bar{q}, \bar{r})$ policy. To do so, assume for now that there is one connected and one disconnected period such that $\bar{q} = q$ and $\bar{r} = r$ and, to distinguish it from the more general case, denote it as a (p, q, r) policy. Define the

new state

$$\hat{x}(k) = x(kp).$$

The dynamics in this state are given by

$$\begin{aligned}
\hat{x}(k+1) &= x(kp+p) \\
&= A^p x(kp) + A^{p-1}Bu(kp) + A^{p-2}Bu(kp+1) + \\
&\quad \cdots + A^r Bu(kp+q-1) + A^{p-1}d(kp) + \\
&\quad + A^{p-2}d(kp+1) + \cdots + d(kp+p-1) \\
&= A^p \hat{x}(k) + \\
&\quad \begin{bmatrix} A^{p-1}B & \cdots & A^{p-q}B \end{bmatrix} \begin{pmatrix} u(kp) \\ u(kp+1) \\ \cdots \\ u(kp+q-1) \end{pmatrix} + \\
&\quad \begin{bmatrix} A^{p-1} & \cdots & I \end{bmatrix} \begin{pmatrix} d(kp) \\ d(kp+1) \\ \cdots \\ d(kp+p-1) \end{pmatrix}.
\end{aligned} \tag{3.2}$$

To simplify this, define

$$\begin{aligned}
\hat{u}(k) &= \begin{bmatrix} u(kp) & u(kp+1) & \cdots & u(kp+q-1) \end{bmatrix}^T, \\
\hat{D}(k) &= \begin{bmatrix} A^{p-1} & \cdots & I \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} d(kp) & d(kp+1) & \cdots & d(kp+p-1) \end{bmatrix}^T, \\
\hat{A} &= A^p, \\
\hat{B} &= \begin{bmatrix} A^{p-1}B & A^{p-2}B & \cdots & A^r B \end{bmatrix},
\end{aligned} \tag{3.3}$$

where $r = p - q$ is the number of time steps in the period where no control is applied. With this, (3.2) can be written

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}\hat{u}(k) + \hat{D}(k). \tag{3.4}$$

Note that k refers to a single step in the lifted system as well as to the number of periods p that have elapsed in the original system.

Ignoring the disturbance, we focus on reachability of the system

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}\hat{u}(k). \tag{3.5}$$

which is the lifted version of

$$x(k+1) = Ax(k) + Bu(k) \tag{3.6}$$

evolving under a (p, q, r) policy.

The reachability matrix of (3.5) is

$$\mathfrak{R}_p = \begin{bmatrix} \hat{B} & \hat{A}\hat{B} & \hat{A}^2\hat{B} & \dots \end{bmatrix}. \tag{3.7}$$

Expressing this over the first k communication periods and using (3.3) to write it in terms of the original system matrices yields

$$\begin{aligned} \mathfrak{R}_{kp} = [&A^r[B, AB, ..., A^{q-1}B], A^{p+r}[B, AB, ..., A^{q-1}B], \\ &...A^{(k-1)p+r}[B, AB, ..., A^{q-1}B]]. \end{aligned}$$

This can be written concisely as

$$\mathfrak{R}_{kp} = A^r\bar{\mathfrak{R}}_{kp}B \tag{3.8}$$

by defining the $n \times kqm$ matrix

$$\bar{\mathfrak{R}}_{kp} = \left[[I, A, ..., A^{q-1}], A^p[I, ..., A^{q-1}], ..., A^{(k-1)p}[I, ..., A^{q-1}]\right]. \tag{3.9}$$

If there exist n linearly independent columns in this matrix, then the system is reachable within k periods.

Recall that the original system (3.1) is reachable and let l denote the number of steps for the reachability matrix of (3.1) to achieve full rank. It is clear, then, that (3.5) is reachable for $q \geq l$ and we therefore consider only $q < l$.

The lifted system (3.4) is LTI and consequently its reachability can be determined using the Hautus test which states that an LTI MIMO system (A, B) is reachable if and only if

$$\text{rank}[\lambda I - A \| B] = n, \, \forall \, \lambda \in \mathbb{C},$$

where A is the state matrix and B is the input matrix(Hautus, 1970).

Lemma 12. *Consider system* (3.6) *with invertible* $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ *and with* (A, B) *a reachable pair. Suppose the controller and plant are connected under a* (p, q, r) *communication policy with* $q < l$, $p \geq l$. *Then the system will preserve its reachability if and only if* $\forall \lambda \in \mathbb{C}$,

$$rank\left[\lambda I - A^p \| \ \left[A^r B \ A^{r+1} B \ \ldots \ A^{r+q-1} B\right]\right] = n.$$

Proof. The condition in the lemma is equivalent to

$$\text{rank}\left[\lambda I - \hat{A} \| \ \hat{B}\right] = n.$$

The Hautus test then applies directly, yielding the result.

□

In the following subsections, we will establish specific conditions under which SISO and MIMO systems lose reachability. For SISO systems we establish a necessary and sufficient condition, while in MIMO systems a sufficient condition is provided. In both cases we first limit to a (p, q, r) policy and then extend the results to the more general $(p, \bar{q}, \bar{r})$ setting in which the target is visited multiple times by (possibly) multiple agents inside one period.

3.2.1 Periodic communication on a SISO plant

In (Yu and Andersson, 2013) we considered a SISO system under a $(p, q = 1, r)$ policy and showed that reachability will be lost if and only if there are repeated eigenvalues in A^{r+1}. We now seek to establish a similar result for a policy with arbitrary q.

In order to get a result involving only the eigenvalues, assume for the moment that the system matrix A is diagonalizable. Let $T \in \mathbb{C}^{n\times n}$ be an invertible matrix such that

$$\tilde{A} = T\hat{A}T^{-1}$$

is a diagonal matrix. Perform a similarity transform $\tilde{x} = T\hat{x}$ to arrive at the new system

$$\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}\hat{u}(k) \tag{3.10}$$

where

$$\tilde{B} = T\hat{B} = \begin{bmatrix} TA^{p-1}B & TA^{p-2}B & \cdots & TA^{p-q}B \end{bmatrix}.$$

Clearly, by similarity, checking the reachability of (3.6) is equivalent to checking the reachability of (3.10).

The matrix $\tilde{A}$ is diagonal with eigenvalues given by those of A^p. Since (A, B) is a reachable pair (and the system is SISO), the q columns in $[A^{p-1}B \quad A^{p-2}B \quad \cdots \quad A^{p-q}B]$ are all linearly independent. The rank of $\hat{B}$ (and of $\tilde{B}$) is therefore q. To establish reachability by the Hautus test, then, the rank of the matrix $\lambda I - \tilde{A}$ must be at least $n - q$ for all $\lambda \in \mathbb{C}$.

Since A is invertible, the rank of $\tilde{A}$ is n. Thus the rank of $\lambda I - \tilde{A}$ is n for any λ not equal to an eigenvalue of $\tilde{A}$. If λ is an eigenvalue with multiplicity β, then the

rank of $\lambda I - \tilde{A}$ drops by β. Thus if $\tilde{A}$ has an eigenvalue with a multiplicity greater than q, the rank of $\lambda I - \tilde{A}$ is less than $n - q$ and the rank of $[\lambda I - \tilde{A} \| \tilde{B}]$ must be less than n.

Consider now a system with an A that is not diagonalizable. There exists a similarity transformation $T \in \mathbb{C}^{n \times n}$ that maps A into a Jordan form J_A. The same operator can also map $\hat{A}$ into a Jordan form $J_{\hat{A}} = T\hat{A}T^{-1}$.

The above discussion leads to the following lemma for a SISO plant connected under a (p, q, r) policy.

Lemma 13. *Consider system* (3.6) *with* $A \in \mathbb{R}^{n \times n}$*,* A *invertible,* $B \in \mathbb{R}^{n \times 1}$ *and with* (A, B) *a reachable pair. Suppose the controller and plant are connected under a* (p, q, r) *policy with* $q < l$*,* $p \geq l$*. Then the system will lose reachability if and only if* A *is diagonalizable and* A^p *has at least one eigenvalue with a multiplicity strictly greater than* q*.*

Proof. We first consider the diagonalizable case and show the necessity of the condition for losing reachability. From the discussion above, if all eigenvalues of A^p have a multiplicity of q or less, then

$$\operatorname{rank}\left[\lambda I - \tilde{A}\right] \geq n - q.$$

We now show that the columns of $\tilde{B}$ are linearly independent of the columns of $\lambda I - \tilde{A}$. Notice first that since (A, B) is a reachable pair, the pair (TAT^{-1}, TB) is also reachable. Then, by the Hautus test, the n columns of

$$\left[\lambda I - TAT^{-1} \| TB\right]$$

are linearly independent for all $\lambda \in \mathbb{C}$. Since the matrix TAT^{-1} is diagonal, every entry in the vector TB must be non-zero to prevent the combined matrix above from dropping rank when λ is an eigenvalue of A.

Every column in $\tilde{B}$ is constructed by multiplying TB by a diagonal matrix $TA^{p-s}T^{-1}$, $(s = 1, 2, \ldots, q)$. Since TB has no zero entries and since all the diagonal elements of $TA^{p-s}T^{-1}$ are nonzero (since A is invertible), all the columns of $\tilde{B}$ contain non-zero entries and they remain independent of $\lambda I - \tilde{A}$ when λ matches one of the eigenvalues of A^p. Then the rank of

$$\left[\lambda I - \tilde{A} \| \tilde{B}\right]$$

must be n and, by the Hautus test, the system (3.10) is reachable. By similarity, the system (3.4) is also reachable and thus, by Lemma 12 the original system (3.6) under the (p, q, r) policy is as well.

We now establish sufficiency in the diagonalizable case. Suppose then that there is at least one eigenvalue of A^p with multiplicity strictly greater than q; denote this eigenvalue as λ_q. The matrix $[\lambda I - \hat{A} \| \hat{B}]$ contains $n + q$ columns. Let $\lambda = \lambda_q$. Then at least $q + 1$ columns of $\lambda I - \tilde{A}$ will be zero and the rank of $[\lambda I - \hat{A} \| \hat{B}]$ must be less than n. By the Hautus test, the system (3.10) is not reachable. By similarity, the system (3.4) is also not reachable and thus, by Lemma 12 the original system (3.6) under the (p, q, r) policy is not reachable as well.

Consider now the non-diagonalizable case. Since $\tilde{A}$ is in Jordan normal form, any selection of λ could turn at most one of the columns in $\lambda I - \tilde{A}$ into zero. We have

$$\operatorname{rank}\left[\lambda I - \tilde{A}\right] \geq n - 1.$$

We represent J_A with π distinguished eigenvalues as

$$J_A = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_\pi \end{bmatrix};$$

and TB as

$$TB = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_\pi \end{bmatrix},$$

such that each sub-matrix β_i in TB carries the same number of rows as J_i, for $i \in \{1...\pi\}$. As we know

$$\operatorname{rank}\left[\lambda I - J_A \| TB\right] = n.$$

Thus, from the structure of the Jordan form, all the entries in the last row of each b_i must be non-zero, for $i \in \{1...\pi\}$.

We have

$$\tilde{B} = T\hat{B} = \left[TA^{p-1}B \;\; TA^{p-2}B \;\; \cdots \;\; TA^{p-q}B\right],$$

with A invertible. By construction, every column in $T\tilde{B}$ has non-zero entries at those positions where TB has non-zero entries. Therefore each of the columns is linearly independent of all the columns in $\left[\lambda I - \tilde{A}\right]$. This implies that if A is non-diagonalizable, then

$$\left[\lambda I - \tilde{A} \| \tilde{B}\right] = n$$

always holds.

□

The results in Lemmas 12 and 13 can be easily extended to the case of a $(p, \bar{q}, \bar{r})$ policy that consists of multiple connected and disconnected stages corresponding to multiple visits to the target by (possibly) different agents. The construction of the system in (3.4) is easily extended to the multiple q_i and r_i case through the obvious redefinition of $\hat{A}$ and $\hat{B}$. Under such a policy, the reachability matrix of the extended plant can be written as

$$\begin{aligned}\mathfrak{R}_p = &[A^{r_1}, ..., A^{q_1+r_1-1}, A^{r_1+q_1+r_2}, \\ &..., A^{r_1+q_1+r_2+q_2-1}...]B.\end{aligned} \tag{3.11}$$

As with the case of a single connected stage, this reachability matrix has qm columns. Lemma 12 can be directly extended to this setting as follows.

Lemma 14. *Consider system* (3.6) *with* $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$ *and with* (A, B) *a reachable pair. Suppose the controller and plant are connected under a* $(p, \bar{q}, \bar{r})$ *policy with* $q = \sum_i q_i$. *Assume* $q < l$, $p \geq l$. *Then the system will preserve its reachability if and only if* $\forall \lambda \in \mathbb{C}$

$$rank\,[\lambda I - A^p \| \mathfrak{R}_p] = n.$$

Proof. The proof is based on the Hautus test and follows directly the proof of Lemma 12. □

If the system of (3.6) is SISO, the $(p, \bar{q}, \bar{r})$ policy provides q columns to $\mathfrak{R}_p$ in one period p. However, the q steps are not sequential (the blocks being separated by r_i disconnected steps), and therefore, while the q_i columns in the corresponding block of $\mathfrak{R}_p$ are linearly independent of each other, there is no guarantee that the columns in one block are linearly independent of those in another. As a result, a general $(p, \bar{q}, \bar{r})$ policy provides $q^* \leq q$ linearly independent columns in $\mathfrak{R}_p$.

With this we can extend the results based on repeated eigenvalues as follows.

Lemma 15. *Consider system* (3.6) *with* $A \in \mathbb{R}^{n\times n}$*,* A *invertible,* $B \in \mathbb{R}^{n\times 1}$ *and with* (A, B) *a reachable pair. Suppose the controller and plant are connected under a* $(p, \bar{q}, \bar{r})$ *policy. Then the system will lose reachability if and only if* A *is diagonalizable and* A^p *has at least one eigenvalue with a multiplicity strictly greater than* q^**.*

Proof. By definition, $\hat{B}$ provides q^* independent columns. The proof then follows analogously to Lemma 13 with q^* playing the role of q. □

Since q^* must be determined from the reachability matrix while q is simply defined by the policy, it is easier to consider q directly. The fact that $q^* \leq q$ leads to the following corollaries.

Corollary 16. *Consider system* (3.6) *with* $A \in \mathbb{R}^{n\times n}$*,* A *invertible,* $B \in \mathbb{R}^{n\times 1}$ *and with* (A, B) *a reachable pair. Suppose the controller and plant are connected under a* $(p, \bar{q}, \bar{r})$ *policy with* $q < l$*,* $p \geq l$*. Then the system will lose reachability if* A *is diagonalizable and* A^p *has at least one eigenvalue with a multiplicity strictly greater than* q*.*

Corollary 17. *Consider system* (3.6) *with* $A \in \mathbb{R}^{n\times n}$*,* A *invertible,* $B \in \mathbb{R}^{n\times 1}$ *and with* (A, B) *a reachable pair. Suppose the controller and plant are connected under a* $(p, \bar{q}, \bar{r})$ *policy with* $q < l$*,* $p \geq l$*. Then the system will preserve its reachability if* A *is diagonalizable and* A^p *has no eigenvalue with a multiplicity strictly greater than* $\max(q_1, q_2, ...)$*.*

3.2.2 Periodic communication on a MIMO plant

In this section we extend the results of Sec.3.2.1 to MIMO systems connected under a general $(p, \bar{q}, \bar{r})$ policy (which includes the special case of a (p, q, r) policy).

Lemma 13 established a condition for SISO systems under a (p, q, r) policy to maintain reachability that was based only on the multiplicity of the eigenvalues in A^p. Unlike Lemma 12, this result relied strongly on the SISO setting to establish that the columns of $\hat{B}$ were linearly independent of those of $\lambda I - \hat{A}$. In the MIMO setting, of course, B has multiple columns. For any given column b_i, $i = 1, \ldots, m$, it is still true that if b_i is linearly independent of the columns of $\lambda I - A$ for all $\lambda \in \mathbb{C}$, then it will be linearly independent of the columns in $\lambda I - \hat{A}$ for all $\lambda \in \mathbb{C}$ as well. As before, it follows that the collection of vectors $A^{p-1}b_i, A^{p-2}b_i, ..., A^r b_i$ are also linearly independent of the columns in $\lambda I - \hat{A}$. From this we can conclude that there is at least one column in each block of $[A^{p-1}B \| A^{p-2}B \| ... \| A^r B]$ that is linearly independent of the columns in $\lambda I - \hat{A}$.

While the above discussion gets us near to proving a result similar to that of Lemma 13, there are unfortunately additional ways in the MIMO setting under which the Hautus test matrix can fail to be full rank. Perhaps the most straightforward of these is that the columns in each of the blocks of $[A^{p-1}B \| A^{p-2}B \| \cdots \| A^r B]$ are not necessarily linearly independent of one another. In general, the span of those columns may have a non-empty intersection with the span of the columns of $\lambda I - \hat{A}$.

As a result, one does not expect a simple necessary and sufficient test such as was found for SISO systems. We instead pursue a sufficient condition only. The following lemma establishes that if no eigenvalues of the state matrix increase in multiplicity when the system is connected with a $(p, \bar{q}, \bar{r})$ policy, then reachability is preserved.

Lemma 18. *Consider system* (3.6) *with invertible* $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ *and with* (A, B) *a reachable pair. Suppose the controller and plant are connected under a* $(p, \bar{q}, \bar{r})$ *policy with* $q < l$, $p \geq l$. *Let* $V_A = \{v_1, ..., v_n\}$ *be the collection of eigenvalues*

of A (the state matrix from the original system) and let $V_{A^p} = \{v_1^p, ..., v_n^p\}$ be the collection of eigenvalues of $\hat{A} = A^p$. If

$$v_i^p \neq v_j^p \, \forall \, i, j \text{ such that } v_i \neq v_j,$$

then the system connected under a $(p, \bar{q}, \bar{r})$ policy will be reachable for any values of q_i satisfying $\sum_i q_i = q$.

Proof. To prove this lemma we assume that the system uses only one connected step in the period p, independent of the actual values of $\bar{q}$. The addition of additional connected steps may lead to reachability being achieved in fewer steps but will not alter the fact that reachability is preserved. We thus assume we have a $(p, q = 1, r)$ policy.

The system can be lifted into an LTI version of the form in (3.4) with $\hat{A} = A^p$ and $\hat{B} = A^{p-1}B$ where, as previously, the beginning of the period is defined to be the start of the disconnected stage. As the reachability of the lifted system is independent of where we define the beginning of a period, however, we are free to consider any equivalent lifted system where the period begins at a different point. For convenience, then, in this proof we set the beginning of the period to coincide with the connected step. The lifted LTI system then has $\hat{A} = A^p$ and $\hat{B} = B$.

We first consider a system with diagonalizable A. As before, similarity of the systems implies that establishing reachability can be done using the diagonalized system. Thus, without loss of generality we assume the system matrix is diagonal. Recall that the Hautus test says that the system under a $(p, 1, r)$ policy will be reachable if and only if

$$\text{rank}\,[\lambda I - A^p \| B] = n, \;\; \forall \, \lambda \in \mathbb{C}.$$

Since the original system is, by assumption, reachable, we have

$$\text{rank}\,[\lambda I - A \| B] = n$$

for all $\lambda \in \mathbb{C}$ and in particular for $\lambda \in V_A$. Consider now the matrix

$$[\lambda^p I - A^p \| B]\,.$$

Since A^p is invertible, this matrix is guaranteed to have rank n except possibly when

$\lambda^p \in V_{A^p}$. Set $\lambda = v_i$ Then the j^{th} column of $\lambda I - A$ and of $\lambda^p I - A^p$ become

$$a_j = (0, ..., 0, (v_i - v_j), 0, ..., 0)^T,$$
$$a_j^p = \left(0, ..., 0, (v_i^p - v_j^p), 0, ..., 0\right)^T.$$

By the assumption in the Lemma, the column a_j^p is all zero only if the column a_j is. Further, we can write

$$(v_i^p - v_j^p) = c(v_i - v_j),$$

where c is a constant defined by a bivariate polynomial in v_i and v_j. Since v_i and v_j are fixed, this implies that each column of $\lambda_i^p I - A^p$ is a scalar multiple of the corresponding column of $\lambda_i I - A$. Thus the matrices $[\lambda I - A \| B]$ and $[\lambda^p I - A^p \| B]$ have exactly the same linearly independent columns. Since the original system was reachable, this yields

$$\text{rank}\,[\lambda I - A^p \| B] = n \tag{3.12}$$

for all $\lambda \in \mathbb{C}$.

For non-diagonalizable systems, we use the Jordan form of the system matrix instead of the diagonalized form in the proof above. An analysis following the same steps as above shows that (3.12) also holds for non-diagonalizable systems. Thus the condition in the Lemma ensures the Hautus test holds and this establishes the Lemma.

□

The above results allow one to test whether a particular $(p, \bar{q}, \bar{r})$ policy will lead to loss of reachability. If reachability is preserved, finding a controller for the system to achieve stability or other desired objectives can be achieved using standard algorithms (see, e.g. (Zhang and Hristu-Varsakelis, 2006), (Yu and Andersson, 2013)). We now briefly discuss how to regain reachability when it is lost (allowing the same algorithms to be used). We focus on the general MIMO setting and a basic (p, q, r) policy; as before extension to the more general case is straightforward.

3.2.3 Regaining reachability

When a given (p, q, r) policy fails to preserve a system's reachability, there are two options for regaining it. The first is to modify the system matrix to ensure the conditions of Lemma 18. In practice, of course, it can be difficult or impossible to change the system dynamics. The second option is to adjust the (p, q, r) policy. While the system may enforce a lower bound on the delay, it is reasonable to expect that it is possible to add delay. Perhaps surprisingly, this can lead to regaining reachability.

Lemma 19. *Consider system* (3.6) *with invertible* $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ *and with* (A, B) *a reachable pair. Suppose the controller and plant are connected under a* (p, q, r) *policy with* $q < l$, $p \geq l$ *and suppose further that, under this policy the systems loses its reachability. Then there exists a new policy with the same* q *but with extended delay* $\hat{r} > r$ *such that under this policy the system's reachability is preserved.*

Proof. The conditions for sufficiency of losing reachability or observability discussed in Lemma 18 are based on the repeated eigenvalues in the system matrix A^p. Since there are a finite number of eigenvalues to test, there are a finite number of integer periods will lead to a loss of reachability or observability. Let these integers be called 'critical periods'. Let $\hat{p}$ be the least common multiple of the currently known critical periods. Then $A^{\hat{p}+1}$ does not satisfy any of the conditions in 18 and a $(\hat{p}+1, q, \hat{r})$ policy with $\hat{r} = \hat{p} + 1 - q$ will preserve the properties for the original system. □

We note that this is a sufficient condition; there may be an $\hat{r}$ shorter than the one based on the least common multiple of $\hat{p}$ such that the system regains reachability.

3.2.4 Impact of the disturbance

Given that the system (3.4) is reachable for a given $(p, \bar{q}, \bar{r})$ policy, the presence of the unknown but bounded disturbance simply implies that the system can be driven only to a bounded domain containing the desired state. After one period p, the size of

this domain is determined by the form of the disturbance in the lifted state, namely

$$\hat{D}(k) = \begin{bmatrix} A^{p-1} & \cdots & I \end{bmatrix} \begin{pmatrix} d(kp) \\ d(kp+1) \\ \cdots \\ d(kp+p-1) \end{pmatrix} \tag{3.13}$$
$$= \sum_{i=1}^{p} A^{p-i} d(kp+i-1).$$

A worse-case upper bound can be established by assuming the disturbance takes its maximum magnitude δ at each step. Let λ^* denote the maximum magnitude eigenvalue of A. Then the state after p steps is guaranteed to lie inside a ball of radius

$$\sum_{i=1}^{p} (|\lambda^*|)^{i-1} \delta \tag{3.14}$$

centered on the desired state. Eqn. (3.14) demonstrates (the perhaps obvious fact) that longer periods amplify the effect of the disturbance. Thus, while it may be necessary to extend the period in order to regain reachability of the system under a given $(p, \bar{q}, \bar{r})$ policy, this comes at the cost of poorer performance in terms of the size of the disturbance ball.

3.3 Reachability Analysis with Disturbance Rejection

In Sec. 3.2, we ignored the constraint on the control input. We now bring back that limitation and investigate its impact on an appropriate notion of reachability. As before, the system dynamics are described by (3.1) under a given $(p, \bar{q}, \bar{r})$ policy. We also follow the same sequence as before, beginning with a simple (p, q, r) policy (that is, one with a single connected and single disconnected stage) before extending to the more general $(p, \bar{q}, \bar{r})$ setting.

In Sec. 3.3.1 we ignore the disturbance, focusing on the reachability of the sys-

tem under the control constraint. We measure this reachability using an approach introduced in (Viswanathan et al., 1984) based on the concept of a *recovery region*, a subset of the state space such that every element in the subset can be brought back to the origin in a fixed, finite time window. We develop a method to calculate a lower bound on the size of this set and then, in Sec. 3.3.3, reintroduce the disturbance to analyze the competing effects of control and disturbance to determine whether a given system will remain reachable.

3.3.1 Measurement of reachability

We will start with the definition of recovery region for a system with bounded input.

Definition 3.3.1. *For a system of the form* (3.1) *with* $u \in \mathfrak{U}$*, the recovery region over* q *steps is defined to be the set*

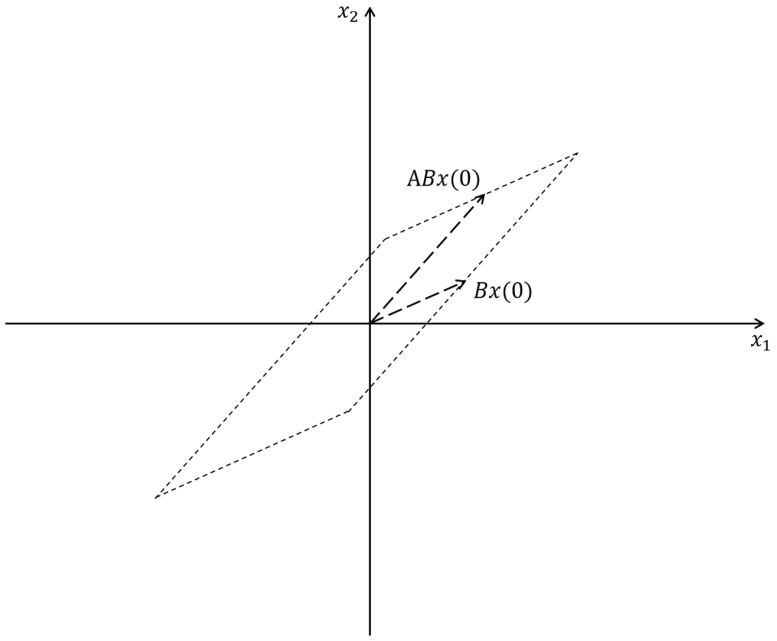
$$\begin{aligned}\mathfrak{S}(q) = \{x \in \mathbb{R}^n \,|\, \exists\, u(0), \cdots, u(q-1), u(k) \in \mathfrak{U},\\ k = 0, ..., q-1, \ \textit{steering } x(0) = x, \textit{to } x(q) = 0\}.\end{aligned}$$

Note that since the bound on the control is symmetric about the origin, this region is equivalent to the usual notion of the reachable region from the origin after q steps.

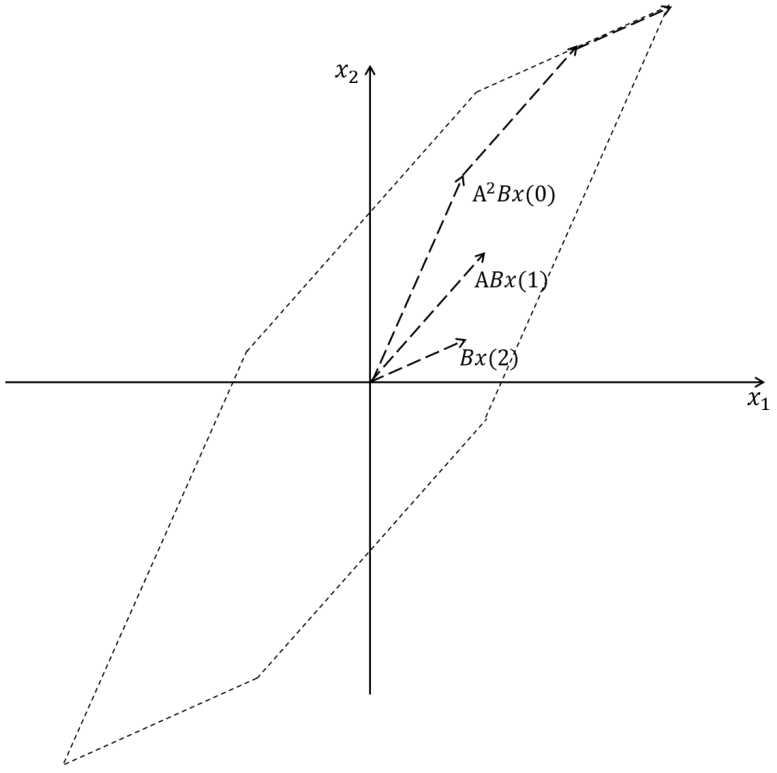
This recovery region is an expanding convex polytope and is illustrated in Fig. 3·1 for a two dimensional system with single input after two and after three steps. We use the radius of the largest inscribed ball of this polytope as the measurement of this system's reachability and refer to this as the *recovery distance*, $\rho_r^*(q)$. It is a function of the number of time steps and is given by

$$\begin{aligned}\rho_r^*(q) = \max_{x \in \mathfrak{S}(q)} \|x\| \qquad (3.15)\\ \text{such that } \forall y \in \mathbb{R}^n, \ \text{if } \|y\| \leq \rho_r^*(q), \ \text{then } y \in \mathfrak{S}(q).\end{aligned}$$

While (3.15) makes clear the notion of the recovery distance, it is not particularly useful for calculating its value. In the sequel, we develop a method for calculating



(a) Recovery region after two steps.



(b) Recovery region after three steps.

Figure 3·1: Recovery region of a two dimensional SISO system with bounded input.

$\rho_r^*(q)$ for discrete-time systems.

Given a finite $\rho_r^*(q)$, there is at least one initial condition with norm larger than this recovery distance from which the system cannot be brought back to the origin. This then implies that there is (at least) one direction such that any initial condition in this direction with norm greater than $\rho_r^*(q)$ cannot be brought to the origin in q steps. As illustrated in Fig. 3·2, we denote this *critical direction* with the vector v_c (choosing one arbitrarily if there are multiple possibilities) and set $\|v_c\| = \rho_r^*(q)$. Our calculation of $\rho_r^*(q)$ proceeds by considering the effect of the inputs over all q steps on this direction.

Geometrically, v_c points to where the inscribed ball is tangent to a facet of the convex polytope. This facet is a co-dimension one linear subspace and as such is spanned by $n-1$ linearly independent directions. By construction of the recovery region, these directions must be from those generated by the system over the q steps. Collecting these directions as columns of a single $n \times mq$ matrix, we define

$$\mathfrak{D}(q) = \{ \; B \quad AB \quad \cdots \quad A^{q-1}B \; \}. \tag{3.16}$$

Recalling that the input is bounded with $u_{(k)} \in \mathcal{U}$, we have

$$\rho_r^*(q) = \sum_{j=1}^{q} \sum_{l=1}^{m} \|P_{v_c}(A^{j-1}b_l)\|,$$

where P_{v_c} is the projection operator onto the direction of v_c.

Clearly, if there are no more than $n-1$ linearly independent directions in $\mathfrak{D}(q)$, then the system is not reachable over q steps and $\rho_r^*(q)$ equals to zero. If, on the other hand, $\mathfrak{D}(q)$ is full rank, then the facet that is orthogonal to the critical direction v_c is spanned by some choice of $n-1$ columns of the matrix.

Let $\mathfrak{I}(q)$ denote the collection of all choices of $n-1$ columns of $\mathfrak{D}_q$. Since $\mathfrak{D}_q$ has mq columns, there are $\binom{mq}{n-1}$ different subsets in $\mathfrak{I}(q)$. Let $\mathfrak{I}(q)_j$ denote the j^{th} entry

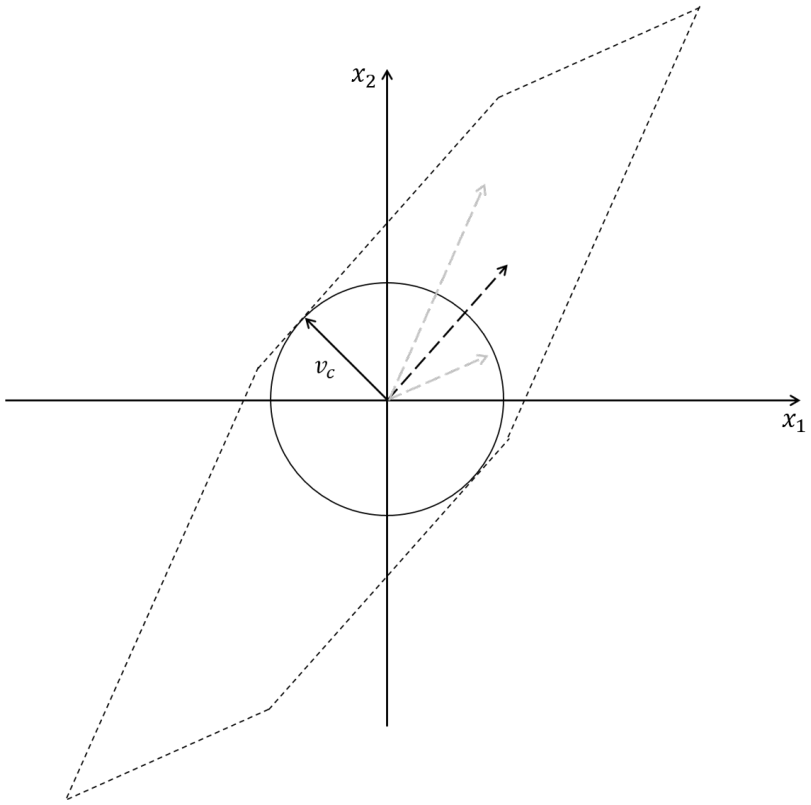


Figure 3·2: The inscribed ball of the recovery region and the critical direction v_c. v_c is paralleled to one of the input vectors.

of the collection and let $\hat{v}_k$ denote the unit vector that is orthogonal to all entries of the collection. Then the recovery distance can be found by

$$\rho_r^*(q) = \min_{\mathfrak{I}_k(q)} \sum_{j=1}^{q} \sum_{l=1}^{m} \|P_{v_k}(A^{j-1}b_l)\|. \tag{3.17}$$

With this notion of reachability, we turn our attention back to system (3.1) under a given (p, q, r) communication policy. Using the lifting method described in Sec. 3.2 yields the system

$$\bar{x}(k+1) = A^p\bar{x}(k) + \bar{B}\bar{u}(k)$$

where $\bar{B} = [B, AB, ..., A^{q-1}B]$. Recall that it is assumed that A (and thus A^p) is

unstable. Let us focus on one step of this lifted system (that is, on one period p) and consider how the recovery distance changes as a function of the number of connected steps q.

For the sake of simplicity, we focus on state matrices that are diagonalizable over the field of reals. In addition, the mathematical arguments are simplest in the two-dimensional SISO setting and thus we present them in that context, extending to the more general case in Sec. 3.3.2.

Consider therefore a system with

$$A = \begin{bmatrix} (1+\alpha)\lambda & 0 \\ 0 & \lambda \end{bmatrix}, \quad B = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix},$$

with $|\lambda| \geq 1$ and $\alpha > 0$ and where β_1 and β_2 are arbitrary non-zero real numbers. (Note that zero is not allowed because (A, B) is a controllable pair.) The lifted system with a (p, q, r) policy $(q > 2)$, yields

$$\mathfrak{D} = \left\{ \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}, \begin{bmatrix} (1+\alpha)\lambda\beta_1 \\ \lambda\beta_2 \end{bmatrix}, ..., \begin{bmatrix} (1+\alpha)^{q-1}\lambda^{q-1}\beta_1 \\ \lambda^{q-1}\beta_2 \end{bmatrix} \right\}.$$

The critical direction v_c is orthogonal to one of the vectors in $\mathfrak{D}$. For the k-th vector in $\mathfrak{D}$, an orthogonal vector is given by

$$v_k = \begin{bmatrix} \beta_2 \\ -(1+\alpha)^{k-1}\beta_1 \end{bmatrix}.$$

The magnitude of the projection of any vector in $\mathfrak{D}$ on the direction of v_k can be calculated to be

$$\begin{aligned} \|P_{v_k}(\mathfrak{D}_j)\| &= \left| \frac{\mathfrak{D}_j \cdot v_k}{\|v_k\|} \right| \\ &= \frac{|\left((1+\alpha)^{j-1} - (1+\alpha)^{k-1}\right)\lambda^{j-1}\beta_1\beta_2|}{\sqrt{(1+\alpha)^{2(k-1)}\beta_1^2 + \beta_2^2}}, \end{aligned}$$

and the sum of projections of the inputs over the q steps is

$$\rho_r^k(q) = \sum_{j=1}^{q} \|P_{v_k}(\mathfrak{D}_j)\|. \tag{3.18}$$

A different selection of k yields different v_k and therefore alters the sum value. Using these results in (3.17) and denoting the index that minimizes the sum of these projections as k_q, we get

$$\rho_r^*(q) = \sum_{j=1}^{q} \frac{|\left((1+\alpha)^{j-1} - (1+\alpha)^{k_q-1}\right)\lambda^{j-1}\beta_1\beta_2|}{\sqrt{(1+\alpha)^{2(k_q-1)}\beta_1^2 + \beta_2^2}}. \tag{3.19}$$

Equation (3.19) describes a monotone increasing sequence indexed by q. Unfortunately, because there is no analytical formula for k_q, we cannot determine an analytical transition rule from $\rho_r^*(q)$ to $\rho_r^*(q+1)$. Note that (3.18) actually defines a collection of sequences in q with each sequence distinguished by choice of k (which may itself be a function of q). Below we consider a specific choice, namely the one given by setting $k = q$ to yield $\rho_r^q(q)$. We show that this sequence dominates the one we are actually interested in that defines the recovery distance, $\rho_r^*(q)$ where, as illustrated in Fig.3·3, a dominating sequence is one that is larger at every index. The figure also shows that at small q, the behavior of $\rho_r^*(q)$ is non-trivial while that of $\rho_r^q(q)$ is nicely exponential, yet at large q both approach a simple exponential. Analysis of the simpler, dominating sequence will provide insight into the rate of growth of the recovery distance.

We begin by establishing that the optimizing k, that is k_q, cannot be unity.

Lemma 20. *For any $q > 1$, $\rho_r^1(q) > \rho_r^*(q)$ always holds.*

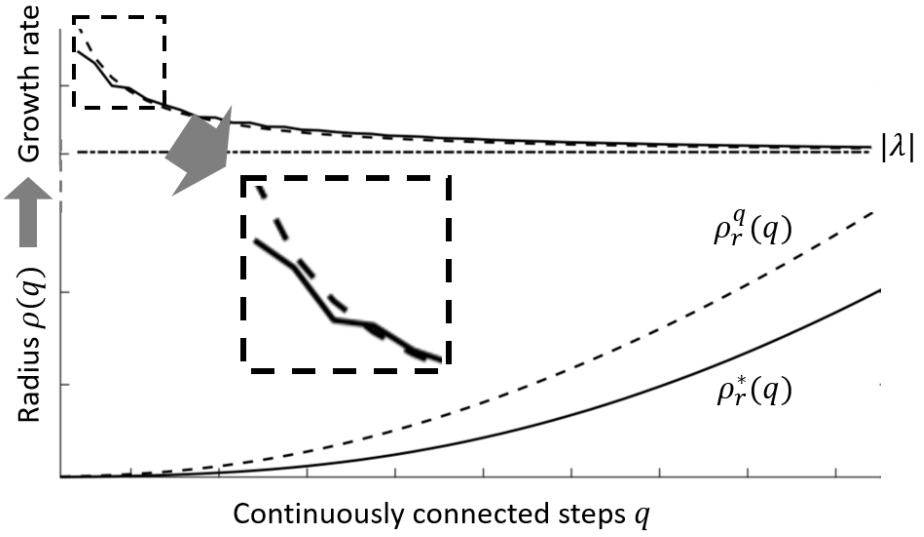


Figure 3·3: The growth rate of the sequence $\rho_r^*(q)$ may not be analytical, while a dominating sequence $\rho_r^q(q)$ (shown in dash lines) is easier to analyze.

Proof. We simply compare $\rho_r^1(q)$ and $\rho_r^2(q)$

$$\rho_r^1(q) = \sum_{j=1}^{q} \frac{|\left((1+\alpha)^{j-1}-1\right)\lambda^{j-1}\beta_1\beta_2|}{\sqrt{\beta_1^2+\beta_2^2}},$$
$$\rho_r^2(q) = \sum_{j=1}^{q} \frac{|\left((1+\alpha)^{j-1}-(1+\alpha)\right)\lambda^{j-1}\beta_1\beta_2|}{\sqrt{(1+\alpha)^2\beta_1^2+\beta_2^2}}.$$

It is clear from inspection that $\rho_r^2 < \rho_r^1$. □

It is clear that Lemma 20 implies that $k_q - 1 > 0$. We will take advantage of this in later proofs.

We now establish a lower bound on the relative size of two sequential elements in one of these sequences.

Theorem 21. *For any k, we have*

$$\frac{\rho_r^{k+1}(q+1)}{\rho_r^k(q)} > |\lambda|.$$

Proof. Writing out the q^{th} term in the sequence ρ_r^k yields

$$\rho_r^k(q) = \sum_{j=1}^{q} \frac{|\left((1+\alpha)^{j-1} - (1+\alpha)^{k-1}\right)\lambda^{j-1}\beta_1\beta_2|}{\sqrt{(1+\alpha)^{2(k-1)}\beta_1^2 + \beta_2^2}}, \tag{3.20}$$

while the $(q+1)^{st}$ term in ρ_r^{k+1} is

$$\begin{aligned}\rho_r^{k+1}(q+1) &= \sum_{j=1}^{q+1} \frac{|\left((1+\alpha)^{j-1} - (1+\alpha)^{k}\right)\lambda^{j-1}\beta_1\beta_2|}{\sqrt{(1+\alpha)^{2k}\beta_1^2 + \beta_2^2}} \\ &= \sum_{j=2}^{q+1} \frac{|\left((1+\alpha)^{j-2} - (1+\alpha)^{k-1}\right)\lambda^{j-1}\beta_1\beta_2|}{\sqrt{(1+\alpha)^{2(k-1)}\beta_1^2 + \left(\frac{\beta_2}{1+\alpha}\right)^2}} \qquad (3.21a) \\ &\quad + \frac{|\left((1+\alpha)^{k} - 1\right)\lambda^{j-1}\beta_1\beta_2|}{\sqrt{(1+\alpha)^{2k}\beta_1^2 + \beta_2^2}}, \qquad (3.21b)\end{aligned}$$

Note that the portion (3.21b) is positive and that (3.21a) can be rewritten as

$$\sum_{j=1}^{q} \frac{|\left((1+\alpha)^{j-1} - (1+\alpha)^{k-1}\right)\lambda^{j}\beta_1\beta_2|}{\sqrt{(1+\alpha)^{2(k-1)}\beta_1^2 + \left(\frac{\beta_2}{1+\alpha}\right)^2}} > |\lambda| \cdot \rho_r^k(q) \tag{3.22}$$

Thus since $|\lambda| \geq 1$, $\rho_r^{k+1}(q+1) > |\lambda|\rho_r^k(q)$ holds. □

Next we formally state the dominating sequence before considering some of its properties.

Lemma 22. *The sequence ρ_r^q dominates the sequence ρ_r^*. That is*

$$\rho_r^*(q) \leq \rho_r^q(q), \quad \forall\, q.$$

Proof. This is immediate from the definition of $\rho_r^*(q)$ in (3.17) as the smallest distance at every q. □

Theorem 23. *The rate of increase of the sequence $\rho_r^q(q)$ is always greater than $|\lambda|$ and converges to $|\lambda|$ as $q \to \infty$.*

Proof. The rate of increase is given by

$$\frac{\rho_r^{q+1}(q+1)}{\rho_r^q(q)}.$$

By Theorem. 21 this is strictly greater than $|\lambda|$. To calculate the limit as $q \to \infty$ we first write

$$\begin{aligned}\rho_r^q(q) &= \sum_{j=1}^{q} \frac{|\left((1+\alpha)^{q-1} - (1+\alpha)^{j-1}\right)\lambda^{j-1}\beta_1\beta_2|}{\sqrt{(1+\alpha)^{2q-2}\beta_1^2 + \beta_2^2}} \\ &= \frac{\left|\left((1+\alpha)^{q-1}\sum_{j=1}^{q}|\lambda|^{j-1} - \sum_{j=1}^{q}(1+\alpha)^{j-1}|\lambda|^{j-1}\right)\beta_1\beta_2\right|}{\sqrt{(1+\alpha)^{2q-2}\beta_1^2 + \beta_2^2}} \\ &= \left[\frac{(1+\alpha)^{q-1}\frac{|\lambda|^q-1}{|\lambda|-1} - \frac{(1+\alpha)^q|\lambda|^q-1}{(1+\alpha)|\lambda|-1}}{\sqrt{(1+\alpha)^{2q-2}\beta_1^2 + \beta_2^2}}\right]|\beta_1\beta_2|.\end{aligned}$$

Similarly,

$$\rho_r^{q+1}(q+1) = \left[\frac{(1+\alpha)^{q}\frac{|\lambda|^{q+1}-1}{|\lambda|-1} - \frac{(1+\alpha)^{q+1}|\lambda|^{q+1}-1}{(1+\alpha)|\lambda|-1}}{\sqrt{(1+\alpha)^{2q}\beta_1^2 + \beta_2^2}}\right]|\beta_1\beta_2|.$$

Taking limit of the ratio of the numerators, we get

$$\lim_{q\to\infty} \frac{(1+\alpha)^{q}\frac{|\lambda|^{q+1}-1}{|\lambda|-1} - \frac{(1+\alpha)^{q+1}|\lambda|^{q+1}-1}{(1+\alpha)|\lambda|-1}}{(1+\alpha)^{q-1}\frac{|\lambda|^{q}-1}{|\lambda|-1} - \frac{(1+\alpha)^{q}|\lambda|^{q}-1}{(1+\alpha)|\lambda|-1}} = (1+\alpha)|\lambda|.$$

Looking now at the denominators, we get

$$\lim_{q\to\infty} \frac{\sqrt{(1+\alpha)^{2q}b_1^2 + b_2^2}}{\sqrt{(1+\alpha)^{2q-2}b_1^2 + b_2^2}} = 1+\alpha$$

Combining these we get that the limit of the ratios converges to $|\lambda|$ and the lemma is established. □

Alongside these properties of the sequence $\rho_r^q(q)$, we also establish the following property of the sequence $\rho_r^*(q)$.

Lemma 24. *The recovery distance $\rho_r^*(q)$ is monotonically increasing in q with*

$$\frac{\rho_r^*(q+1)}{\rho_r^*(q)} > |\lambda|.$$

Proof. Let the indices yielding the recovery distance after q and $q+1$ steps be denoted by k_q and k_{q+1} respectively. By Lemma 20, we have $k_q > 1$ and $k_{q+1} > 1$. Thus $\rho_r^{k_{q+1}-1}(q)$ exists. Now, by Theorem 21 we have

$$\rho_r^{k_{q+1}}(q+1) > |\lambda| \cdot \rho_r^{k_{q+1}-1}(q). \tag{3.23}$$

Since $\rho_r^{k_{q+1}}(q+1) = \rho_r^*(q+1)$ and $\rho_r^{k_{q+1}-1}(q) \geq \rho_r^*(q)$ (by the definition of $\rho_r^*(q)$), we have

$$\rho_r^*(q+1) > |\lambda| \cdot \rho_r^*(q) \tag{3.24}$$

and the Lemma is established. □

With Lemmas 22-24 in hand, we are able to show that the growth of ρ^* is bounded by a particular function of q.

Theorem 25. *For the sequence $\rho_r^*(q)$ and for arbitrarily small $\epsilon > 0$, we can always find two indexes $q_1(\epsilon), q_2(\epsilon) \in \mathbb{N}$ with $q_1(\epsilon) > q_2(\epsilon)$, such that*

$$|\lambda|^{q-q_2(\epsilon)} < \frac{\rho_r^*(q)}{\rho_r^*(q_2(\epsilon))} < (|\lambda|+\epsilon)^{q-q_2(\epsilon)}, \forall q > q_1(\epsilon). \tag{3.25}$$

Proof. The left side of (3.25) follows directly from Lemma 24. We now establish the right side. From Theorem 23 we have that, for any ϵ, an index $q_3(\epsilon)$ can be found such that

$$\frac{\rho_r^{q+1}(q+1)}{\rho_r^q(q)} < |\lambda| + \frac{\epsilon}{2}, \forall q > q_3(\epsilon). \tag{3.26}$$

Also, according to Lemma 24 we have $\lim_{q\to\infty} \rho_r^*(q) = \infty$. Thus, there exists an index $q_2(\epsilon)$ such that

$$\rho_r^*(q_2(\epsilon)) > \rho_r^{q_3(\epsilon)}(q_3(\epsilon)) \tag{3.27}$$

According to Lemma 22, we know that $q_2(\epsilon) > q_3(\epsilon)$. Therefore,

$$\frac{\rho_r^*(q)}{\rho_r^*(q_2(\epsilon))} < \frac{\rho_r^{q+1}(q)}{\rho_r^{q_3(\epsilon)}(q_3(\epsilon))} < (|\lambda|+\frac{\epsilon}{2})^{q-q_\epsilon}, \forall q > q_2(\epsilon). \tag{3.28}$$

Since $|\lambda| > 1$ and $\epsilon > 0$, there always exists $q_1(\epsilon) > q_2(\epsilon)$, such that

$$(|\lambda| + \frac{\epsilon}{2})^{q-q_3(\epsilon)} < (|\lambda| + \epsilon)^{q-q_2(\epsilon)}, \forall q > q_1(\epsilon), \tag{3.29}$$

establishing the right side of (3.25) and thus the Theorem. □

Theorem 25 shows that the exponential growth rate of the sequence $\rho_r^*(q)$ can be upper bounded by any rate arbitrarily close to but larger than $|\lambda|$. This will be useful in the next section where we compare the the size of the recovery region with the size of the accumulated disturbance.

In Sec.3.3.2, we extend this calculation to the general MIMO setting.

3.3.2 Calculating $\rho_r^*(q)$ in the general MIMO case

In Sec.3.3.1, we developed the calculation of $\rho_r^*(q)$ and the analysis of its growth rate in the two dimensional, single input setting. In this subsection, we show how to extend those results to the general n-dimensional multiple input setting. We begin with systems whose state matrix is diagonalizable over the reals and then move to systems that need a Jordan form. We first consider the calculation of the recovery radius and then use that to generalize Lemmas 20 through Theorem 23.

Consider, then, an n-dimensional system with matrices

$$A = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$
$$B = \begin{bmatrix} \beta_{1,1}, & \beta_{2,1}, & \cdots & \beta_{m,1} \\ \beta_{1,2}, & \beta_{2,2}, & \cdots & \beta_{m,2} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{1,n}, & \beta_{2,n}, & \cdots & \beta_{m,n} \end{bmatrix}.$$

As before, we are only interested in the unstable portion of the system and thus we assume $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| > 1$ where, without loss of generality, we have ordered the eigenvalues.

Over q continuously connecting steps (with $q > n$), the matrix $\mathfrak{D}(q)$ has qm

columns, given by

$$\mathfrak{D}(q) = \left\{ \begin{bmatrix} \beta_{1,1} \\ \beta_{1,2} \\ \vdots \\ \beta_{1,n} \end{bmatrix}, \cdots, \begin{bmatrix} \beta_{m,1} \\ \beta_{m,2} \\ \vdots \\ \beta_{m,n} \end{bmatrix}, \begin{bmatrix} \lambda_1\beta_{1,1} \\ \lambda_2\beta_{1,2} \\ \vdots \\ \lambda_n\beta_{1,n} \end{bmatrix}, \cdots, \right.$$
$$\left. \begin{bmatrix} \lambda_1\beta_{m,1} \\ \lambda_2\beta_{m,2} \\ \vdots \\ \lambda_n\beta_{m,n} \end{bmatrix}, \cdots, \begin{bmatrix} \lambda_1^{q-1}\beta_{1,1} \\ \lambda_2^{q-1}\beta_{1,2} \\ \vdots \\ \lambda_n^{q-1}\beta_{1,n} \end{bmatrix}, \cdots, \begin{bmatrix} \lambda_1^{q-1}\beta_{m,1} \\ \lambda_2^{q-1}\beta_{m,2} \\ \vdots \\ \lambda_n^{q-1}\beta_{m,n} \end{bmatrix} \right\}.$$

We index the columns of $\mathfrak{D}(q)$ with two indices, the first selecting the group of columns with the same power of λ and the second selecting a particular column in that group. For example,

$$\mathfrak{D}(q)(i,j) = \begin{bmatrix} \lambda_1^{i-1}\beta_{j,1} & \lambda_2^{i-1}\beta_{j,2} & \cdots & \lambda_n^{i-1}\beta_{j,n} \end{bmatrix}^T.$$

Notice that $\mathfrak{D}(q+1)$ is formed by appending additional columns to $\mathfrak{D}(q)$. Thus

$$\mathfrak{D}(q+1)(i,j) = \mathfrak{D}(q)(i,j), \quad i = 1, \ldots, q,\ j = 1, \ldots, m.$$

Similar to the two dimensional setting, the critical direction v_c should be orthogonal to a subset of $n-1$ of the vectors in $\mathfrak{D}(q)$. We again denote the collection of all such possible subsets as $\mathfrak{I}(q)$ and index each of the $\binom{mq}{n-1}$ subsets in this collection as $\mathfrak{I}(q)(\alpha)$. Generically, one of these subsets can be written as the matrix

$$\mathfrak{I}(q)(\alpha) = \begin{bmatrix} \lambda_1^{k_1-1}\beta_{j_1,1} & \ldots & \lambda_1^{k_{n-1}-1}\beta_{j_{n-1},1} \\ \lambda_2^{k_1-1}\beta_{j_1,2} & \ldots & \lambda_2^{k_{n-1}-1}\beta_{j_{n-1},2} \\ \vdots & \vdots & \vdots \\ \lambda_{n-1}^{k_1-1}\beta_{j_1,n-1} & \ldots & \lambda_{n-1}^{k_{n-1}-1}\beta_{j_{n-1},n-1} \\ \lambda_n^{k_1-1}\beta_{j_1,n} & \ldots & \lambda_n^{k_{n-1}-1}\beta_{j_{n-1},n} \end{bmatrix}$$

for some choice of $k_1, \ldots, k_{n-1}$ and $j_1, \ldots, j_{n-1}$. Thus, $\mathcal{I}(q)(\alpha)$ contains $n-1$ columns.

Let

$$v_q(\alpha) = \left[\begin{array}{cccc} v_{q,1}(\alpha) & v_{q,2}(\alpha) & \cdots & v_{q,n}(\alpha) \end{array} \right]^T$$

denote a vector that is orthogonal to the $n-1$ vectors in $\mathfrak{I}(q)(\alpha)$. As a specific choice, we set

$$v_{q,i}(\alpha) = \det \mathfrak{I}_i(q)(\alpha) \tag{3.30}$$

where $\mathfrak{I}_i(q)(\alpha)$ is constructed by eliminating the i^{th} row in $\mathfrak{I}(q)(\alpha)$.

As in the 2-D setting, we look at the sum of the projections of all of the input directions in $\mathfrak{D}(q)$ onto a given $v_q(\alpha)$,

$$\rho_r^{\alpha}(q) = \sum_{\forall \kappa, \forall j} P_{v_q(\alpha)} \left(\mathfrak{D}(q)(\kappa, j) \right),$$

where $P_{v_q(\alpha)}$ is the corresponding projection operator onto $v_q(\alpha)$. The critical direction is the minimum among all these. That is,

$$\rho_r^*(q) = \min_{\forall \alpha} \sum_{\forall \kappa, \forall j} P_{v_q(\alpha)} \left(\mathfrak{D}(q)(\kappa, j) \right)$$

Now we establish how the conclusion of Theorem 25 (bounding the growth rate of the sequence) holds for the n dimensional case. Following Lemma 24, it is easy to show that the growth rate of $\rho_r^*(q)$ is larger than $|\lambda_n|$ (the smallest magnitude eigenvalue). We now establish a dominating sequence $\tilde{\rho}(q)$ and show that the growth rate of the dominating sequence converges to $|\lambda_n|$ as $q \to \infty$.

For any given subset $\mathfrak{I}(q)(\alpha)$, expressed as a matrix as shown above, we define

$$\mathfrak{I}^+(q)(\alpha) = A\mathfrak{I}(q)(\alpha).$$

Note that because every element of $\mathfrak{I}^+(q)$ belongs to the collection $\mathfrak{D}(q+1)$, we have

that

$$\mathfrak{I}^{+}(q)(\alpha) \in \mathfrak{I}(q+1).$$

Let $v_q^{+}(\alpha)$ denote a vector that is orthogonal to all the vectors in $\mathfrak{I}^{+}(q)$.

We define a dominating sequence by starting with a random $v_q(\alpha)$ and denoting the accumulated effect of all the (possible) input directions over q steps on this direction as $\tilde{\rho}_r(q)$. The next entry in this sequence is the summed projections onto the direction of $v_q^{+}(\alpha)$ of the input directions available over $q+1$ steps, and so on. That is,

$$\begin{aligned}
\tilde{\rho}_r(q) &= \sum_{i=1,\ldots,q,\forall j} P_{v_q(\alpha)}\left(\mathfrak{D}(q)(i,j)\right), \\
\tilde{\rho}_r(q+1) &= \sum_{i=1,\ldots,q+1,\forall j} P_{v_q^{+}(\alpha)}\left(\mathfrak{D}(q+1)(i,j)\right).
\end{aligned}$$

Obviously for any q we have

$$\rho_r^{*}(q) \leq \tilde{\rho}_r(q).$$

The growth rate of $\tilde{\rho}_r(q)$ can be calculated as

$$\frac{\tilde{\rho}_r(q+1)}{\tilde{\rho}_r(q)} = \frac{\sum_{i,j} P_{v_q^{+}(\alpha)}\left(\mathfrak{D}(q+1)(i,j)\right)}{\sum_{i,jj} P_{v_q(\alpha)}\left(\mathfrak{D}(q)(i,j)\right)}.$$

For any given i, j we have

$$\begin{aligned}
& P_{v_q^{+}(\alpha)}\left(\mathfrak{D}(q+1)(i+1,j)\right) \\
=& \frac{|\lambda_1|^{\kappa}\beta_{j,1}v_{q,1}^{+} + \cdots + |\lambda_n|^{\kappa}\beta_{j,n}v_{q,n}^{+}}{\sqrt{\left(v_{q,1}^{+}\right)^2 + \left(v_{q,2}^{+}\right)^2 + \cdots + \left(v_{q,n}^{+}\right)^2}} \\
=& \prod_{i=1}^{n} |\lambda_i| \frac{|\lambda_1|^{\kappa-1}\beta_{j,1}v_{q,1} + \cdots + |\lambda_n|^{\kappa-1}\beta_{j,n}v_{q,n}}{\sqrt{\left(v_{q,1}^{+}\right)^2 + \left(v_{q,2}^{+}\right)^2 + \cdots \left(v_{q,n}^{+}\right)^2}}.
\end{aligned}$$

Note that to simplify the notation somewhat, we have omitted the dependence of v_q

on the index α in the expressions. With this result we calculate

$$\frac{P_{v_q^+(\alpha)}\left(\mathfrak{D}(q+1)(i+1,j)\right)}{P_{v_q(\alpha)}\left(\mathfrak{D}(q)(i,j)\right)} = \prod_{i=1}^{n}|\lambda_i|\sqrt{\frac{\left(v_{q,1}\right)^2+\left(v_{q,2}\right)^2+\cdots+\left(v_{q,n}\right)^2}{\left(v_{q,1}^+\right)^2+\left(v_{q,2}^+\right)^2+\cdots+\left(v_{q,n}^+\right)^2}}$$

Using $v_{q,i}^+ = \det \mathfrak{I}_i^+(q)$ yields

$$v_{q,i}^+ = v_{q,i}\prod_{j=1\ldots n, j\neq i}|\lambda_j|$$

Since all $v_{q,l}(\alpha) > 1$ and since $|\lambda_1| > |\lambda_2| \ldots > |\lambda_n|$, we have

$$\lim_{q\to\infty}\frac{P_{v_q^+(\alpha)}\left(\mathfrak{D}(q+1)(\kappa+1,j)\right)}{P_{v_q(\alpha)}\left(\mathfrak{D}(q)(\kappa,j)\right)} = \prod_{i=1}^{n}|\lambda_i|\cdot\frac{1}{|\lambda_1|\cdot|\lambda_2|\cdot\ldots\cdot|\lambda_{n-1}|} = |\lambda_n|$$

and

$$\frac{P_{v_q^+(\alpha)}\left(\mathfrak{D}(q+1)(\kappa+1,j)\right)}{P_{v_q(\alpha)}\left(\mathfrak{D}(q)(\kappa,j)\right)} > |\lambda_n|$$

Therefore we have

$$\lim_{q\to\infty}\frac{\tilde{\rho}_r(q+1)}{\tilde{\rho}_r(q)} = |\lambda_n|$$

and

$$\frac{\tilde{\rho}_r(q+1)}{\tilde{\rho}_r(q)} > |\lambda_n|.$$

The rest of the proof holds as in the two dimensional case.

The analysis above is established on the case of a diagonalizable A. Following a

similar discussion using the Jordan form, one can show that

$$\frac{P_{v_q^+(\alpha)}\left(\mathfrak{D}(q+1)(\kappa+1,j)\right)}{P_{v_q(\alpha)}\left(\mathfrak{D}(q)(\kappa,j)\right)} > \frac{q}{q+1}|\lambda_n|$$

if A has a repeating eigenvalue. The rest of the proof holds as before since $\frac{q}{q+1}$ goes to 1 as $q \to \infty$.

From here it can be established similarly as in Sec.3.3.1, that the growth rate becomes arbitrarily close to the smallest magnitude eigenvalue of the system matrix. For simplicity, we will continue to refer to this as simply λ. We now reintroduce the disturbance.

3.3.3 Effect of the disturbance

In Sec. 3.3.1 we showed that the radius of the recovery region grows with time and the growth rate gets arbitrarily close to λ. Now we are interested in the cumulative effect of the disturbance. Clearly, from the system dynamics, under the effect of the disturbance, each direction grows exponentially, giving rise to an ellipsoid. We bound this ellipse by the largest radius. Let this radius be ρ_e. Since the disturbance doesn't care whether the control is connected or not, we consider the effect after the entire period p rather than just after the connected portion of q steps. From the system dynamics, the growth of ρ_e is defined as in equation 3.14

$$\rho_e(p) = \sum_{i=1}^{p} \delta(|\lambda^*|)^{i-1} = \delta\frac{|\lambda^*|^p - 1}{|\lambda^*| - 1}. \tag{3.31}$$

Here we focus on the effect of the disturbance and therefore set the initial condition to zero. We remark upon the effect of non-zero initial conditions later in Remark 1.

With these two results, we can compare the recovery region with the escaping ball and make a conservative estimate of whether the system has enough control authority to handle the accumulated disturbance.

Theorem 26. *The growth rate of the sequence describing the escaping ball, $\rho_e(p), p = 1, 2, ...,$ lies within the range of $[|\lambda^*|, |\lambda^*| + 1]$. Furthermore, as $p \to \infty$, the sequence converges to an exponentially growing sequence $c|\lambda^*|^p$, for some constant c that depends on the system parameters.*

Proof. We have

$$\frac{\rho_e(p+1)}{\rho_e(p)} = \frac{|\lambda^*|^{p+1} - 1}{|\lambda^*|^p - 1}.$$

The growth rate is monotonically decreasing since

$$\frac{\left(|\lambda^*|^{p+2} - 1\right)\left(|\lambda^*|^p - 1\right)}{\left(|\lambda^*|^{p+1} - 1\right)^2} < 1.$$

Hence the growth rate of $\rho_e(p)$ is upper bounded by its value at $p = 1$. Thus,

$$\frac{\rho_e(p+1)}{\rho_e(p)} \leq \frac{|\lambda^*|^2 - 1}{|\lambda^*| - 1} = |\lambda^*| + 1.$$

Taking the limit yields

$$\lim_{p\to\infty} \frac{|\lambda^*|^{p+1} - 1}{|\lambda^*|^p - 1} = |\lambda^*|, \text{ for } |\lambda^*| > 1,$$

and the lemma is established. □

It was shown in the previous section (and the appendix) that the size of the recovery ball grows with q at a growth rate arbitrarily close to $|\lambda|$.We refer to the radius of this ball as the "recovery radius". We then define the ratio of the recovery radius over the radius of the escaping ball as the "recovery-escape ratio",

$$\mathfrak{r}_{(p,q,r)} = \frac{\rho_r^*(q)}{\rho_e(p)}.$$

The system is considered to be reachable under a given (p, q, r) policy and in the face of both limited control authority and disturbance when this ratio exceeds unity.

If we fix the number of blind steps r in the communication policy, then as we

increase the number of connected steps q we find

$$\lim_{q\to\infty} \mathfrak{r}_{(p,q,r)} = \lim_{q\to\infty} \frac{\kappa|\lambda|^q}{c|\lambda^*|^p} = \frac{\kappa}{c|\lambda^*|^r} \lim_{q\to\infty} \left(\frac{|\lambda|}{|\lambda^*|}\right)^q.$$

It is obvious that, because $|\lambda| < |\lambda^*|$, this sequence $\mathfrak{r}_{(p,q,r)}$ approaches an exponentially decreasing sequence when q is large enough. However, for small q, this ratio may increase since the growth rate of $\rho_e(p)$ is upper bounded and the growth rate of $\rho_r^*(q)$ may initially be large enough to dominate $\rho_e(p)$.

That is to say, there exists a finite $\bar{q}$, such that any number of q beyond this $\bar{q}$ yields to a decreasing recover-escape ratio. We shall be able to find out an optimal number of $q^* \leq \bar{q}$ such that the recover-escape ratio is maximized. As shown in Fig. 3·4, $\bar{q}$ is when $\mathfrak{r}_{(p,q,r)}$ drops to 1, and q^* is approximately 10.

Noted that this result also serves for an always connected system of disturbance to find its best 'lifting' period. Such a system 'lifted' by q steps can be considered as a system under a $(q, q, 0)$ communication condition. The conclusion still holds that there exists an optimal q^* to maximize the recover-escape ratio.

As elaborated in Sec.3.1, the concept of reachability in a system with *a priori* unknown disturbance is about the ability of a system bringing its states into a limited ball. This limited ball grows with the increase of the 'lifting' periodicity. Therefore the choice of the lifting periodicity is a trade off between maximizing the recover-escape ratio and to limit the final range in which the states is guaranteed to arrive.

Remark 1. *While the feature of the evolution of $\mathfrak{r}_{(p,q,r)}$ is defined by system parameters, the curve can be shifted by several factors. Changing δ is one of them. The initial system status also has an effect on it. In this work we take "reachable" as the capability of bringing the system status back into a limited ball around zero. The recovery-escape ratio $\mathfrak{r}_{(p,q,r)}$ compares the power to drive the system status far away from zero and the power to reverse this effect. This comparison is based on the assumption that the system status is exactly zero at the beginning of this period. Notice that even if $\mathfrak{r}_{(p,q,r)}$ is greater than 1, it only implies the recovery power in this period*

is strong enough to cancel the escaping power. In practice, the recovery power is provided through inputs and cannot be designed against time sequence. Thus it is always possible, that the system status is NOT zero but in a limited ball at the end of one visit, and also at the beginning of next period. Therefore it is important to understand the system dynamics with non-zero initial status. The recovery-escape ratio would be affected by carrying extra escaping power driven by the non-zero initial status. We take the worst case scenario to make a conservative estimation, that $\rho_e(p)$ *becomes* $(|\lambda|^*)^p|x(0)| + \sum_{i=1}^{p} \delta|\lambda|^* i - 1$. *The system is still reachable if the recovery-escape ratio stays greater than* 1.

Remark 2. *In the general* $(p, \bar{q}, \bar{r})$ *setting, everything just goes through swimmingly. Given* $(p, \bar{q}, \bar{r})$ *sequence, we are able to calculate its recovery-escape ratio by enumerating all possible combinations of the spanning directions provided by connected steps.*

3.3.4 Example

Let's consider an example that illustrate the results in this section.

A system with dynamics as in (3.1)

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + d(k), \\ x &\in \mathbb{R}^x \\ u(k) &\in \mathfrak{U},\ \mathfrak{U} = \{u \in \mathbb{R}^1 \| |u_1| \leq 1\}, \\ \|d(k)\|_2 &\leq \delta, \end{aligned}$$

where the state matrix A has three eigenvalues $[1.0136; 1.0580; 1.2269]$, such that the maximum magnitude eigenvalue is $|\lambda^*| = 1.2269$ and the minimum magnitude eigenvalue is $|\lambda| = 1.0136$. The input matrix is $B = [1.8186; 1.8987; 1.1723]^T$. The boundary of disturbance is set to $\delta = 0.01$.

Fig 3·4 shows that, by fixing $r = 1$, let p grows with q, how the recovery and escape radius grow at small q. The figure shows that the recovery power is strong enough to reverse the escape power if we set $q \in (3, 20)$. Notice that it is an estimate based on the worst case. Even if q does not fall in the given range, there is still chance

to bring the status back depending on the direction of disturbance.

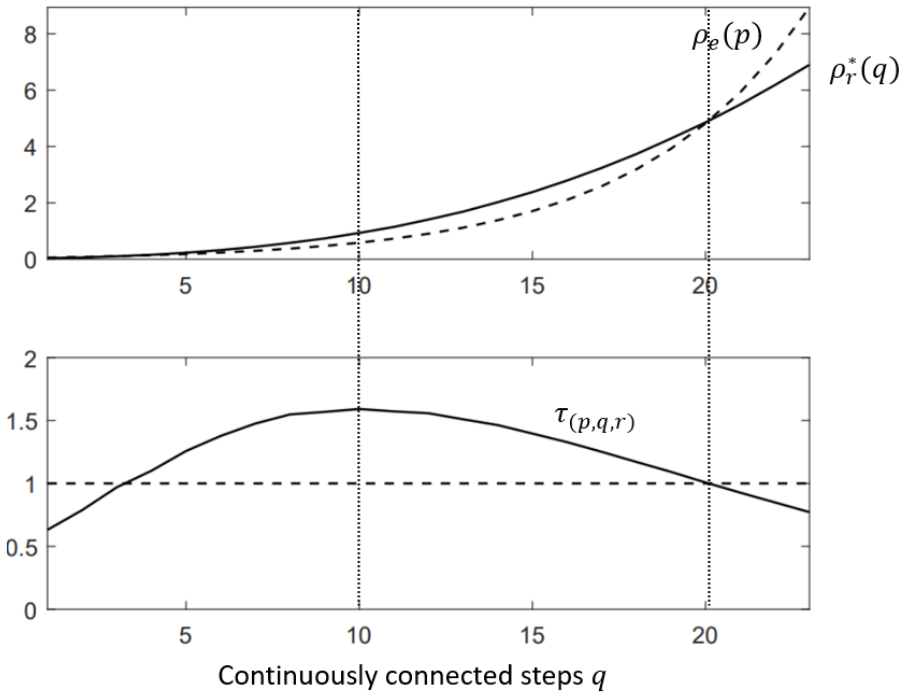


Figure 3·4: The escape and recovery radius, and the recovery-escape ratio. This ratio reaches maximum at $q = 10$.

At large q, the recovery-escape ratio $\mathfrak{r}_{(p,q,r)}$ converges to zero, as shown in Fig.3·5. As suggested in the analysis, the growth rate of recovery radius $\rho_r^*(q)$ will get arbitrarily close to the minimum magnitude eigenvalue at large q, while increasing q results in an increase in p and the escape radius $\rho_e(p)$ grows in a faster rate and will finally dominate the system.

Fig. 3·6 shows the effect while altering the disturbance boundary to $\delta = 0.008$ and $\delta = 0.012$. Greater disturbance would cause a larger escape radius. The feature of the recovery-escape ratio curve remains the same but the available range of q is smaller.

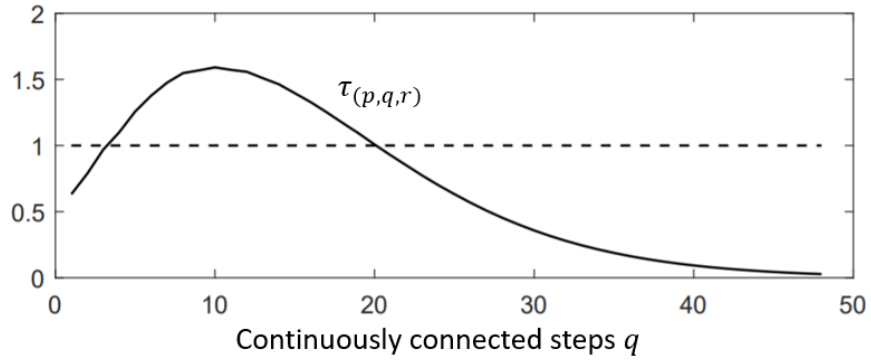


Figure 3·5: The recover-escape ratio $\mathfrak{r}_{(p,q,r)}$ increases at first and converges to zero.

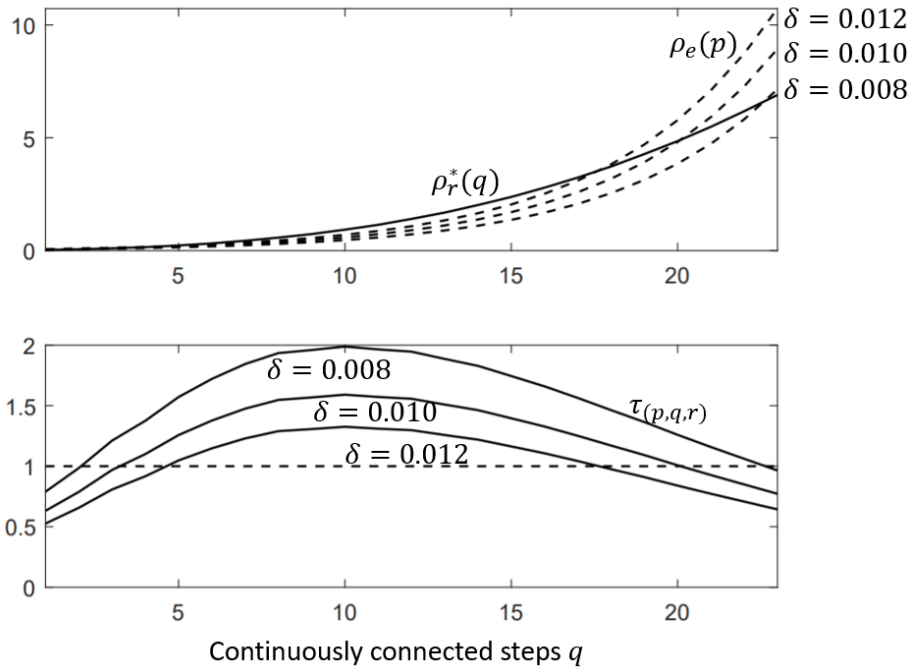


Figure 3·6: Changing δ is altering the available range of q.

Chapter 4

Conclusions and future work

This thesis studied the multiple-agent multiple-target persistent monitoring problem from both the agents and the targets perspectives. The first was to schedule the agents to visit the targets, dwell at the targets, and perform control on them to drive the dynamic states carried by the targets as desired. The scheduling can be designed by either modeling the target space as a continuous domain or taking the targets as discrete tasks distributed according to a graph describing the geometry of the targets' layout. Following the discrete approach, we assumed each of the targets carried an uncertainty state with linear increasing and decreasing dynamics, and determined the optimal dwell time of an agent moving between multiple targets while seeking to minimize some function of the targets' uncertainty states. The problem was abstracted into a discrete dynamic system and we showed the existence of stable solutions to the dynamics of the uncertainty states under certain conditions. We further showed that the simple choice of staying with a target until its uncertainty reached zero and then switching to another was optimal for minimizing the steady state of uncertainty, peak uncertainty, and period length. In addition, when the targets had homogeneous dynamics in their uncertainty, the zero policy also minimized the steady state average uncertainty and, under this special condition, the problem was in fact reduced to a TSP.

Building upon these results, we considered a fixed sequence that may include multiple visits to individual targets and determined under what conditions it was

optimal for the agent to depart from its current target as soon as the information state reached zero. We then fixed this policy to establish conditions that an optimal sequence must satisfy, including bounding the total number of visits. Finally, we developed an algorithm for refining a giving sequence by looking at possible single-visit additions.

Notice that these results were generated under the assumption of a linear dynamic model of the target's state. A natural extension of the work would be considering much more complicated dynamic models. For example, an exponential increasing and decreasing curve would be of great interest. Instead of considering all targets carrying the same type of well defined dynamic states only with different parameter, it would be interesting to consider a general case where only general properties, such as target states monotonically decreasing when attended by an agent and monotonically increasing otherwise.

As shown in 2.2.2.3 and 2.2.3.3, the condition that the overall uncertainty remains bounded echoes the results in queuing theory, such that the service rate should be faster than the arrival rate. The results that the switching time occurs when the uncertainty drops to zero can also be related to queuing theory, since an agent remaining after that time point would not result in any decrease of uncertainty, and would only reduce the service rate in one period of visit. A possible extension is to build upon this analogy. For example, the agent should leave the current target when the decreasing rate (service rate) at the target is slower than the sum of the increasing rate (arrival rate) of other targets. This hypothesis is not rigorous since it assumes an non-increasing service rate and a nondecreasing arrival rate. One possible extension of the work could be to bound the states' evolution with a monotonically decreasing curve with non-increasing slope and monotonically increasing curve with nondecreasing slope, and then analyze the optimized switching condition using these

bounds.

The second perspective considered in this work was to allow each target to carry a state evolving according to a discrete time linear system that could be controlled only when being attended by an agent. Under the assumption that a target's dynamics were reachable when being attended to continuously, we established conditions such that the intermittent nature of the control, due to the need of the agents to move among the targets, did not cause a loss of reachability of the system. We then considered the effect of both a finite disturbance and finite input power in the face of this intermittent control, showing that there is a range of periodicity on the visits to a target that will guarantee the state of the system can be driven to a finite ball at the end of each period.

These results provided a way to evaluate the effectiveness of a specific sequence. In addition to use this conclusion to filter for feasible visiting sequences, these results can also be used in a persistent monitoring system with mobile targets. Given a pre-selected sequence of agents visiting mobile targets in the target space, the periodicity will change with the evolving geometry. Our results can be applied on a per-cycle basis so long as the the target motion is slow enough relative to the periodicity of the monitoring.

References

Ashley, T. T., Chan-Tse, C., and Andersson, S. B. (2012). Validation of a nonlinear reactive control law for three-dimensional particle tracking in confocal microscopy. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 2328–2333. IEEE.

Ashley, T. T., Gan, E. L., Pan, J., and Andersson, S. B. (2016). Tracking single fluorescent particles in three dimensions via extremum seeking. *Biomedical Optics Express*, 7(9):3355–3376.

Baillieul, J. and Kong, Z. (2014). Saliency based control in random feature networks. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 4210–4215. IEEE.

Bamieh, B., Pearson, J. B., Francis, B. A., and Tannenbaum, A. (1991). A lifting technique for linear periodic systems with applications to sampled-data control. *Systems & Control Letters*, 17(2):79–88.

Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219.

Blanchini, F. (1999). Set invariance in control. *Automatica*, 35(11):1747–1767.

Blondel, V. D. and Tsitsiklis, J. N. (1999). Complexity of stability and controllability of elementary hybrid systems. *Automatica*, 35(3):479–489.

Breitenmoser, A., Schwager, M., Metzger, J.-C., Siegwart, R., and Rus, D. (2010). Voronoi coverage of non-convex environments with a group of networked robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4982–4989. IEEE.

Busoniu, L., Babuska, R., and De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(2):156–172.

Cao, M., Morse, A. S., Yu, C., Anderson, B., Dasgupta, S., et al. (2011). Maintaining a directed, triangular formation of mobile autonomous agents. *Communications in Information and Systems*, 11(1):1.

Cassandras, C. G., Lin, X., and Ding, X. (2013). An optimal control approach to the multi-agent persistent monitoring problem. *IEEE Transactions on Automatic Control*, 58(4):947–961.

Cassandras, C. G., Wardi, Y., Panayiotou, C. G., and Yao, C. (2010). Perturbation analysis and optimization of stochastic hybrid systems. *European Journal of Control*, 16(6):642–661.

Cromer Berman, S. M., Walczak, P., and Bulte, J. W. (2011). Tracking stem cells using magnetic nanoparticles. *Wiley Interdisciplinary Reviews: Nanomedicine and Nanobiotechnology*, 3(4):343–355.

De Gennaro, M. C. and Jadbabaie, A. (2006). Decentralized control of connectivity for multi-agent systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3628–3633. IEEE.

Friedland, B. (1975). Controllability index based on conditioning number. *Journal of Dynamic Systems, Measurement, and Control*, 97(4):444–445.

Gawronski, W. and Lim, K. (1996). Balanced actuator and sensor placement for flexible structures. *International Journal of Control*, 65(1):131–145.

Gusrialdi, A., Hirche, S., Hatanaka, T., and Fujita, M. (2008). Voronoi based coverage control with anisotropic sensors. In *Proceedings of American Control Conference*, pages 736–741. IEEE.

Hamdan, A. and Nayfeh, A. (1989). Measures of modal controllability and observability for first-and second-order linear systems. *Journal of guidance, control, and dynamics*, 12(3):421–428.

Hautus, M. (1970). Stabilization controllability and observability of linear autonomous systems. In *Indagationes mathematicae (proceedings)*, volume 73, pages 448–455. Elsevier.

Horling, B. and Lesser, V. (2004). A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19(04):281–316.

Junkins, J. L. and Kim, Y. (1991). Measure of controllability for actuator placement. *Journal of Guidance, Control, and Dynamics*, 14(5):895–902.

Kang, O., Park, Y., Park, Y., and Suh, M. (2009). New measure representing degree of controllability for disturbance rejection. *Journal of Guidance, Control, and Dynamics*, 32(5):1658–1661.

Khazaeni, Y. and Cassandras, C. G. (2016). Event excitation for event-driven control and optimization of multi-agent systems. In *IEEE International Workshop on Discrete Event Systems*. IEEE.

Klein, G., Ongman, R., and Indberg, R. (1982). Computation of a degree of controllability via system discretization. *Journal of Guidance, Control, and Dynamics*, 5(6):583–588.

Koller, D., Daniilidis, K., and Nagel, H.-H. (1993). Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281.

Lahijanian, M., Wasniewski, J., Andersson, S. B., and Belta, C. (2010). Motion planning and control from temporal logic specifications with probabilistic satisfaction guarantees. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3227–3232. IEEE.

Lakadamyali, M., Rust, M. J., Babcock, H. P., and Zhuang, X. (2003). Visualizing infection of individual influenza viruses. *Proceedings of the National Academy of Sciences*, 100(16):9280–9285.

Lan, X. and Schwager, M. (2013). Planning periodic persistent monitoring trajectories for sensing robots in gaussian random fields. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2415–2420. IEEE.

Laporte, G., Gendreau, M., Potvin, J.-Y., and Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7(4-5):285–300.

Li, X. R. and Jilkov, V. P. (2003). Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364.

Li, X. R. and Jilkov, V. P. (2005). Survey of maneuvering target tracking. part v. multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321.

Lin, X. and Cassandras, C. G. (2013). An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 6886–6891. IEEE.

Lin, X. and Cassandras, C. G. (2015). An Optimal Control Approach to the Multi-Agent Persistent Monitoring Problem in Two-Dimensional Spaces. *IEEE Transactions on Automatic Control*, 60(6):1659–1664.

Lygeros, J., Tomlin, C., and Sastry, S. (1999). Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370.

Mao, G. (2009). *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking: Monitoring and Surveillance Techniques for Target Tracking*. IGI Global.

Mathew, N., Smith, S. L., and Waslander, S. L. (2013). A graph-based approach to multi-robot rendezvous for recharging in persistent tasks. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3497–3502. IEEE.

Michael, N., Stump, E., and Mohta, K. (2011). Persistent surveillance with a team of mavs. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Moerner, W. E. and Fromm, D. P. (2003). Methods of single-molecule fluorescence spectroscopy and microscopy. *Review of Scientific Instruments*, 74(8):3597–3619.

Oh, K.-K. and Ahn, H.-S. (2014). Formation control and network localization via orientation alignment. *IEEE Transactions on Automatic Control*, 59(2):540–545.

Raković, S., Kerrigan, E. C., Mayne, D. Q., and Kouramas, K. I. (2007). Optimized robust control invariance for linear discrete-time systems: Theoretical foundations. *Automatica*, 43(5):831–841.

Ren, W., Beard, R. W., and Atkins, E. M. (2005). A survey of consensus problems in multi-agent coordination. In *Proceedings of American Control Conference*, pages 1859–1864. IEEE.

Schwager, M., Rus, D., and Slotine, J.-J. (2009). Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375.

Shen, Z. and Andersson, S. B. (2011). Tracking nanometer-scale fluorescent particles in two dimensions with a confocal microscope. *IEEE Transactions on Control Systems Technology*, 19(5):1269–1278.

Shoham, Y., Powers, R., and Grenager, T. (2003). Multi-agent reinforcement learning: a critical survey. *Web manuscript*.

Smith, S. L., Schwager, M., and Rus, D. (2011a). Persistent monitoring of changing environments using a robot with limited range sensing. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 5448–5455. IEEE.

Smith, S. L., Schwager, M., and Rus, D. (2012). Persistent Robotic Tasks: Monitoring and Sweeping in Changing Environments. *IEEE Transactions on Robotics*, 28(2):410–426.

Smith, S. L., Tumova, J., Belta, C., and Rus, D. (2011b). Optimal path planning for surveillance with temporal-logic constraints. *The International Journal of Robotics Research*, 30(14):1695–1708.

Spengler, M. and Schiele, B. (2003). Automatic detection and tracking of abandoned objects. In *Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. Citeseer.

Stone, P. and Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383.

Stump, E. and Michael, N. (2011). Multi-robot persistent surveillance planning as a vehicle routing problem. In *Proceedings of IEEE Conference on Automation Science and Engineering*, pages 569–575. IEEE.

Sun, X. and Cassandras, C. G. (2015). Optimal dynamic formation control of multi-agent systems in environments with obstacles. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pages 2359–2364. IEEE.

Sun, Z., Ge, S. S., and Lee, T. H. (2002). Controllability and reachability criteria for switched linear systems. *Automatica*, 38(5):775–786.

Viswanathan, C. and Longman, R. (1983). The determination of the degree of controllability for dynamic systems with repeated eigenvalues. *Engineering Science and Mechanics*, pages 1091–1111.

Viswanathan, C., Longman, R., and Likins, P. (1984). A degree of controllability definition-fundamental concepts and application to modal systems. *Journal of Guidance, Control, and Dynamics*, 7(2):222–230.

Wardi, Y., Adams, R., and Melamed, B. (2010). A unified approach to infinitesimal perturbation analysis in stochastic flow models: the single-stage case. *IEEE Transactions on Automatic Control*, 55(1):89–103.

Yang, P., Freeman, R., Lynch, K. M., et al. (2008). Multi-agent coordination by decentralized estimation and control. *IEEE Transactions on Automatic Control*, 53(11):2480–2496.

Yildiz, A., Park, H., Safer, D., Yang, Z., Chen, L.-Q., Selvin, P. R., and Sweeney, H. L. (2004). Myosin vi steps via a hand-over-hand mechanism with its lever arm undergoing fluctuations when attached to actin. *Journal of Biological Chemistry*, 279(36):37223–37226.

Yu, J., Karaman, S., and Rus, D. (2015). Persistent monitoring of events with stochastic arrivals at multiple stations. *IEEE Transactions on Robotics*, 31(3):521–535.

Yu, X. and Andersson, S. B. (2013). Effect of switching delay on a networked control system. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 5945–5950. IEEE.

Yu, X. and Andersson, S. B. (2014). Preservation of system properties for networked linear, time-invariant control systems in the presence of switching delays. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 5260–5265. IEEE.

Yu, X., Andersson, S. B., Zhou, N., and Cassandras, C. G. (2017). Optimal dwell times for persistent monitoring of a finite set of targets. In *Proceedings of American Control Conference*, pages 5544–5549. IEEE.

Zhang, L. and Hristu-Varsakelis, D. (2006). Communication and control co-design for networked control systems. *Automatica*, 42(6):953–958.

Zhong, M. and Cassandras, C. G. (2011). Distributed coverage control and data collection with mobile sensor networks. *IEEE Transactions on Automatic Control*, 56(10):2445–2455.

Zhou, N., Yu, X., Andersson, S. B., and Cassandras, C. G. (2016). Optimal event-driven multi-agent persistent monitoring of a finite set of targets. In *Proceedings of the 55th IEEE Conference on Decision and Control*, pages 1814–1819. IEEE.

Curriculum Vitae