Spring 4-17-2017

Computing the Reach-Avoid Set for Space Vehicle Maneuvering in the Presence of Debris

Matthew Shubert

Matt Shubert

Candidate

Electrical and Computer Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Meeko Oishi , Chairperson

Rafael Fierro

R. Scott Erwin

Computing the Reach-Avoid Set for Space Vehicle Maneuvering in the Presence of Debris

by

Matt Shubert

B.S., Mechanical Engineering, New Mexico Tech, 2015

THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science
in Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

May, 2017

Acknowledgments

I would like to thank my lab partners Abraham Vinod and Joseph Gleason as well as Dr. Baisravan HomChaudhuri for their support and insight into my work. I would also like to thank my advisor, Dr. Meeko Oishi, for her invaluable help and guidance throughout my time at UNM.

This material is based upon work supported by the National Science Foundation and by the Air Force Office of Scientific Research, under Grant Number CMMI-1254990 (CAREER, Oishi) and an AFOSR Summer Faculty Fellowship (Oishi). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Computing the Reach-Avoid Set for Space Vehicle Maneuvering in the Presence of Debris

by

Matt Shubert

B.S., Mechanical Engineering, New Mexico Tech, 2015

M.S., Electrical Engineering, University of New Mexico, 2017

Abstract

The ability to guarantee the safety of autonomously controlled space vehicles is of great importance to help avoid accidents and ensure mission success. In this paper we investigate the safety verification of a satellite attempting to maneuver to a new position while avoiding multiple pieces of debris. We assume that the satellite, desired rendezvous point, and all debris are near the same circular orbit with dynamics modeled by Clohessy-Wiltshire-Hill (CWH) equations. We will use reachability methods to guarantee the satellite is able to reach a desired point while avoiding all debris. We will first develop a computationally efficient method for computing the Reach-Avoid set for a system modeled by CWH dynamics, and then extend this method to the minimal and maximal reach calculations. We then review a system decomposition method for computing reach sets in large dimensions and apply the methods to the debris avoidance problem. Finally, we develop computationally efficient methods to compute an under-approximation of the Reach-Avoid set and present numerical examples for single and multiple debris scenarios.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The development of autonomous control systems has become more prevalent in a wide variety of applications including government satellites. The ability to generate control sequences such that the systems will continue to operate in a safe manner has also become an important point of research in recent decades. The ability to ensure the safe operation of a multimillion dollar satellite is a chief concern. The DART mishap gives an example of the importance of safe controller synthesis for large scale projects. In this example, the DART satellite, which intended to demonstrate autonomous rendezvous technologies of space vehicles, missed a critical waypoint. Because of this, the DART satellite did not transition to its close proximity stage of the rendezvous process. It instead collided with its rendezvous target at a high velocity resulting in a NASA designated "Type A" mishap, meaning a NASA mission failure resulting in a government loss of more than one million dollars [1]. Because of accidents like this, the ability to guarantee safety of autonomously controlled systems is of great importance; not only for the monetary costs involved with failures but also to ensure the systems can continue to perform their required tasks.

One method often used to guarantee the safety of autonomous systems is reacha-

bility [2]. With this technique, safety is defined by unsafe regions of the state space, or specific state configurations the controlled vehicle should not enter. For a debris avoidance problem this region might look like a bounding box around the position of the debris. Reachability can then give a new region in the state space from which it is possible for the controlled vehicle to avoid the unsafe region. Additionally, a region in the state space can be defined as the target. In a rendezvous problem this might be a bounding box around the position of the rendezvous target vehicle. Under these classifications reachability can generate a region in the state space from which it is possible to reach the target. Finally, both of these concepts can be combined and reachability can generate a region in the state space from which it is possible to reach the target region while avoiding all unsafe regions; this set is referred to as the Reach-Avoid set [3], [4]. Applying this technique to a space vehicle rendezvous problem, coupled with debris avoidance, can lead to a set of states from which the controlled vehicle will be able to safely reach its rendezvous target while avoiding all the debris.

In recent years there has been much research into reachability. However, one of the main issues that still plagues the field is computational constraints due to high dimensionality of the systems. To find a solution for a system with greater than three or four dimension becomes intractable. In the past, others have developed methods for systems which match a specific set of assumptions, or for systems with unsafe and/or target regions that are defined by specific shapes [7], [8], [9], [10]. The methods developed for these specific cases can handle systems with larger dimensions.

In more recent months, methods have been developed for computing a single reach or avoid calculation for a large dimension system by decoupling the system dynamics and the target or avoid set [18]. This method, referred to as system decomposition, allows the current computational methods to be applied to the smaller dimension systems independently, and then reconstructs the larger dimension reach or avoid

set through specific combinations of lower dimension reach or avoid sets. While this method does allow for exact solutions of large dimension systems, it only applies to specific reach calculations under specific target or avoid set formulations.

In this thesis we apply reachability techniques to the problem of space vehicle rendezvous with debris. Specifically, we look at a satellite attempting to maneuver to a new region near the same orbit while avoiding a collision with a single, or multiple, piece(s) of uncontrolled debris near the same orbit. Under the assumption that all objects are near the same circular reference orbit we utilize the well studied Clohessy-Wiltshire-Hill (CWH) dynamics to relate the relative positions and velocities of all objects to one another. We develop a computationally tractable method to calculate the Reach-Avoid set for the case where a satellite is maneuvering to a new position while staying outside an unsafe region, referred to as the two-vehicle problem. Because of the non-convexity of the safe region of the debris avoidance problem the method developed for the two-vehicle problem can not be directly applied to the debris avoidance problem. We instead develop two different methods to under approximate the full Reach-Avoid set by applying system decomposition to the multiple debris avoidance problem for one method and to the multiple debris invariance problem for the other method. The debris invariance method can be implemented using independent calculations for each piece of debris while the debris avoidance method requires the avoid set to be calculated for all debris at once. The main contributions of this thesis are: 1) applying system decomposition reachability techniques to the satellite debris avoidance problem and 2) under approximating the full Reach-Avoid set for the satellite rendezvous problem with debris avoidance using independent calculations for reaching the target and avoiding the debris.

Chapter 2

Problem Formulation

2.1 CWH Dynamics

The CWH dynamics are a well studied linear time-invariant (LTI) model relating the six dimensional position and velocity of one spacecraft to another [11], [12], [13]. The vehicle at the origin of the coordinates is ofter referred to as the chief while the other vehicle is referred to as the chaser. These dynamics assume that both the chief and the chaser are near the same circular orbit and are used to model close proximity maneuvers. While the CWH equations do model the full six dimensional system, the out-of-plane relative position and velocity are decoupled from the in-plane dynamics. For the purposes of our work we are only interested in the in-plane dynamics and

thus our four dimensional CWH equations are as follows

$$\begin{aligned}
\dot{z}(t) &= f(z,u) \\
\dot{z}(t) &= Az(t) + Bu(t) \\
&= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3\omega^2 & 0 & 0 & 2\omega \\ 0 & 0 & 0 & 1 \\ 0 & -2\omega & 0 & 0 \end{bmatrix} z(t) + \begin{bmatrix} 0 & 0 \\ \frac{1}{m_c} & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_c} \end{bmatrix} u(t)
\end{aligned} \tag{2.1}$$

With state $z = [x, \dot{x}, y, \dot{y}]^T \in \mathbb{R}^4$ representing the relative in-plane position and velocity of the chaser with respect to the chief. Space vehicles often utilize "on/off" thrusters which can either be fully on or fully off. To model this, our control is given by $u = [u_1, u_2]^T \in \mathcal{U} = \{[0,0]^T,\ [-u_{\max}, -u_{\max}]^T, [-u_{\max}, u_{\max}]^T, [u_{\max}, -u_{\max}]^T, [u_{\max}, u_{\max}]^T\}$. Finally, we have known constants ω and m_c representing orbital and mass constants for the chaser vehicle.

Furthermore, the state transition matrix for the system modeled using CWH dynamics as a function of time can easily be calculated as follows

$$\begin{aligned}
\Phi &= e^{At} \\
&= \begin{bmatrix} 4 - 3\cos\omega t & \frac{1}{\omega}\sin\omega t & 0 & -\frac{2}{\omega}(\cos\omega t - 1) \\ 3\omega\sin\omega t & \cos\omega t & 0 & 2\sin\omega t \\ 6(\sin\omega t - \omega t) & \frac{2}{\omega}(\cos\omega t - 1) & 1 & \frac{4}{\omega}\sin\omega t - 3t \\ 6\omega(\cos\omega t - 1) & -2\sin\omega t & 0 & 4\cos\omega t - 3 \end{bmatrix}
\end{aligned} \tag{2.2}$$

For our problem we are not concerned with the rendezvous of two space craft, but instead with simply maneuvering one space craft to another point in a nearby region. For the purposes of our problem the chief will not represent a second satellite, but will instead represent a new point in the state space which the chaser is trying to reach. For the sake of simplicity this point will still be referred to as the chief.

We now introduce the state $z_{d,i} = [x_{d,i}, \dot{x}_{d,i}, y_{d,i}, \dot{y}_{d,i}]^T \in \mathbb{R}^4$ which represents the relative position and velocity of the i^{th} piece of debris with respect to the chief point. We will assume that every piece of debris has no control input and therefore undergoes only autonomous state evolution. Hence, the dynamics for the state evolution of the debris are given by

$$\begin{aligned} \dot{z}_{d,i}(t) &= f_d(z_{d,i}) \\ \dot{z}_{d,i}(t) &= Az_{d,i}(t) \\ &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3\omega^2 & 0 & 0 & 2\omega \\ 0 & 0 & 0 & 1 \\ 0 & -2\omega & 0 & 0 \end{bmatrix} z_{d,i}(t) \end{aligned} \tag{2.3}$$

We also assume that the every piece of debris considered in our calculation is near the same circular orbit as the chaser satellite. In this case, the A matrix for the debris state dynamics is the same as the A matrix for the chaser state dynamics.

2.2 Target and Avoid Sets

To guarantee the safety of the chaser satellite using reachability techniques we must first define different sets in the state space. We first define a target set as the set of states the chaser wishes to reach at some final time T. For our maneuvering problem we will define the set, $\mathbf{Target}_z$, as a four dimension bounding box around the chief point which can be described by the vector inequality $z_{\min} \leq z \leq z_{\max}$. The $\mathbf{Target}_z$ set can also be viewed as an infinity-norm around the chief point.

We then define an avoid set as the set of states which the chaser satellite should not enter for all time over the time horizon. For the debris avoidance problem this set simply becomes a bounding box around the position of the debris. To differentiate

the debris from the target set the avoid sets will be given by the 1-norm around the position of each piece of debris. However, because the debris is moving, the position of the debris is not necessarily constant over all time. Hence, our avoid set becomes a function of time. We then define the set, $\mathbf{Avoid}_{d,i}(s)$, as the avoid set for the i^{th} piece of debris at time s. The $\mathbf{Avoid}_{d,i}(s)$ set should ensure that the chaser does not hit the debris at any velocity. Ideally, the $\mathbf{Avoid}_{d,i}(s)$ set will then become a bounding box around only the position states of the debris. That is $x_{d,i\ \min} \leq x_{d,i} \leq x_{d,i\ \max}$ and $y_{d,i\ \min} \leq y_{d,i} \leq y_{d,i\ \max}$, in general, leaving the velocities unbounded. This set will contain the chaser satellite if it enters the position states of the debris at any velocity. We also define the union of all the debris avoid sets as

$$\overline{\mathbf{Avoid}}_d(s) = \bigcup_{i=1}^{q} \mathbf{Avoid}_{d,i}(s) \tag{2.4}$$

Figure 2.1 shows a two dimension representation of the states for the chaser and an arbitrary number, q, pieces of debris, along with the $\mathbf{Target}_z$ set and the $\mathbf{Avoid}_{d,i}(s)$ sets for each piece of debris.

2.3 Reachability Framework

To guarantee the safety of a satellite maneuvering in the presence of debris we will utilize reachability techniques. Using our previous definitions for the target and avoid sets we would like to compute the Reach-Avoid set. This will represent the set of initial conditions for the chaser satellite from which it can reach the target while avoiding all debris for all time, hence guaranteeing the safety of the satellite. Additionally, reachability can also provide the specific controller required for the chaser from the previously generated set of initial states [5], [6]. The following is a general framework for calculating a Reach-Avoid set.

We begin with a continuous time system of the form $\dot{z} = f(z, u, v)$ where $z \in \mathbb{R}^n$

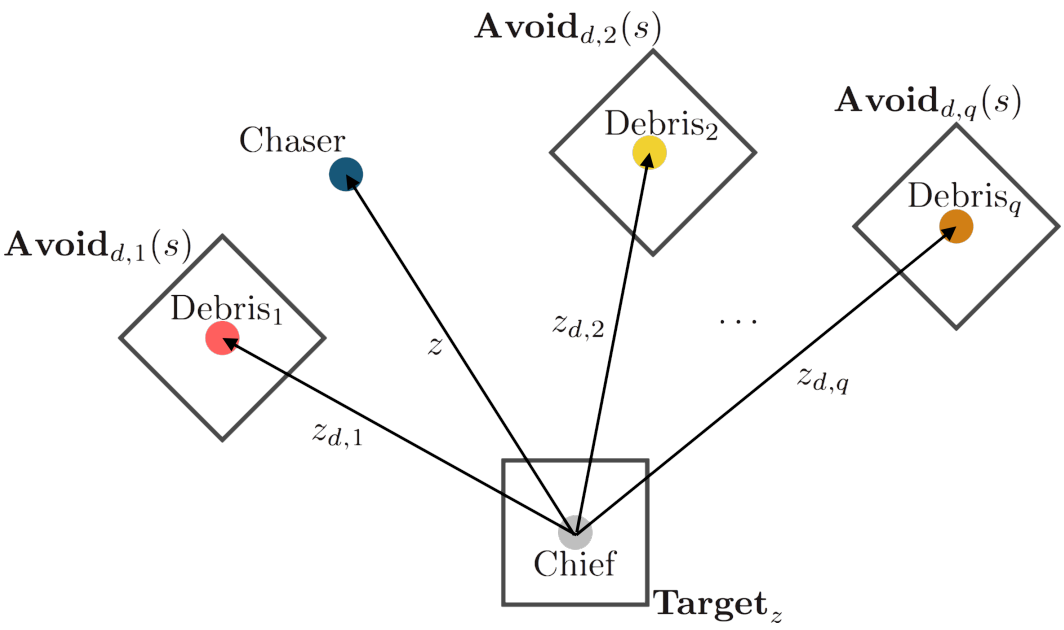

Figure 2.1: Diagram of relative coordinates between Chief, Chaser, and Debris and Target and Avoid sets

is the state of the system, $u \in \mathbb{R}^m$ is the control input, and $v \in \mathbb{R}^p$ is a disturbance acting on the system. Let $z(t)$ represent the current state of the system at a time t. Now define two known sets in the state space as follows, the target set, $\mathbf{Target} \subseteq \mathbb{R}^n$, and the avoid set, $\mathbf{Avoid} \subseteq \mathbb{R}^n$. Both the **Target** and **Avoid** sets can be related to the level sets of two bounded, Lipschitz continuous functions. The function, $l : \mathbb{R}^n \to \mathbb{R}$, will represent the **Target** and the function, $h : \mathbb{R}^n \to \mathbb{R}$, will represent the **Avoid** as follows

$$\begin{aligned} \mathbf{Target} &= \{z \in \mathcal{Z} \mid l(z) \leq 0\} \\ \mathbf{Avoid} &= \{z \in \mathcal{Z} \mid h(z) > 0\} \end{aligned} \tag{2.5}$$

From [3] the **ReachAvoid** set is defined as

$$\begin{aligned} \mathbf{ReachAvoid}(t, \mathbf{Target}, \mathbf{Avoid}) = \{z \in \mathcal{Z} \mid \exists u \in \mathcal{U}_{[t,T]},\ \forall v \in \mathcal{V}_{[t,T]}, \\ (z(T) \in \mathbf{Target}) \wedge (\forall \tau \in [t, T]\ z(\tau) \notin \mathbf{Avoid})\} \end{aligned} \tag{2.6}$$

Where u is the control input to the system, v is a disturbance to the system, t is the initial time, and T is the final time for the time horizon. The **ReachAvoid** set can be thought of as a set of initial conditions from which there exists a control input to the system such that, for all disturbances acting on the system, the state at the final time T will be inside the **Target** set while never entering the **Avoid** set for any time in the time horizon.

The **ReachAvoid** set given in (2.6) has a corresponding value function $V : \mathbb{R}^n \times [0, T] \to \mathbb{R}$ given by

$$V(z,t) = \inf_{u \in \mathcal{U}_{[t,T]}} \sup_{v \in \mathcal{V}_{[t,T]}} \max\{l(z(T),\ \max_{\tau \in [t,T]}\{h(z(\tau)\}\} \tag{2.7}$$

and has been shown to be the unique viscosity solution of the quasi-variational inequality given by [3]

$$\max\{h(z) - V(z,t),\ \tfrac{\partial V}{\partial t}(z,t) + \sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} \tfrac{\partial V}{\partial z}(z,t) f(z,u,v)\} = 0 \tag{2.8}$$

with terminal condition $V(z,T) = \max\{l(z), h(z)\}$. Finally, the **ReachAvoid** set given in (2.6) can be related to the level set of the value function.

$$\textbf{ReachAvoid}(t, \textbf{Target}, \textbf{Avoid}) = \{z \in \mathcal{Z} \mid V(z,t) \leq 0\} \tag{2.9}$$

The Hamiltonian H for the system is defined as

$$H(z,p) = \sup_{v \in \mathcal{V}} \inf_{u \in \mathcal{U}} p^T f(z,u,v) \tag{2.10}$$

with costate vector $p \in \mathbb{R}^n = \frac{\partial V}{\partial z}$. The optimal control and disturbance inputs become

$$u^*(t) = \arg\inf_{u \in \mathcal{U}} \sup_{v \in \mathcal{V}} H(z,p,u,v) \tag{2.11}$$

$$v^*(t) = \arg\sup_{v \in \mathcal{V}} H(z,p,u^*,v) \tag{2.12}$$

In this case the controller is acting first and the disturbance can react to the control input. This will ensure the safety of the vehicle for any value of the disturbance. In our problem, however, we will not be considering a disturbance and will only be considering the controlled input.

In addition to the ReachAviod set, reachability analysis can characterize other useful sets of initial conditions for the system with unique relationships to relevant target or avoid sets. The following are a few of these sets which will be useful for us later on. We begin with the minimal Backwards Reach Set (BRS) which is defined as

$$\mathbf{Reach}_T^{\flat}(\mathcal{K}) = \{x \in \mathcal{X} \mid \forall u \in \mathcal{U},\ x(T) \in \mathcal{K}\} \tag{2.13}$$

and represents the set of states from which for all control input the state at final time T will be in the set $\mathcal{K}$. The maximal BRS can be defined as

$$\mathbf{Reach}_T^{\sharp}(\mathcal{K}) = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U},\ x(T) \in \mathcal{K}\} \tag{2.14}$$

and represents the set of states from which there exists a control input such that at final time T the state will be in the set $\mathcal{K}$. Furthermore, we can define the minimal Backwards Reach Tube (BRT) as

$$\mathbf{Reach}_{[0,T]}^{\flat}(\mathcal{K}) = \{x \in \mathcal{X} \mid \forall u \in \mathcal{U},\ \exists s \in [0,T],\ x(s) \in \mathcal{K}\} \tag{2.15}$$

which represents the set of states for which under all control inputs the state will end up in the set $\mathcal{K}$ at some time s in the time horizon. Similarly, we define the maximal BRT as

$$\mathbf{Reach}_{[0,T]}^{\sharp}(\mathcal{K}) = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U},\ \exists s \in [0,T],\ x(s) \in \mathcal{K}\} \tag{2.16}$$

which represents the set of states for which there exists a control such that the state will be in the set $\mathcal{K}$ at some time s in the time horizon. The maximal BRT can

be related to the maximal BRS through the following [2].

$$\mathbf{Reach}^{\sharp}_{[0,T]}(\mathcal{K}) = \bigcup_{s\in[0,T]} \mathbf{Reach}^{\sharp}_{s}(\mathcal{K}) \tag{2.17}$$

If $\forall s \in [0,T]$, $\mathbf{Reach}^{\flat}_{s}(\mathcal{K}) \neq \emptyset$ then the minimal BRT can be related to the minimal BRS through the following [18].

$$\mathbf{Reach}^{\flat}_{[0,T]}(\mathcal{K}) = \bigcup_{s\in[0,T]} \mathbf{Reach}^{\flat}_{s}(\mathcal{K}) \tag{2.18}$$

We now define the Viability Kernel as

$$\mathrm{Viab}_{[0,T]}(\mathcal{K}) = \{x \in \mathcal{X} \mid \exists u \in \mathcal{U},\ \forall s \in [0,T],\ x(s) \in \mathcal{K}\} \tag{2.19}$$

which is the set of states for which there exists a control input such that the state remains in the set $\mathcal{K}$ for all time in the time horizon. We define the Invariance Kernel as

$$\mathrm{Inv}_{[0,T]}(\mathcal{K}) = \{x \in \mathcal{X} \mid \forall u \in \mathcal{U},\ \forall s \in [0,T],\ x(s) \in \mathcal{K}\} \tag{2.20}$$

which is the set of states for which for all control inputs the state will remain in the set $\mathcal{K}$ for the entire time horizon. The viability kernel can be related to the minimal BRT by

$$\mathrm{Viab}_{[0,T]}(\mathcal{K}^c) = \left(\mathbf{Reach}^{\flat}_{[0,T]}(\mathcal{K})\right)^c \tag{2.21}$$

and the the invariance kernel can be related to the maximal BRT by

$$\mathrm{Inv}_{[0,T]}(\mathcal{K}^c) = \left(\mathbf{Reach}^{\sharp}_{[0,T]}(\mathcal{K})\right)^c \tag{2.22}$$

In the following sections and chapters of this thesis we will use these sets and their relations to develop a conservative Reach-Avoid set.

2.4 Method For Computing Reach-Avoid For the Two-Vehicle Problem

The current numerical methods for computing the Reach-Avoid set become computationally intractable as the system dimensions become larger and larger. The amount of time it takes to calculate the Reach-Avoid set for the four dimension CWH system becomes unreasonable. In this section, we will develop a computationally tractable solution for computing the Reach-Avoid set for the four dimension CWH system. This method is developed for the two-vehicle problem where the chaser satellite is attempting to reach a target while staying outside and unsafe region in the state space. This method can be extended to higher dimension systems as well.

We begin with our state state $z(t) \in \mathbb{R}^4 = [x, \dot{x}, y, \dot{y}]^T$ from Section 2.1 with dynamics, $\dot{z}(t)$, the same as in (2.1) with the same control policy $u \in \mathcal{U}$. Additionally, we will assume there is no disturbance acting on the system. For the sake of simplicity, we will assume we have a general **Target** set we would like to reach and a general **Avoid** set we would like to avoid, as we did in Section 2.3. We now assume that we can write the **Target** and **Avoid** sets as the following

$$\begin{aligned} \textbf{Target} &= \{z \in \mathcal{Z} \mid c_R^T z \leq d_R\} \\ \textbf{Avoid} &= \{z \in \mathcal{Z} \mid c_A^T z > d_A\} \end{aligned} \tag{2.23}$$

Furthermore, this allows us to write the $\textbf{Avoid}^c$ set as $\textbf{Avoid}^c = \{z \in \mathcal{Z} \mid c_A^T z \leq d_A\}$. The **Target** and **Avoid** sets given in (2.23) can be related to the **Target** and **Avoid** sets defined in (2.5) by letting $l(x) = \max\{c_R^T z \leq d_R\}$ and $h(x) = \max\{c_A^T z > d_A\}$. We note that under the assumption our **Target** and **Avoid** sets can be written in the form given by (2.23) we have also assumed that our **Target** set and $\textbf{Avoid}^c$ sets are convex.

Next, we rewrite the **ReachAvoid** set from (2.6) for a system without distur-

bance and in terms of the $\mathbf{Avoid}^c$ set.

$$\begin{aligned}\mathbf{ReachAvoid}(t, \mathbf{Target}, \mathbf{Avoid}) = \{z \in \mathcal{Z} \mid \exists u \in \mathcal{U}_{[t,T]}, \\ (z(T) \in \mathbf{Target}) \wedge (\forall \tau \in [t, T]\ z(\tau) \in \mathbf{Avoid}^c\}\end{aligned} \tag{2.24}$$

which has the corresponding value function

$$V(z,t) = \inf_{u \in \mathcal{U}_{[t,T]}} \max\{l(z(T),\ \max_{\tau \in [t,T]}\{h(z(\tau)\}\} \tag{2.25}$$

given by the unique viscosity solution of the quasi-variational inequality

$$\max\{h(z) - V(z,t),\ \tfrac{\partial V}{\partial t}(z,t) + \inf_{u \in \mathcal{U}} \tfrac{\partial V}{\partial z}(z,t) f(z,u,v)\} = 0 \tag{2.26}$$

The **ReachAvoid** for a system without disturbance is still related to its value function $V(z,t)$ through the relationship given in (2.9). The optimal control for the system without disturbance is given by

$$u^*(t) = \arg\inf_{u \in \mathcal{U}} \tfrac{\partial V}{\partial z}(z,t)(f(z,u)) \tag{2.27}$$

As in Section 2.3 we define the costate of our system as $p = \frac{\partial V}{\partial z}$. The costate dynamics are given by

$$\dot{p} = \begin{cases} -A^T p & \text{for } \dfrac{\partial V}{\partial t}(z,t) = -\inf\limits_{u \in \mathcal{U}} \dfrac{\partial V}{\partial z}(z,t)(f(z,u)) \\ 0 & \text{for } h(z) > 0 \end{cases} \tag{2.28}$$

With this, if a state is to enter the **Avoid** set, i.e. $h(z) > 0$, the evolution of the costate is stopped. This prevents a state from potentially moving into, and then out of, the **Avoid** set. Following the dynamics in (2.1) for $f(z,u)$ and writing the state and costate as $z = [z_1, z_2, z_3, z_4,]^T$ $p = [p_1, p_2, p_3, p_4,]^T$, the Hamiltonian for the

CWH dynamics can then be written as

$$\begin{aligned} H(z,p) &= \inf_{u\in\mathcal{U}} p^T(f(z,u)) \\ &= \inf_{u\in\mathcal{U}} \Big(p_1 z_2 + p_2\Big(3\omega^2 z_1 + 2\omega z_4 + \tfrac{1}{m_c}u_1\Big) + p_3 z_4 \\ &\quad + p_4\Big(-2\omega z_2 + \tfrac{1}{m_c}u_2\Big)\Big) \\ &= p_1 z_2 + p_2\Big(3\omega^2 z_1 + 2\omega z_4 - \tfrac{u_{\max}}{m_c}\mathrm{sgn}(p_2)\Big) \\ &\quad + p_3 z_4 + p_4\Big(-2\omega z_2 - \tfrac{u_{\max}}{m_c}(\mathrm{sgn}(p_4))\Big) \end{aligned} \tag{2.29}$$

The optimal control input becomes

$$\begin{aligned} u_1^* &= -\mathrm{sgn}(p_2)\cdot u_{\max} \\ u_2^* &= -\mathrm{sgn}(p_4)\cdot u_{\max} \end{aligned} \tag{2.30}$$

We note that the optimal control is a function of the costate and, due to the signum function, is undefined when $p_2,\, p_4 = 0$. We now develop a method to handle cases when $p_2\, p_4 = 0$. We first introduce two switching functions given by the following

$$\begin{aligned} s_1(t) &\triangleq p_2(t) \\ s_2(t) &\triangleq p_4(t) \end{aligned} \tag{2.31}$$

Then, using the dynamics for the costate given in (2.28), the time derivatives of the switching functions are given by the following

$$\begin{aligned} \dot{s}_1 &= -(p_1 - 2\omega p_4) \\ \dot{s}_2 &= -(2\omega p_2 + p_3) \end{aligned} \tag{2.32}$$

In the case that $s_1(t) = 0$, if $\dot{s}_1(t) > 0$ then at a time, t^-, just before t, $s_1(t^-) < 0$. We then use this as our choice for the sign of p_2, since the optimal control is only dependent on the sign of the costate. We take $p_2(t) < 0$ and, from (2.30), the optimal control becomes $u_1^*(t) = u_{\max}$. Alternatively, if $\dot{s}_1(t) < 0$ then at a time, t^-, just before t, $s_1(t^-) > 0$. We then take $p_2(t) > 0$ and the optimal control becomes

$u_1^*(t) = -u_{\max}$. We formally define the optimal control when $s_1 = 0$ or $s_2 = 0$ as

$$\begin{aligned} u_1^* &= -\text{sgn}(p_1 - 2\omega p_4) \cdot u_{\max} \quad \text{for } s_1(t) = 0 \\ u_2^* &= -\text{sgn}(2\omega p_2 + p_3) \cdot u_{\max} \quad \text{for } s_2(t) = 0 \end{aligned} \tag{2.33}$$

In the event that $s_1 = 0$ and $\dot{s}_1 = 0$, or $s_2 = 0$ and $\dot{s}_2 = 0$, this method is repeated using the second derivatives of the switching functions s_1 and s_2. The second time derivatives of the switching functions are given by

$$\begin{aligned} \ddot{s}_1(t) &= -\omega(\omega p_2 + 2p_3) \\ \ddot{s}_2(t) &= -2\omega(-p_1 + 2\omega p_4) \end{aligned} \tag{2.34}$$

The optimal control becomes

$$\begin{aligned} u_1^* &= -\text{sgn}(\omega p_2 + 2p_3) \cdot u_{\max} \quad \text{for } \dot{s}_1(t) = 0 \\ u_2^* &= -\text{sgn}(-p_1 + 2\omega p_4) \cdot u_{\max} \quad \text{for } \dot{s}_2(t) = 0 \end{aligned} \tag{2.35}$$

If needed, the method can be repeated for higher derivatives.

We will compute the **ReachAvoid** set backwards in time, that is, the initial values for our calculations will represent the desired values for the state of our system at the final time, $z(T)$. Clearly, for any state to be in the **ReachAvoid** set it must be in the **Target** and $\textbf{Avoid}^c$ sets at the final time T. Hence, the **ReachAvoid**(T) is given by the intersection of the **Target** and $\textbf{Avoid}^c$ sets and can be described by the polytope which satisfies the inequality $c^T z \leq d$, where $c = [\, c_R \;\; c_A \,]$ and $d = [\, d_R^T \;\; d_A^T \,]^T$. Furthermore, the **ReachAvoid** set will only grow from the "usable part" of the boundary of the final time **ReachAvoid**(T) set. Therefore, the initial **ReachAvoid**(T) set can further be constrained by the boundary of its "usable part" for a specific facet j through the following inequality

$$\frac{\partial V_j}{\partial z}(Az + Bu^*) \leq 0 \tag{2.36}$$

Where V_j is the value function which corresponds to propagating the j^{th} facet of the **ReachAvoid**(T). More specifically, V_j is the solution to (2.26) whose final

costate is given by $p(T) = c_j$, or the j^{th} column vector of c. A single facet of $\mathbf{ReachAvoid}(T)$ is defined as $c_j^T z = d_j$ where c_j is the j^{th} column vector in c and d_j is the corresponding j^{th} entry of d.

We now have an initial set of states for a given facet j from which the **ReachAvoid** set can grow backwards in time. This set can formally be defined for a given facet j as follows

$$\mathcal{P}_j = \left\{ z \;\middle|\; \begin{array}{rcl} c_R^T z & \leq & d_R \\ c_A^T z & \leq & d_A \\ c_j^T A z & \leq & -c_j^T B u^* \\ c_j^T z & = & d_j \end{array} \right\} \tag{2.37}$$

The $\mathbf{ReachAvoid}(t)$ for a facet j can then be computed by taking the vertices $v_j^i(T)$ of $\mathcal{P}_j$ and evolving them backwards in time using the dynamics given in (2.1), (2.28), and (2.26). The convex hull, denoted by $\text{conv}(\cdot)$, is then taken of all the vertices at time t to give the boundary of the $\mathbf{ReachAvoid}(t)$.

$$\partial\mathbf{ReachAvoid}(t) = \bigcup_j \text{conv}_i \left(v_j^i(t) \right) \tag{2.38}$$

It is important to note that this method takes advantage of the fact that the $\mathbf{ReachAvoid}(T)$ is convex, the dynamics of the state and costate are switch linear dynamics, and that the control set is compact.

Ideally, we would like to apply this method directly to the debris avoidance problem with the $\mathbf{Target}_z$ set as our **Target** and the $\overline{\mathbf{Avoid}}_d(s)$ set as our **Avoid** set. However, because we have defined our $\mathbf{Avoid}_{d,i}(s)$ sets as bounding boxes, their complements, $\mathbf{Avoid}_{d,i}^c(s)$, become non convex sets. Hence, these methods can not be directly applied to the debris avoidance Reach-Avoid problem. We will, however, apply these methods for computing the minimal and maximal BRS of a single $\mathbf{Avoid}_{d,i}(s)$ set and the $\mathbf{Target}_z$ set. The following subsection outlines the extension of this method from the full Reach-Avoid set to a single reach set.

2.4.1 Method for Computing Maximal Reach Set

We begin with the extension of the methods in Section 2.4 to the maximal BRS given in equation (2.14). For the sake of simplicity, we will assume that we have a general **Target** set we are trying to reach which can be written in the form given in (2.23). We will then compute $\mathbf{Reach}_T^\sharp(\mathbf{Target})$. If we take the definition for the **ReachAvoid** given by equation (2.6) and make the following assumptions: 1) There is no **Avoid** set, and 2) There is no disturbance acting on the system, we see that the full **ReachAvoid** set simply becomes the maximal BRS. Furthermore, applying these assumptions to equations (2.7) and (2.8) we see that the value function corresponding to the maximal BRS is given by

$$V^\sharp(z,t) = \inf_{u\in\mathcal{U}_{[t,T]}} l(z(T)) \tag{2.39}$$

where $l(z) = c_{R,z}^T z - d_{R,z}$. The value function $V^\sharp(z,t)$ is the unique viscosity solution of the quasi-variational inequality given by

$$\frac{\partial V^\sharp}{\partial t}(z,t) + \inf_{u\in\mathcal{U}} \frac{\partial V^\sharp}{\partial z}(z,t) f(z,u) = 0 \tag{2.40}$$

Then, from equation (2.12), the optimal control for the maximal BRS is given by

$$u^*(t) = \arg\inf_{u\in\mathcal{U}} \frac{\partial V^\sharp}{\partial z}(z,t)(f(z,u)) \tag{2.41}$$

We note that the optimal control for this maximal BRS is the same as the optimal control for the **ReachAvoid** set without disturbance. Finally, from equation (2.9), we have the maximal BRS related to the value function, $V^\sharp$, through the inequality $\mathbf{Reach}_T^\sharp(\mathcal{K}) = \{z \in \mathcal{Z} \mid V^\sharp(z,t) \leq 0\}$.

We again define our costate, p, with dynamics similar to those given in (2.28). We note that while we still have sets we are trying to avoid, $\mathbf{Avoid}_{d,i}(s)$, we are simply calculating a reach set. Unlike in the **ReachAvoid** set calculation our state

evolution will never be halted because there are no constraints to violate in the reach calculation. Because our state evolution is never halted, our costate never needs to be "frozen", therefore, the costate dynamics simply reduce to $\dot{p} = -A^T p$. The Hamiltonian for the maximal BRS is the same as that for the full **ReachAvoid** set and is given by

$$\begin{aligned} H^\sharp(z,p) &= \inf_{u\in\mathcal{U}} p^T(f(z,u)) \\ &= \inf_{u\in\mathcal{U}} \Big(p_1 z_2 + p_2\Big(3\omega^2 z_1 + 2\omega z_4 + \tfrac{1}{m_c}u_1\Big) + p_3 z_4 \\ &\quad + p_4\Big(-2\omega z_2 + \tfrac{1}{m_c}u_2\Big)\Big) \\ &= p_1 z_2 + p_2\Big(3\omega^2 z_1 + 2\omega z_4 - \tfrac{u_{\max}}{m_c}\mathrm{sgn}(p_2)\Big) \\ &\quad + p_3 z_4 + p_4\Big(-2\omega z_2 - \tfrac{u_{\max}}{m_c}\mathrm{sgn}(p_4)\Big) \end{aligned} \tag{2.42}$$

Therefore, the optimal control for the maximal BRS is the same as the the optimal control for the full **ReachAvoid** set without disturbance and is given by

$$\begin{aligned} u_1^{\sharp,*} &= -\mathrm{sgn}(p_2)\cdot u_{\max} \\ u_2^{\sharp,*} &= -\mathrm{sgn}(p_4)\cdot u_{\max} \end{aligned} \tag{2.43}$$

As with the optimal control for the full **ReachAvoid** set, the optimal control for the maximal BRS is a function of the signum of the costate and is undefined when $p_2,\ p_4 = 0$. Because of the similarities between the maximal BRS and the **ReachAvoid** without disturbance the optimal control for the maximal BRS when $p_2,\ p_4 = 0$ can be found by following equations (2.31) through (2.35).

We then define the final time polytope for the maximal reach calculation of the **Target** set from which the vertices will be evolved backwards in time to create the maximal BRS. Using the boundary of the usable part given by equation (2.36) the final time polytope for a single facet of the set is defined by

$$\mathcal{P}_j^\sharp = \left\{ z \;\middle|\; \begin{array}{rcl} c_R^T z &\leq& d_R \\ c_j^T A z &\leq& -c_j^T B u^{\sharp,*} \\ c_j^T z &=& d_j \end{array} \right\} \tag{2.44}$$

Where c_j is the j^{th} column vector of c_R and d_j is the j^{th} entry of d_R. Finally, we take the vertices $v_j^{\sharp,i}(T)$ of $\mathcal{P}_j^{\sharp}$ and evolve them backwards in time according to the costate dynamics and the CWH state dynamics to achieve the boundary for the maximal BRS of a **Target** set given by

$$\partial \mathbf{Reach}_t^{\sharp}(\mathbf{Target}) = \bigcup_j \mathrm{conv}_i \left(v_j^{\sharp,i}(t) \right) \tag{2.45}$$

2.4.2 Method For Computing Minimal Reach Set

We can also extend the methods in Section 2.4 to the minimal BRS of a general target set **Target**, with the minimal BRS given by (2.13). In other words, we are trying to compute the $\mathbf{Reach}_T^{\flat}(\mathbf{Target})$ set. Following the framework from 2.4.1, we take the definition for the **ReachAvoid** set given by equation (2.6) and make the following three assumption: 1) There is no **Avoid** set, 2) There is no disturbance action of the system, and 3) our control input u is represented as the disturbance v in (2.6). Under these assumptions we see that the full **ReachAvoid** set simply becomes the minimal BRS. We again apply these new assumptions to (2.7) and (2.8) and the corresponding value function for the minimal BRS becomes

$$V^{\flat}(z,t) = \sup_{u \in \mathcal{U}_{[t,T]}} l(z(T)) \tag{2.46}$$

where $l(z) = c_R^T z - d_R$. The value function $V^{\flat}(z,t)$ is the unique viscosity solution of the quasi-variational inequality given by

$$\frac{\partial V^{\flat}}{\partial t}(z,t) + \sup_{u \in \mathcal{U}} \frac{\partial V^{\flat}}{\partial z}(z,t) f(z,u) = 0 \tag{2.47}$$

Then, from equation (2.12), the optimal control for the minimal BRS is given by

$$u^{\flat,*}(t) = \arg\sup_{u \in \mathcal{U}} \frac{\partial V^{\flat}}{\partial z}(z,t)(f(z,u)) \tag{2.48}$$

Finally, from equation (2.9), we have the maximal BRS related to the value function $V^\flat$ through the inequality $\mathbf{Reach}_T^\flat(\mathcal{K}) = \{z \in \mathcal{Z} \mid V^\flat(z,t) \leq 0\}$.

We again have our costate p with dynamics $\dot{p} = -A^T p$ because we are only performing a reach calculation. The Hamiltonian for the minimal BRS becomes

$$\begin{aligned} H^\flat(z,p) &= \sup_{u\in\mathcal{U}} p^T(f(z,u)) \\ &= \sup_{u\in\mathcal{U}} \Big(p_1 z_2 + p_2\Big(3\omega^2 z_1 + 2\omega z_4 + \tfrac{1}{m_c}u_1\Big) + p_3 z_4 \\ &\quad + p_4\Big(-2\omega z_2 + \tfrac{1}{m_c}u_2\Big)\Big) \\ &= p_1 z_2 + p_2\Big(3\omega^2 z_1 + 2\omega z_4 + \tfrac{u_{\max}}{m_c}\mathrm{sgn}(p_2)\Big) \\ &\quad + p_3 z_4 + p_4\Big(-2\omega z_2 + \tfrac{u_{\max}}{m_c}\mathrm{sgn}(p_4)\Big) \end{aligned} \tag{2.49}$$

Therefore, the optimal control for the minimal BRS is given by

$$\begin{aligned} u_1^{\flat,*} &= \mathrm{sgn}(p_2)\cdot u_{\max} \\ u_2^{\flat,*} &= \mathrm{sgn}(p_4)\cdot u_{\max} \end{aligned} \tag{2.50}$$

As with the optimal control for the full **ReachAvoid** set and the maximal BRS, the optimal control for the minimal BRS is a function of the signum of the costate and is undefined when $p_2,\ p_4 = 0$. Because of the similarities between the minimal BRS and the **ReachAvoid** without disturbance the optimal control for the minimal BRS when $p_2,\ p_4 = 0$ can be found by following the same steps in equations (2.31) through (2.35).

We then define the final time polytope for the minimal reach calculation of the **Target** set from which the vertices will be evolved backwards in time to create the minimal BRS. Using the boundary of the usable part given by equation (2.36) the final time polytope for a single facet of the minimal BRS is defined by

$$\mathcal{P}_j^\flat = \left\{ z \;\middle|\; \begin{array}{rcl} c_R^T z &\leq& d_R \\ c_j^T A z &\leq& -c_j^T B u^{\flat,*} \\ c_j^T z &=& d_j \end{array} \right\} \tag{2.51}$$

Finally, we take the vertices $v_j^{\flat,i}(T)$ of $\mathcal{P}_j^\flat$ and evolve them backwards in time according to the costate dynamics and the CWH state dynamics to achieve the boundary for the minimal BRS of a **Target** set given by

$$\partial\mathbf{Reach}_t^\flat(\mathbf{Target}) = \bigcup_j \mathrm{conv}_i \left(v_j^{\flat,i}(t) \right) \tag{2.52}$$

Chapter 3

Multiple Debris Avoidance Through System Decomposition

Recent work in [18] has shown a method to compute exact minimal and maximal BRS for high dimension systems through system decomposition. We can apply these methods, along with the methods from Sections 2.4.1 and 2.4.2, to generate and under-approximation of the Reach-Avoid set for the multiple debris scenario. We note that for each piece of debris, our system increases by four dimensions for full system dimension of $4q$ for q pieces of debris. The following methods will demonstrate how the larger dimension avoid problem can be decomposed into multiple smaller dimension avoid problems.

3.1 System Decomposition Overview

This section will give a brief overview of the methods developed in [18] and their main contributions. We begin with a system $z(t)$ which can be partitioned into the

following

$$\begin{gathered} z = (y_1, y_2, y_3) \\ y_1 \in \mathbb{R}^{n_1}, y_2 \in \mathbb{R}^{n_2}, y_3 \in \mathbb{R}^{n_3} \\ n_1, n_2 > 0, n_3 \geq 0 \\ n_1 + n_2 + n_3 = n \end{gathered} \tag{3.1}$$

New subsystems $x_1 \in \mathcal{X}_1 = \mathbb{R}^{n_1+n_3}$, $x_2 \in \mathcal{X}_2 = \mathbb{R}^{n_2+n_3}$ are then created as follows

$$\begin{aligned} x_1 &= (y_1, y_3) \\ x_2 &= (y_2, y_3) \end{aligned} \tag{3.2}$$

It is important to note that, while it is only shown here for two subsystems, this decomposition can be applied to any number of finite subsystems. These subsystems are "self-contained subsystems" if the states x_i evolve independently of each other; that is

$$\begin{aligned} \tfrac{dx_1}{dt} &= \dot{x}_1 = f_1(x_1, u) \\ \tfrac{dx_2}{dt} &= \dot{x}_2 = f_2(x_2, u) \end{aligned} \tag{3.3}$$

We now define projection and back projection operators to relate states, and sets of states, from the full system to each of the subsystems and vice versa. The projection operator for a state z onto a subsystem state space $\mathcal{X}_i$ is defined by

$$\text{proj}_{\mathcal{X}_i}(z) = x_i,\ i = 1, 2 \tag{3.4}$$

The projection operator for sets $\mathcal{S} \subseteq \mathcal{Z}$ is defined by

$$\text{proj}_{\mathcal{X}_i}(\mathcal{S}) = \{x_i \in \mathcal{X}_i \mid \exists z \in \mathcal{S},\ \text{proj}_{\mathcal{X}_i}(z) = x_i\} \tag{3.5}$$

The back projection of a state x_i onto the $\mathcal{Z}$ state space is defined by

$$\text{proj}^{-1}(x_i) = \{z \in \mathcal{Z} \mid \text{proj}_{\mathcal{X}_i}(z) = x_i\} \tag{3.6}$$

The back projection of a set $\mathcal{S}_i \subseteq \mathcal{X}_i$, with abuse of notation, is given by

$$\text{proj}^{-1}(\mathcal{S}_i) = \{z \in \mathcal{Z} \mid \exists x_i \in \mathcal{S}_i, \text{proj}_{\mathcal{X}_i}(z) = x_i\} \tag{3.7}$$

.

Figure 3.1 shows an example of the inverse projections of lower dimensions sets into the full dimension. In this example, we will consider our full state z to be two dimensional consisting of two, single dimension, sub systems represented by y_1 and y_2.

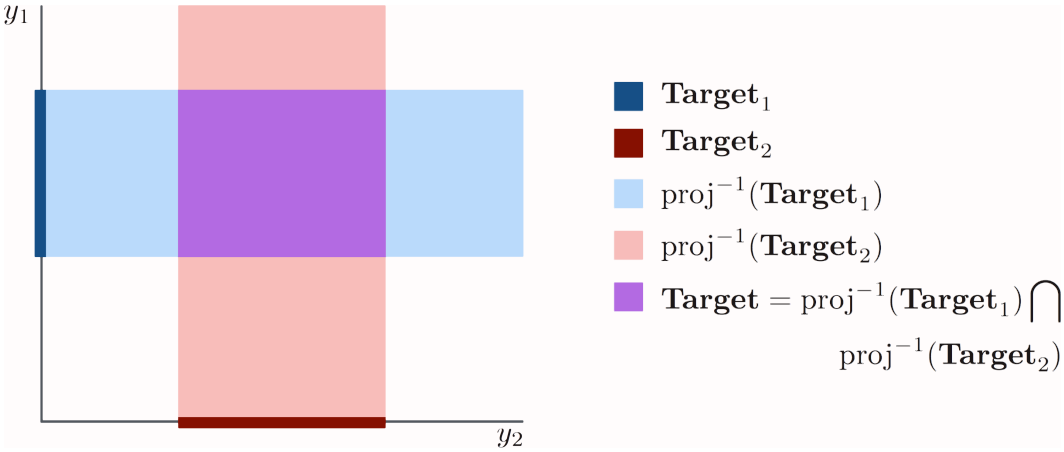

Figure 3.1: Example of inverse projections of sub system sets into full dimension sets

The main idea of the system decomposition method is to compute the full dimension reach set of a target by decoupling both the system dynamics and the target sets into lower dimension systems. The reach calculation is then performed independently on these lower dimension systems and target sets and then the full dimension reach set is reconstructed through intersections or unions of the lower dimension reach sets. The main contributions of [18] are the following two statements relating the full dimension minimal and maximal BRSs to lower dimension minimal and maximal BRSs under specific assumptions about how the full dimension target sets

are decoupled into the lower dimensions.

$$\begin{gathered}\mathbf{Target} = \mathrm{proj}^{-1}(\mathbf{Target}_1) \bigcup \mathrm{proj}^{-1}(\mathbf{Target}_2) \Rightarrow \\ \mathbf{Reach}_T^{\sharp}(\mathbf{Target}) = \mathrm{proj}^{-1}(\mathbf{Reach}_T^{\sharp}(\mathbf{Target}_1)) \bigcup \mathrm{proj}^{-1}(\mathbf{Reach}_T^{\sharp}(\mathbf{Target}_2))\end{gathered} \tag{3.8}$$

$$\begin{gathered}\mathbf{Target} = \mathrm{proj}^{-1}(\mathbf{Target}_1) \bigcap \mathrm{proj}^{-1}(\mathbf{Target}_2) \Rightarrow \\ \mathbf{Reach}_T^{\flat}(\mathbf{Target}) = \mathrm{proj}^{-1}(\mathbf{Reach}_T^{\flat}(\mathbf{Target}_1)) \bigcap \mathrm{proj}^{-1}(\mathbf{Reach}_T^{\flat}(\mathbf{Target}_2))\end{gathered} \tag{3.9}$$

Statement (3.8) says that if our full dimensional target, **Target**, can be written as the union of inverse projections of lower dimension targets $\mathbf{Target}_1$ and $\mathbf{Target}_2$, then the maximal BRS for the full dimension target is the union of the inverse projection of lower dimension maximal BRS of the lower dimension targets. In other words, we can compute the exact full dimension maximal BRS by independently computing the maximal BRS for the lower dimension subsystems and their corresponding target sets. Hence, the numerical calculations can be performed on the lower dimension subsystems making computationally tractable solutions possible for higher dimension systems. Figure 3.2 shows an example of statement (3.8) for a full state in two dimensions and subsystems in one dimension. It follows the same color key in figure 3.1 for the full dimension and subsystem targets.

Statement (3.9) is similar, however, it applies to the case where the full dimension target, **Target**, is written as an intersection of inverse projections of lower dimension targets $\mathbf{Target}_1$ and $\mathbf{Target}_2$. In this case, the minimal BRS can be computed by taking the intersections of the inverse projections of the minimal BRS for the lower dimension target sets. Figure 3.3 shows an example of statement (3.9) for a full state in two dimensions and subsystems in one dimension. It follows the same color key in figure 3.1 for the full dimension and subsystem targets.

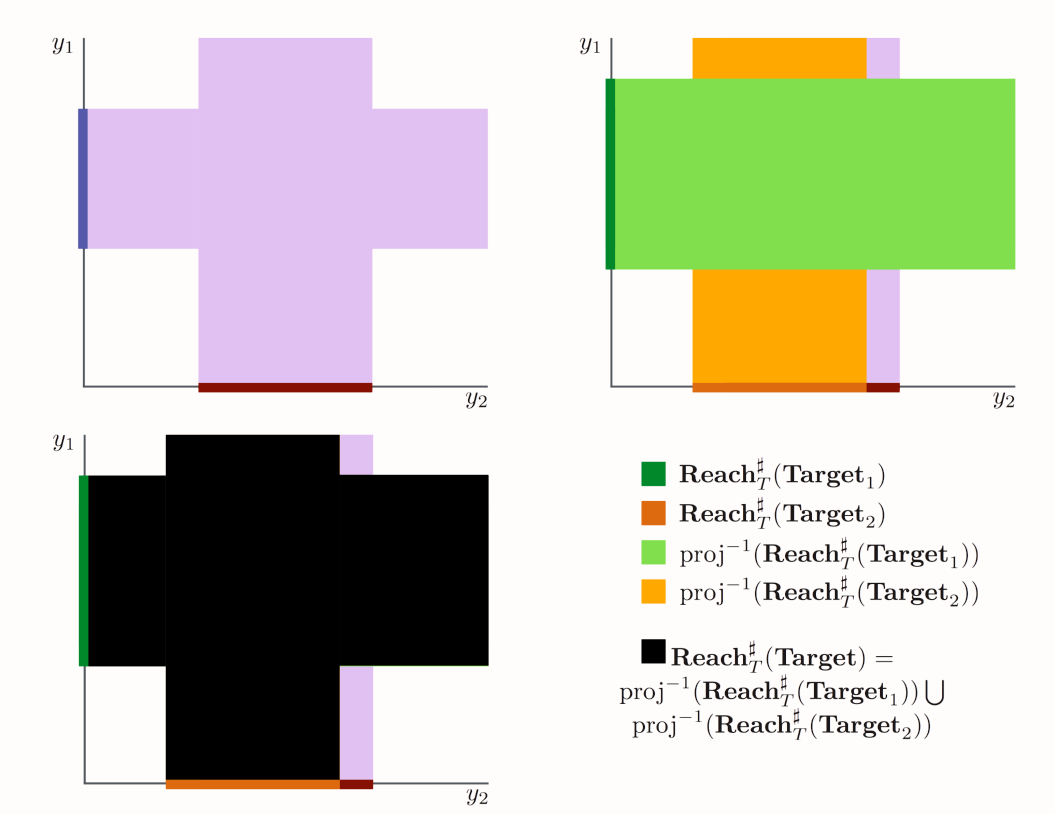

Figure 3.2: Example of decomposition applied to maximal reach

3.2 System Decomposition Applied to Multiple Debris Avoidance

We will now apply the methods for system decomposition to the problem of guaranteeing the safety of a satellite in the presence of multiple pieces of debris. For this calculation we are only concerned with the satellite's ability to avoid all the debris for all time, and will not be concerned with whether or not it reaches a specific target at the final time.

We begin by defining a new state $z_{r,i} = [x_{r,i}, \dot{x}_{r,i}, y_{r,i}, \dot{y}_{r,i}]^T \in \mathbb{R}^4$ which will rep-

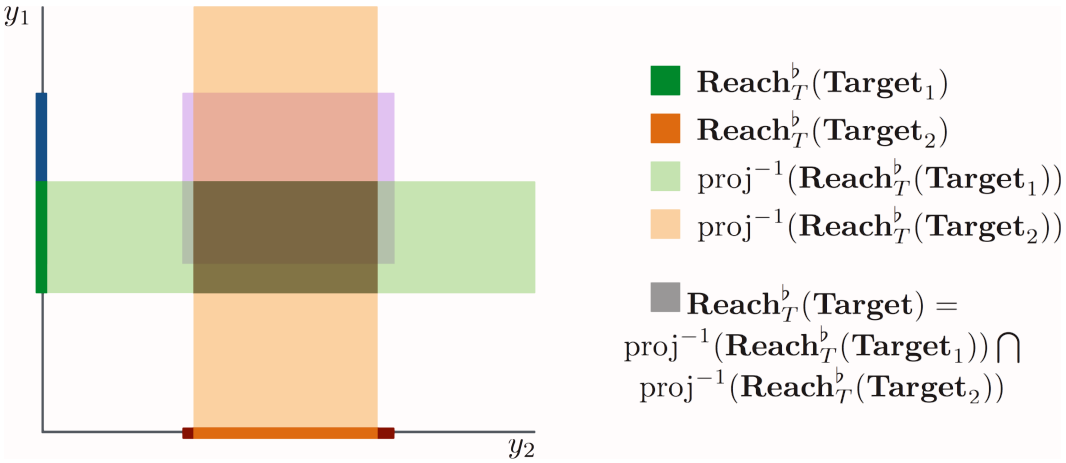

Figure 3.3: Example of decomposition applied to minimal reach

resent the relative position and velocity of the chaser with respect to the i^{th} piece of debris. In the $z_{r,i}$ coordinate frame the debris is fixed at the origin while the chaser satellite and chief point move around it. Figure 3.4 shows a two dimension representation of the z, $z_{d,i}$, and $z_{r,i}$ states for a single piece of debris.

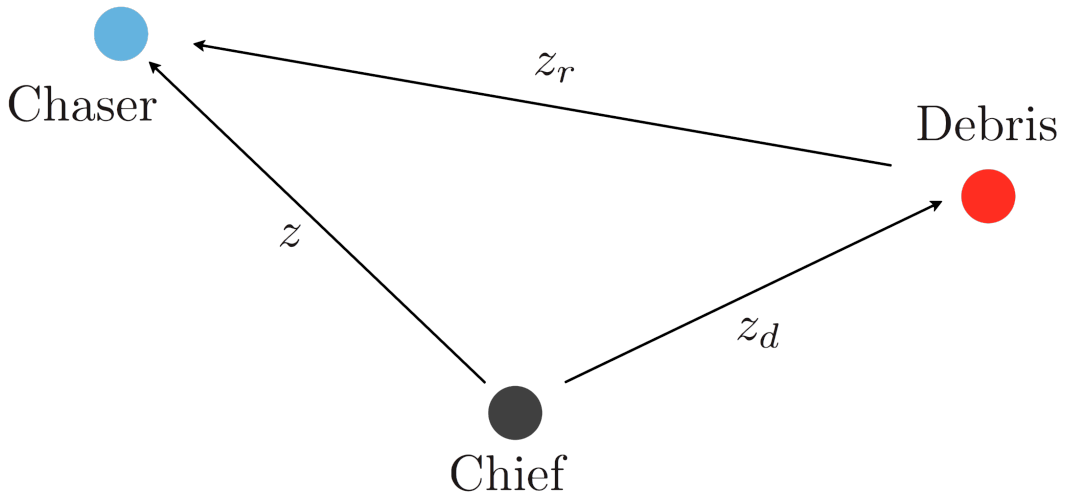

Figure 3.4: Diagram of relative coordinates between chief, chaser, and single debris

The state $z_{r,i}$ can be related to the previously defined states z and $z_{d,i}$ through the equation $z_{r,i} = z - z_{d,i}$. The state dynamics for the $z_{r,i}$ state are then given by the following

$$\begin{aligned} \dot{z}_{r,i}(t) &= \dot{z}(t) - \dot{z}_{d,i}(t) \\ &= Az(t) + Bu(t) - Az_{d,i}(t) \\ &= A(z(t) - z_{d,i}(t)) + Bu(t) \\ &= Az_{r,i}(t) + Bu(t) \end{aligned} \tag{3.10}$$

We note that under our assumption that the chief, chaser satellite, and every piece of debris are near the same circular orbit, that the A and B matrices will be the same for every piece of debris and will also be the same as the A and B matrices defined in the original CWH equations for the chaser (2.1). We now define an augmented state $\hat{z} = [z_{r,1}, z_{r,2}, \ldots, z_{r,q}] \in \mathbb{R}^{4q}$ for an arbitrary number q pieces of debris. The dynamics for the $\hat{z}$ system are then given by

$$\begin{aligned} \dot{\hat{z}}(t) &= \hat{A}\hat{z}(t) + \hat{B}u(t) \\ &= \begin{bmatrix} A & 0 & \ldots & 0 \\ 0 & A & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & A \end{bmatrix} \hat{z}(t) + \begin{bmatrix} B \\ B \\ \vdots \\ B \end{bmatrix} u(t) \end{aligned} \tag{3.11}$$

where $\hat{z}$ can be decoupled into q independent subsystems each with dynamics given by (3.10). Following the system decomposition framework our full dimension system is represented by $\hat{z}$ and our self-contained subsystems are given by $z_{r,i}$. We will define the sets in the subsystems we wish to avoid as bounding boxes around the origin of each $z_{r,i}$ subsystem, $\mathbf{Avoid}_{d,i}$. The sets $\mathbf{Avoid}_{d,i}$ are similar to the previously defined sets $\mathbf{Avoid}_{d,i}(s)$ except constant in time. The $\mathbf{Avoid}_{d,i}$ sets are the lower dimension sets we will perform either maximal or minimal backwards reach calculations on.

We will now define a full dimension avoid set, $\mathbf{Avoid}_{\hat{z}}$, as either an intersection or

a union of all the subsystem avoid sets. From (3.8), if we define the full dimension set as a union of the lower dimension sets we will be required to compute the maximal BRS. Additionally, from (3.9), if we define the full dimension set as an intersection of the lower dimension sets we will be required to compute the minimal BRS. At this point, it is important to remind ourselves that we are interested in guaranteeing the safety of the satellite. To do so, we must guarantee that the satellite stays outside of every $\mathbf{Avoid}_{d,i}$ set for all time. This follows from the definition for the full Reach-Avoid set in (2.6). Referring back to Section 2.3 we note that only the viability and invariance kernels characterize a state for all time and hence, we will want to compute either the $\text{Viab}_{[0,T]}(\mathbf{Avoid}^c_{\hat{z}})$ or $\text{Inv}_{[0,T]}(\mathbf{Avoid}^c_{\hat{z}})$. We also note that the viability and invariance kernels can be related to the minimal and maximal BRTs, respectively. More specifically, the viability kernel of a set complement is the complement of the minimal BRT for the original set, and similarly for the invariance kernel and maximal BRT. Hence, if we define the $\mathbf{Avoid}_{\hat{z}}$ set as an intersection of lower dimension sets we will end up with the viability kernel of the complement of the $\mathbf{Avoid}_{\hat{z}}$ set, and if we define the $\mathbf{Avoid}_{\hat{z}}$ set as a union of lower dimension sets we will end up with the invariance kernel of the complement of the $\mathbf{Avoid}_{\hat{z}}$ set.

Let us first try defining the $\mathbf{Avoid}_{\hat{z}}$ set as an intersection of lower dimension sets, that is, $\mathbf{Avoid}_{\hat{z}} = \bigcap_{i=1}^{q} \text{proj}^{-1}(\mathbf{Avoid}_{d,i})$. The complement of the full dimension avoid set becomes $\mathbf{Avoid}^c_{\hat{z}} = \bigcup_{i=1}^{q} \text{proj}^{-1}(\mathbf{Avoid}^c_{d,i})$. Because we defined our full dimension set as an intersection we are required to calculate the minimal BRS, which will give us the viability kernel $\text{Viab}_{[0,T]}(\mathbf{Avoid}^c_{\hat{z}})$. Now, let us assume we have a state $\bar{\hat{z}}$ such that $\bar{\hat{z}} \in \text{Viab}_{[0,T]}(\mathbf{Avoid}^c_{\hat{z}})$. Then, by definition, $\exists u \in \mathcal{U}$, $\forall s \in [0,T]$, $\bar{\hat{z}}(s) \in \mathbf{Avoid}^c_{\hat{z}}$; which implies $\bar{\hat{z}} \in \bigcup_{i=1}^{q} \text{proj}^{-1}(\mathbf{Avoid}^c_{d,i})$. This only guarantees that one of the subsystem states $z_{r,i}$ is outside its respective avoid set for all time, not that every subsystem state is outside their respective avoid states. Therefore, writing our full dimension avoid set $\mathbf{Avoid}_{\hat{z}}$ as a an intersection of lower dimension sets will not ensure that the satellite will stay outside all the debris for all time.

We will now try defining our set $\mathbf{Avoid}_{\hat{z}}$ as the union of the lower dimension sets, $\mathbf{Avoid}_{\hat{z}} = \bigcup_{i=1}^{q} \text{proj}^{-1}(\mathbf{Avoid}_{d,i})$. The complement of the full dimension avoid set then becomes $\mathbf{Avoid}_{\hat{z}}^{c} = \bigcap_{i=1}^{q} \text{proj}^{-1}(\mathbf{Avoid}_{d,i}^{c})$. We must calculate the maximal BRS because we have defined our full dimension set as a union, which will lead to the invariance kernel $\text{Inv}_{[0,T]}(\mathbf{Avoid}_{\hat{z}}^{c})$. If we assume we have a state $\tilde{\hat{z}}$ such that $\tilde{\hat{z}} \in \text{Inv}_{[0,T]}(\mathbf{Avoid}_{\hat{z}}^{c})$ then, by definition, $\forall u \in \mathcal{U}$, $\forall s \in [0,T]$, $\tilde{\hat{z}}(s) \in \mathbf{Avoid}_{\hat{z}}^{c}$; which implies $\tilde{\hat{z}} \in \bigcap_{i=1}^{q} \text{proj}^{-1}(\mathbf{Avoid}_{d,i}^{c})$. This will guarantee that every subsystem state $z_{r,i}$ is also outside their respective avoid sets for all time. Hence, to guarantee the safety of the chaser satellite for the multiple debris avoidance problem we will define our full dimension avoid set as

$$\mathbf{Avoid}_{\hat{z}} = \bigcup_{i=1}^{q} \text{proj}^{-1}(\mathbf{Avoid}_{d,i}) \tag{3.12}$$

We can now continue with the system decomposition method from Section 3.1. We will utilize the methods from Section 2.4.1 to compute the $\mathbf{Reach}_{T}^{\sharp}(\mathbf{Avoid}_{d,i})$ set for each subsystem. Our full dimension maximal BRS then becomes

$$\mathbf{Reach}_{T}^{\sharp}(\mathbf{Avoid}_{\hat{z}}) = \bigcup_{i=i}^{q} \text{proj}^{-1}(\mathbf{Reach}_{T}^{\sharp}((\mathbf{Avoid}_{d,i})) \tag{3.13}$$

We then use (2.17) to relate the full dimension maximal BRT to the full dimension maximal BRS.

$$\mathbf{Reach}_{[0,T]}^{\sharp}(\mathbf{Avoid}_{\hat{z}}) = \bigcup_{s\in[0,T]} \mathbf{Reach}_{s}^{\sharp}(\mathbf{Avoid}_{\hat{z}}) \tag{3.14}$$

Finally, we relate the full dimension maximal BRT to the invariance kernel of the complement of the full dimension avoid set.

$$\text{Inv}_{[0,T]}(\mathbf{Avoid}_{\hat{z}}^{c}) = (\mathbf{Reach}_{[0,T]}^{\sharp}(\mathbf{Avoid}_{\hat{z}}))^{c} \tag{3.15}$$

We now have a set in the full $4q$ dimensions which represents initial conditions for which, for all control, the chaser satellite is guaranteed not to hit any of the debris for

the entire time horizon. We were able to compute this set by performing q maximal reach calculations on the four dimension subsystems, which is computationally feasible using the methods from Section 2.4.1. We also note that, "as-is", the invariance kernel is in the $z_{r,i}$ coordinates, but can be translated to the z coordinates relative to the chief through the linear relationship $z = z_{r,i} + z_{d,i}$. We also recall that this set will not guarantee the satellite reaches a specified target at the final time.

3.3 Reach-Avoid Under-Approximation Through System Decomposition

In the previous section we were able to extend the system decomposition method to the multiple debris avoidance problem, but we did not require the chaser satellite to also reach a target. In this section we will highlight the issues that prevent extending the system decomposition methods to calculating the full Reach-Avoid set.

We begin by noting that the system decomposition method applies to a single target or avoid set in the full state dimension, which is written as either an intersection or union of the inverse projections of the sub system target or avoid sets. We will assume that our full dimension state is given in the same manner as in Section 3.2 with the addition of state z; this will account for the chaser reaching the target. The addition of state z also adds another subsystem with the corresponding $\mathbf{Target}_z$ set. We again note that to guarantee safety we must characterize a state in the full dimension for all time, implying that we need either an invariance or viability kernel.

Let $\mathcal{S}$ represent the full dimension set for which we will be computing an invariance or viability kernel. While the methods from [18] are not directly applied to the viability or invariance kernel, Section 3.2 shows how the methods can be utilized to generate these full dimension kernels through minimal or maximal BRTs. We would

like $\text{Viab}_{[0,T]}(\mathcal{S})$ or $\text{Inv}_{[0,T]}(\mathcal{S})$ to be an under-approximation of the Reach-Avoid set. We further note that, to follow the system decomposition method, the set $\mathcal{S}$ can be written in one of two ways

$$\begin{aligned} \mathcal{S}_1 &= \text{proj}{-}1(\mathbf{Target}_z) \bigcap_{i=1}^{q} \text{proj}^{-1}(\mathbf{Avoid}_{d,i}^c) \\ \mathcal{S}_2 &= \text{proj}{-}1(\mathbf{Target}_z) \bigcup_{i=1}^{q} \text{proj}^{-1}(\mathbf{Avoid}_{d,i}^c) \end{aligned} \tag{3.16}$$

We note that if $\mathcal{S}$ is written in the form of $\mathcal{S}_2$ we have the same issues from Section 3.2 where neither $\text{Viab}_{[0,T]}(\mathcal{S}_2)$ nor $\text{Inv}_{[0,T]}(\mathcal{S}_2)$ will guarantee that the chaser satellite remains outside of all debris for all time. For the $\text{Viab}_{[0,T]}(\mathcal{S}_2)$ we have

$$\begin{aligned} \text{Viab}_{[0,T]}(\mathcal{S}_2) = \{\hat{z} \in \hat{\mathcal{Z}} \mid \exists u \in \mathcal{U},\, \forall s \in [0,T],\, \hat{z}(s) \in \text{proj}^{-1}(\mathbf{Target}_z) \vee \\ \text{proj}^{-1}(\mathbf{Avoid}_{d,1}^c) \vee \ldots \vee \text{proj}^{-1}(\mathbf{Avoid}_{d,q}^c)\} \end{aligned} \tag{3.17}$$

Comparing (3.17) to the definition of the full Reach-Avoid in (2.6) we note that the $\text{Viab}_{[0,T]}(\mathcal{S}_2)$ only assures that the state will be in the target *or* outside a single piece of debris, not in the target *and* outside *all* debris. Furthermore, from (3.17) we see that the state must be in the target set for all time, as apposed to in the target set at the final time T. The $\text{Inv}_{[0,T]}(\mathcal{S}_2)$ is given by replacing '$\exists u \in \mathcal{U}$' with '$\forall u \in \mathcal{U}$' in (3.17). This change in control policy does not alleviate the issues from the viability kernel under-approximation and we are left with the same issues trying to under approximate the full Reach-Avoid set with the invariance kernel.

Figure 3.5 shows a representation of the set $\mathcal{S}_2$. The top and middle, images show the avoid complement sets and target sets in green. The bottom left image then shows the the union of these sets in purple. Clearly, guaranteeing that the satellite remain in this purple set will not guarantee that it remains outside all debris for all time.

If we were to write the set $\mathcal{S}$ in the form of $\mathcal{S}_1$ then the $\text{Viab}_{[0,T]}(\mathcal{S}_1)$ would become

$$\begin{aligned} \text{Viab}_{[0,T]}(\mathcal{S}_1) = \{\hat{z} \in \hat{\mathcal{Z}} \mid \exists u \in \mathcal{U},\, \forall s \in [0,T],\, \hat{z}(s) \in \text{proj}^{-1}(\mathbf{Target}_z) \wedge \\ \text{proj}^{-1}(\mathbf{Avoid}_{d,1}^c) \wedge \ldots \wedge \text{proj}^{-1}(\mathbf{Avoid}_{d,q}^c)\} \end{aligned} \tag{3.18}$$

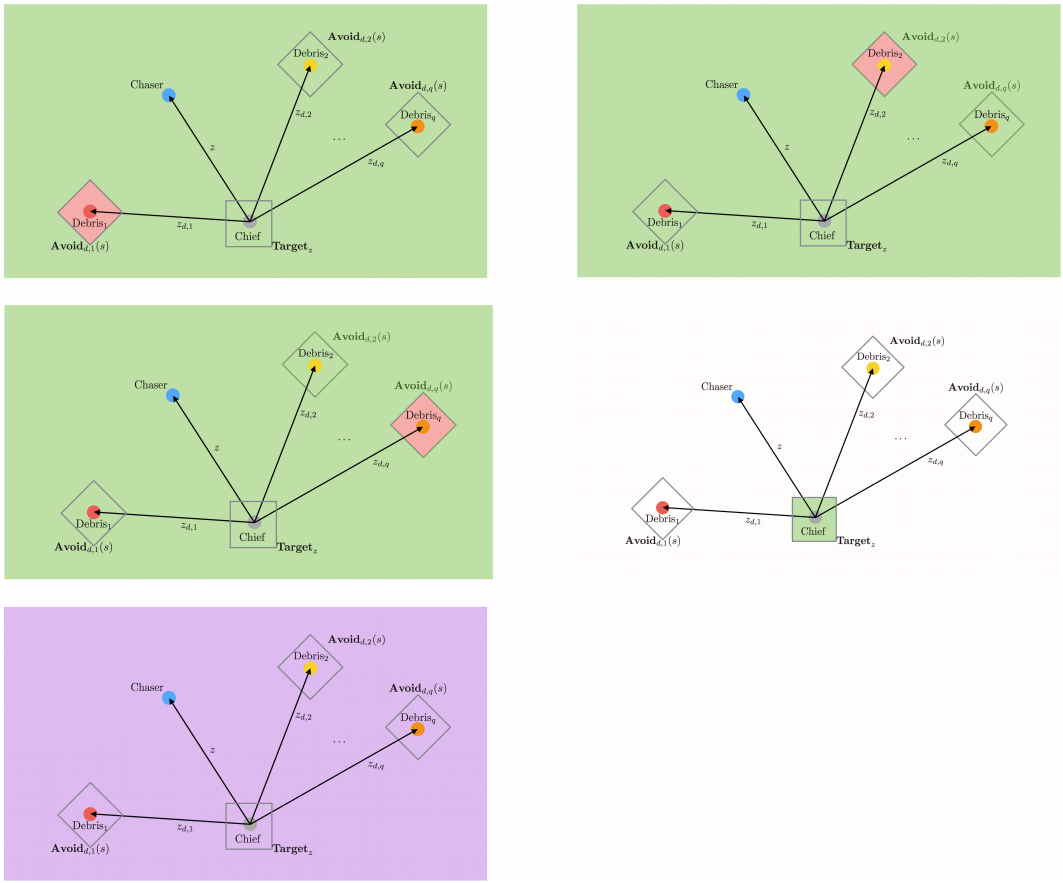

Figure 3.5: Representation of the S_2 set

While this would ensure that the satellite would stay outside all debris it also requires the satellite to remain inside the target set for all time. We can write $\text{Inv}_{[0,T]}(\mathcal{S}_1)$ by simply replacing the '$\exists u \in \mathcal{U}$' with '$\forall u \in \mathcal{U}$' in (3.18). Even with this change, we are still requiring our satellite to remain in the target set for all time. While this technically is a conservative solution for the full Reach-Avoid set, it is overly conservative and not useful in a practical sense. Figure 3.6 depicts an example of the set $\mathcal{S}_1$ in green illustrates why requiring the satellite to remain in this set for all time would be an overly conservative solution.

We see that there is no way to write a full dimension set $\mathcal{S}$ following the system

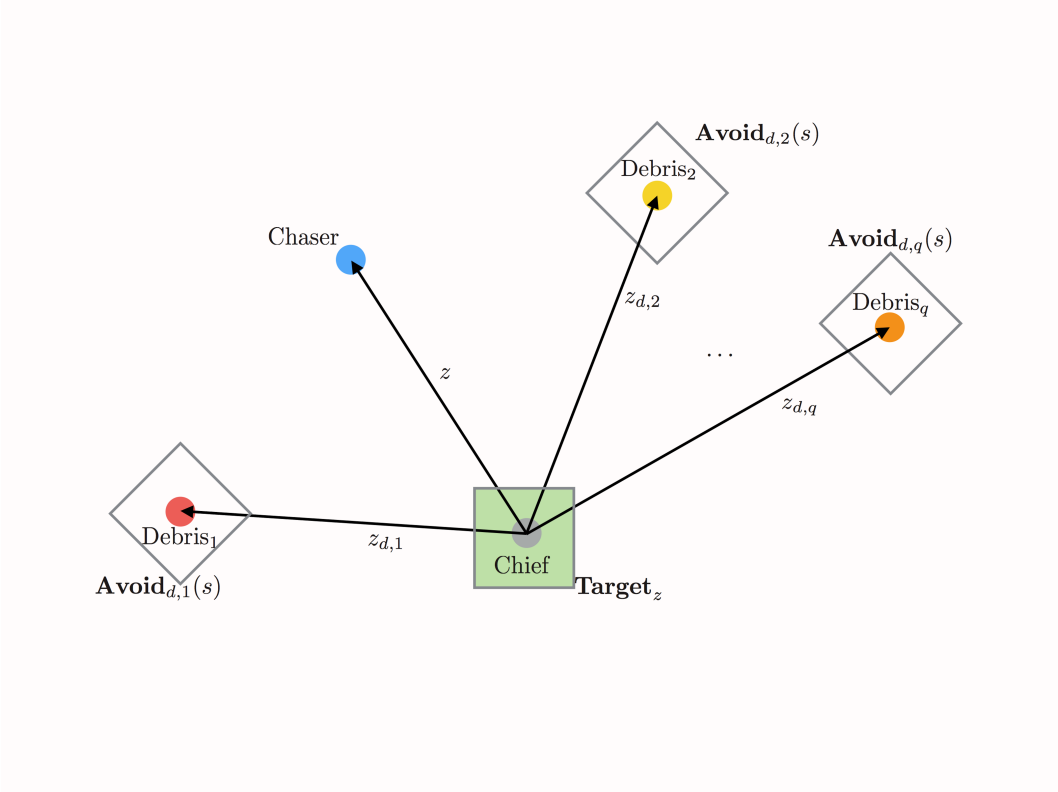

Figure 3.6: Representation of the S_1 set

decomposition framework such that the viability or invariance kernel of $\mathcal{S}$ will under approximate the Reach-Avoid in a useful manner. We either end up with a set which does not guarantee the chaser will be outside all debris for all time, or a set which requires the chaser to begin and stay in the target set.

Chapter 4

Under Approximating the Reach-Avoid Set

In this chapter we will develop a method for computing a conservative Reach-Avoid set using independent reach and avoid calculations. For our specific problem we will be looking at the chaser satellite attempting to reach the chief point and the chaser satellite attempting to avoid every piece of debris. We will develop two different methods which utilize the computationally tractable methods from Sections 2.4.1 and 2.4.2 to compute maximal and minimal BRSs. One method will use the maximal BRS for the target and the invariance kernel for all debris to under approximate the Reach-Avoid while the other method will use the minimal BRS for the target and the viability kernel. We note that in both of these combinations one set is a 'for all control' set while the other set is a 'there exists control' set. This is an important trait which will be leveraged to prove both of their under-approximations of the Reach-Avoid set.

4.1 Maximal Reach and Invariance Kernel

We begin with the maximal BRS and invariance kernel. We again have a target set $\mathbf{Target}_z$ and an arbitrary q number of time varying avoid sets $\mathbf{Avoid}_{d,i}(s)$ for a given time s. We also note that $\overline{\mathbf{Avoid}}^c_d(s) = \bigcap_{i=1}^q \mathbf{Avoid}^c_{d,i}(s)$. We begin by computing $\mathbf{Reach}^\sharp_T(\mathbf{Target}_z)$ using the methods in Section 2.4.1. This provides us with the set of states for which there exists a control such that the chaser satellite will end up in the $\mathbf{Target}_z$ set at final time T. We then begin the computation for the $\mathrm{Inv}_{[0,T]}(\overline{\mathbf{Avoid}}^c_{d,[0,T]})$ by computing $\mathbf{Reach}^\sharp_s(\mathbf{Avoid}_{d,i}(s))$. However, there are two main issues in the way of simply applying Section 2.4.1 to this calculation as is. We first note that we now have a time varying avoid set, where we previously assumed a constant avoid set. This can be fixed by allowing the final time polytope $\mathcal{P}^\sharp_j$ to be time varying. The final time polytope for a single facet of the avoid set then becomes

$$\mathcal{P}^\sharp_j(s) = \left\{ z \;\middle|\; \begin{array}{rcl} c_R^T z & \leq & d_R(s) \\ c_j^T A z & \leq & -c_j^T B u^{\sharp,*} \\ c_j^T z & = & d_j \end{array} \right\} \tag{4.1}$$

By letting $d_R(s)$ vary with time we are simply accounting for the position of the debris avoid set at time s.

The second issue we face with the $\mathbf{Reach}^\sharp_s(\mathbf{Avoid}_{d,i}(s))$ calculation is that the $\mathbf{Avoid}_{d,i}(s)$ sets are unbounded. We had initially required that the chaser satellite not enter the debris for any velocity value, however this leaves us with an unbounded set in the velocity dimensions. Because the methods we are trying to implement require the vertices of the final time polytope $\mathcal{P}^\sharp_j(s)$ we will have to bound the $\mathbf{Avoid}_{d,i}(s)$ sets. We note that if the chaser satellite enters the position of the debris at a larger velocity than our selected velocity bounds it will not violate the constraints, and will be thought of as safe. However, if we bound the avoid sets at too large of velocity values we will end up with an overly conservative answer which

will be of no practical use. We therefore would like to ensure that for any practical velocity value of the chaser satellite over the time horizon, the chaser satellite will not hit the debris. For our case, we will use the maximum and minimum velocity values form the $\mathbf{Reach}^{\sharp}_{T}(\mathbf{Target}_z)$ calculation. This says that the chaser satellite will not hit the debris for any velocity which will also take it to the target.

Finally, with our bounded $\mathbf{Avoid}_{d,i}(s)$ set and a final time polytope accounting for the position of the debris, we can compute the $\mathbf{Reach}^{\sharp}_{s}(\mathbf{Avoid}_{d,i}(s))$ set. We then follow our previous steps to generate the $\mathrm{Inv}_{[0,T]}(\mathbf{Avoid}^{c}_{d,[0,T],i})$ set. This will provide us with the set of states for which for all control will remain in the complement of the i^{th} piece of debris for all time. We then write $\mathrm{Inv}_{[0,T]}(\overline{\mathbf{Avoid}}^{c}_{d,[0,T]}) = \bigcap_{i=i}^{q} \mathrm{Inv}_{[0,T]}(\mathbf{Avoid}^{c}_{d,[0,T],i})$ knowing that for sets $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_m$, $\mathrm{Inv}_{[0,T]}(\bigcap_{i=1}^{m} \mathcal{K}_i) = \bigcap_{i=1}^{m} \mathrm{Inv}_{[0,T]}(\mathcal{K}_i)$. We now have a set of states for which for all control the chaser satellite will not hit any of the debris.

We then define our conservative Reach-Avoid set for the maximal BRS and invariance kernel as

$$\mathbf{RA}^{\sharp}(T, \mathbf{Target}_z, \overline{\mathbf{Avoid}}_{d,[0,T]}) = \mathbf{Reach}^{\sharp}_{T}(\mathbf{Target}_z) \bigcap \mathrm{Inv}_{[0,T]}(\overline{\mathbf{Avoid}}^{c}_{d,[0,T]}) \quad (4.2)$$

which represents the set of states for which there exists a control such that the chaser satellite will reach the target at final time T and, for all control, will not hit any debris for the entire time horizon.

Theorem 1.

$$\mathbf{RA}^{\sharp}(T, \mathbf{Target}_z, \overline{\mathbf{Avoid}}_{d,[0,T]}) \subseteq \mathbf{ReachAvoid}(T, \mathbf{Target}_z, \overline{\mathbf{Avoid}}_{d,[0,T]}) \quad (4.3)$$

Proof.

$$\begin{aligned}\mathbf{RA}^{\sharp}(T, \mathbf{Target}_z, \overline{\mathbf{Avoid}}_{d,[0,T]}) &= \mathbf{Reach}^{\sharp}_{T}(\mathbf{Target}_z) \bigcap \mathrm{Inv}_{[0,T]}(\overline{\mathbf{Avoid}}^{c}_{d,[0,T]}) \\ &= \{z \in \mathcal{Z} \mid \exists u,\ z(T) \in \mathbf{Target}_z\} \\ &\bigcap \{z \in \mathcal{Z} \mid \forall u,\ \forall s \in [0,T],\ z(s) \in \overline{\mathbf{Avoid}}^{c}_{d,[0,T]}\}\end{aligned} \tag{4.4}$$

If $\bar{z} \in \mathbf{RA}^{\sharp}(T, \mathbf{Target}_z, \overline{\mathbf{Avoid}}_{d,[0,T]})$, then by (4.4), there exists a control u that will drive the chaser to the target set $\mathbf{Target}_z$ at final time T, and all control policies should prevent the chaser from hitting all debris for all time. Hence, we have found a control policy that drives the chaser to the target while avoiding all debris implying $\bar{z} \in \mathbf{ReachAvoid}(T, \mathbf{Target}_z, \overline{\mathbf{Avoid}}_{d,[0,T]})$ by (2.6). □

4.2 Minimal Reach and Viability Kernel

Attempting to under approximate the Reach-Avoid set using the minimal BRS for the target and a viability kernel for all debris proves to be not possible when computing the viability kernel for each piece of debris independently. We run into similar issues we faced when trying to compute the viability kernel for multiple pieces of debris using the system decomposition method in Section 3.2. If we compute the viability kernel for each piece of debris independently we end up with a set of states for which there exists a control such that the satellite will not hit that specific piece of debris. There is no guarantee that it wont hit any other pieces of debris. Furthermore, intersecting the viability kernels from two different pieces of debris does not guarantee that the control policy which avoids one piece of debris is the same control policy which avoids the other piece of debris. In other words, for sets $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_m$, $\mathrm{Viab}_{[0,T]}(\bigcap_{i=1}^{m} \mathcal{K}_i) \neq \bigcap_{i=1}^{m} \mathrm{Viab}_{[0,T]}(\mathcal{K}_i)$. If a state $\bar{z} \in \bigcap_{i=1}^{m} \mathrm{Viab}_{[0,T]}(\mathcal{K}_i)$, then $\exists u_1,\ \forall s \in [0,T],\ \bar{z}(s) \in \mathcal{K}_1$, and $\exists u_2,\ \forall s \in [0,T],\ \bar{z}(s) \in$

$\mathcal{K}_2, \text{and} \ldots, \exists u_m,\ \forall s \in [0,T],\ \bar{z}(s) \in \mathcal{K}_m$. However, there is no guarantee that $u_1 = u_2 = \ldots = u_m$ which would be required to ensure that the satellite avoids all the debris.

If it were possible to compute the viability kernel for all debris at once, instead of separately, then it would be possible to intersect that viability kernel with the minimal reach to achieve an under-approximation of the Reach-Avoid. However, if each avoid set is a convex box, we can not say that their unions will also be convex; and thus, we can not apply the methods from Section 2.4.2 to find the viability kernel of the intersection of all debris avoid complement sets through the minimal BRSs of the union of the debris avoid sets.

In general, we can generate a second conservative Reach-Avoid set for multiple pieces of debris and a single target. We will use a generalized set $\overline{\mathbf{Avoid}}_{[0,T]}$ which represents the union of all the debris sets we wish to avoid and a general **Target** set we wish to reach. We will use the minimal BRS for the target and a viability kernel for the $\overline{\mathbf{Avoid}}^c_{[0,T]}$ set. We calculate the $\mathbf{Reach}^\flat_T(\mathbf{Target})$ set, the $\text{Viab}_{[0,T]}(\overline{\mathbf{Avoid}}^c_{[0,T]})$, and define our conservative Reach-Avoid set for the general case of multiple debris using the minimal BRS and viability kernel as

$$\mathbf{RA}^\flat(T, \mathbf{Target}, \overline{\mathbf{Avoid}}_{[0,T]}) = \mathbf{Reach}^\flat_T(\mathbf{Target}) \bigcap \text{Viab}_{[0,T]}(\overline{\mathbf{Avoid}}^c_{[0,T]}) \tag{4.5}$$

This set represents the set of states for which for all control the state will end up in the target at final time T and for which there exists a control such that the state can avoid all the debris for all time.

Theorem 2.

$$\mathbf{RA}^\flat(T, \mathbf{Target}, \overline{\mathbf{Avoid}}_{[0,T]}) \subseteq \mathbf{ReachAvoid}(T, \mathbf{Target}, \overline{\mathbf{Avoid}}_{[0,T]}) \tag{4.6}$$

Proof.

$$\begin{aligned}\mathbf{RA}^\flat(T, \mathbf{Target}, \overline{\mathbf{Avoid}}_{[0,T]}) &= \mathbf{Reach}^\flat_T(\mathbf{Target}) \bigcap \mathrm{Viab}_{[0,T]}(\overline{\mathbf{Avoid}}^c_{[0,T]}) \\ &= \{z \in \mathcal{Z} \mid \forall u,\ z(T) \in \mathbf{Target}\} \\ &\bigcap \{z \in \mathcal{Z} \mid \exists u,\ \forall s \in [0,T],\ z(s) \in \overline{\mathbf{Avoid}}^c_{d,[0,T]}\} \quad (4.7)\end{aligned}$$

If $\bar{z} \in \mathbf{RA}^\flat(T, \mathbf{Target}, \overline{\mathbf{Avoid}}_{[0,T]})$, then by (4.7), for all control policies the state will end up in the target set **Target** at final time T, and there exists a control policy which should prevent the state from hitting all debris for all time. Hence, we have found a control policy that drives the state to the target while avoiding all debris implying $\bar{z} \in \mathbf{ReachAvoid}(T, \mathbf{Target}, \overline{\mathbf{Avoid}}_{[0,T]})$ by (2.6). □

We again note that because $\mathrm{Viab}_{[0,T]}(\bigcap_{i=1}^m \mathcal{K}_i) \neq \bigcap_{i=1}^m \mathrm{Viab}_{[0,T]}(\mathcal{K}_i)$, the conservative Reach-Avoid $\mathbf{RA}^\flat$ does not apply when taking the viability kernel of each debris independently. However, for a single debris avoidance problem the set $\overline{\mathbf{Avoid}}_{[0,T]}$ simply becomes $\mathbf{Avoid}_d(s)$. We can then use the methods from Section 2.4.2 to compute the viability kernel and minimal BRS and generate the under-approximation of the Reach-Avoid set as $\mathbf{RA}^\flat(T, \mathbf{Target}_z, \mathbf{Avoid}_{d,[0,T]})$, for a single piece of debris.

Chapter 5

Examples

We begin our examples by defining the constants for the CWH equations. These will apply for the dynamics of both the chaser vehicle and the debris for all the following examples. We have constants, $R = 850 + 6378.1$ km, $G = 6.673 \times 10^{-11}$ $\frac{\mathrm{m}^3}{\mathrm{kg}\cdot\mathrm{s}^2}$, $M = 5.9472 \times 10^{24}$ kg, and $\mu = \frac{G\cdot M}{1000^3}$ $\frac{\mathrm{km}^3}{\mathrm{S}^2}$, such that $\omega = \sqrt{\frac{\mu}{R^3}}$ $\frac{1}{\mathrm{S}}$, $m_c = 150$ kg, and $u_{\max} = 10^{-3}$ $\frac{\mathrm{kg}\cdot\mathrm{m}}{\mathrm{S}^2}$. Where R is the radius of the orbit, G is the universal gravitational constant, M is the mass of the earth, μ is the standard gravitational parameter, and m_c is the mass of the chaser vehicle.

We then define the constraints for the $\mathbf{Target}_z$ set as

$$c_{R,z}^T = \begin{bmatrix} I_{4\times 4} \\ -I_{4x4} \end{bmatrix}, \; d_{R,z} = \begin{bmatrix} x_{\max} \\ -x_{\min} \end{bmatrix} \tag{5.1}$$

with

$$x_{\max} = \begin{bmatrix} 0.1 \\ 0.001 \\ 0.1 \\ 0.001 \end{bmatrix}, \; x_{\min} = \begin{bmatrix} -0.1 \\ -0.001 \\ -0.1 \\ -0.001 \end{bmatrix} \tag{5.2}$$

where $x_{\max}$ and $x_{\min}$ are in km for the position states and $\frac{\text{km}}{\text{s}}$ for the velocity constraints. The constraints for the $\mathbf{Avoid}_{d,i}(s)$ sets are

$$c_{R,d}^T = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \; d_{R,d}(s) = \begin{bmatrix} -z_{d,i,,x}(s) + z_{d,i,y}(s) + \delta \\ z_{d,i,x}(s) - z_{d,i,y}(s) + \delta \\ z_{d,i,x}(s) + z_{d,i,y}(s) + \delta \\ -z_{d,i,x}(s) - z_{d,i,y}(s) + \delta \\ \dot{x}_{\max} \\ -\dot{x}_{\min} \\ \dot{y}_{\max} \\ -\dot{y}_{\min} \end{bmatrix} \tag{5.3}$$

Where $z_{d,i,x}(s)$ and $z_{d,i,y}(s)$ are the x and y positions of the i^{th} piece of debris at time s and $\dot{x}_{\max}$, $\dot{x}_{\min}$, $\dot{y}_{\max}$, $\dot{y}_{\min}$ are the maximum and minimum velocities of the chaser obtained from the maximal reach calculation of the target.

5.1 Maximal Reach and Invariance Kernel for Single Debris

We begin the under-approximation $\mathbf{RA}^\sharp$ for a single debris by assuming $\overline{\mathbf{Avoid}}_{d,[0,T]}$ = $\mathbf{Avoid}_{d,[0,T]}$. We then begin our calculation of the $\mathbf{RA}^\sharp(T, \mathbf{Target}_z, \mathbf{Avoid}_{d,[0,T]})$ set with the calculation of the $\mathbf{Reach}_T^\sharp(\mathbf{Target}_z)$ set. We construct the $\mathcal{P}_j^\sharp$ polytope for the $\mathbf{Target}_z$ set and final time T according to the constraints given in (2.44), and then evolve each vertex of the polytope backwards in time according to (2.1) with optimal control given by (2.43) and costate dynamics given by $\dot{p} = -A^T p$.

Because the full system is in four dimension visualization of the reach sets can be difficult. To try and make it simpler we will plot projections of the full four dimension sets onto the x and y position plane. This will give us a set of position

values which have at least one set of corresponding velocity values such that they will be in the full four dimension reach set. Figure 5.1 shows a plot of the projection of the $\mathbf{Reach}_T^\sharp(\mathbf{Target}_z)$ for the four position facets for a time horizon of $T = 50$s.

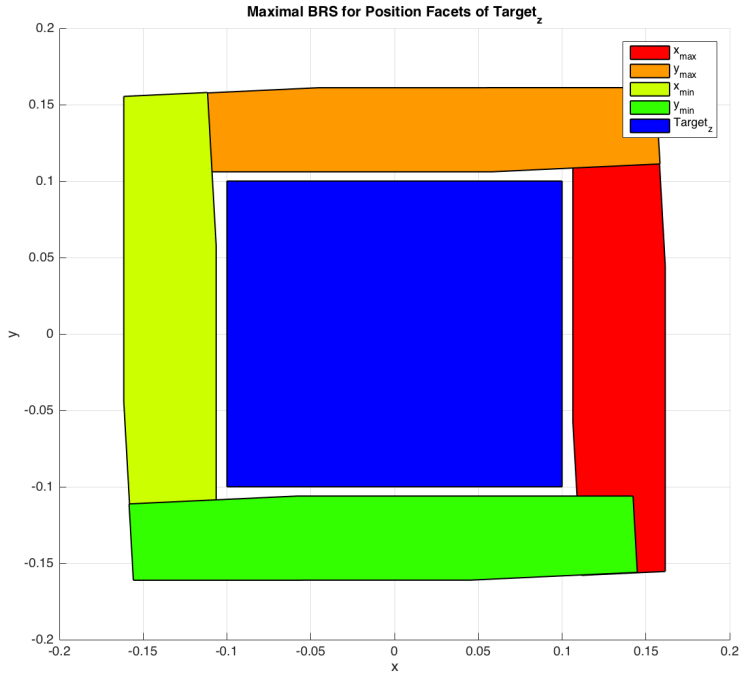

Figure 5.1: Maximal BRS of the position facets of the Target set, $\mathbf{Reach}_T^\sharp(\mathbf{Target}_z)$, for a 50 second time horizon

We note that the methods being used are calculating the maximal BRS for a single facet. In other words, the maximal BRS for the $x_{\max}$ facet are the states for which there exists a control such that they will end up exactly on the $x_{\max}$ facet of the $\mathbf{Target}_z$ set. A state is on the $x_{\max}$ position facet of the $\mathbf{Target}_z$ set if the other states, $\dot{x}, y, \dot{y}$ are within their respective bounds given in (5.1) and (5.2) and the state $x = x_{\max}$. Similarly for the other position facets. Hence, as mentioned in Section 2.4, the union of plots for every facet represents the boundary of the full

maximal BRS set.

We will now compute the maximal BRT for the $\mathbf{Avoid}_{d,[0,T]}$ set by computing the maximal BRS for the $\mathbf{Avoid}_d(s)$ for all time s. However, because these are numerical examples there will be a discretization of the maximal BRT. For our example the time step is 0.25s. This can be decreased to increase the fidelity of the discretization, or increased to decrease the computational time. We give our debris an initial state of $z_d(0) = [0,\ 0.03,\ 0.275,\ 0]^T$ and let the state autonomously evolve according to (2.3). The trajectory of the debris over all time can be seen in Figure 5.2.

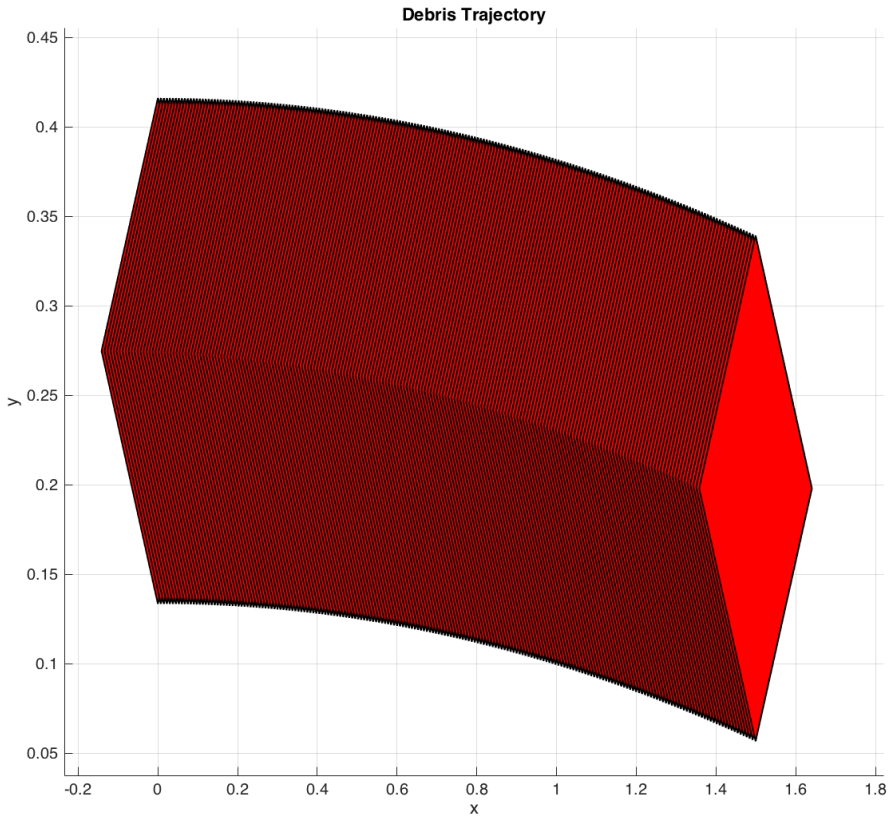

Figure 5.2: Trajectory of the avoid set of the debris over the time horizon T = 50s

We then generate the final time polytope $\mathcal{P}^{\sharp}_{j,d}(s)$ for each facet and every time s according to (2.44). We then evolve each vertex of $\mathcal{P}^{\sharp}_{j,d}(s)$ backwards in time for the corresponding time horizon according to the same state dynamics, optimal control, and costate dynamics as were used for the maximal BRS of the $\mathbf{Target}_z$ set. Figure 5.3 shows the projection of the maximal BRT for the $\mathbf{Avoid}_{d,[0,T]}$ set. It represents the set of states from which there exists a control such that the satellite will hit the debris at some time within the time horizon T. We also recall that everything outside of this set represents the set of states for which for all control the satellite will not hit the debris for all time in the time horizon.

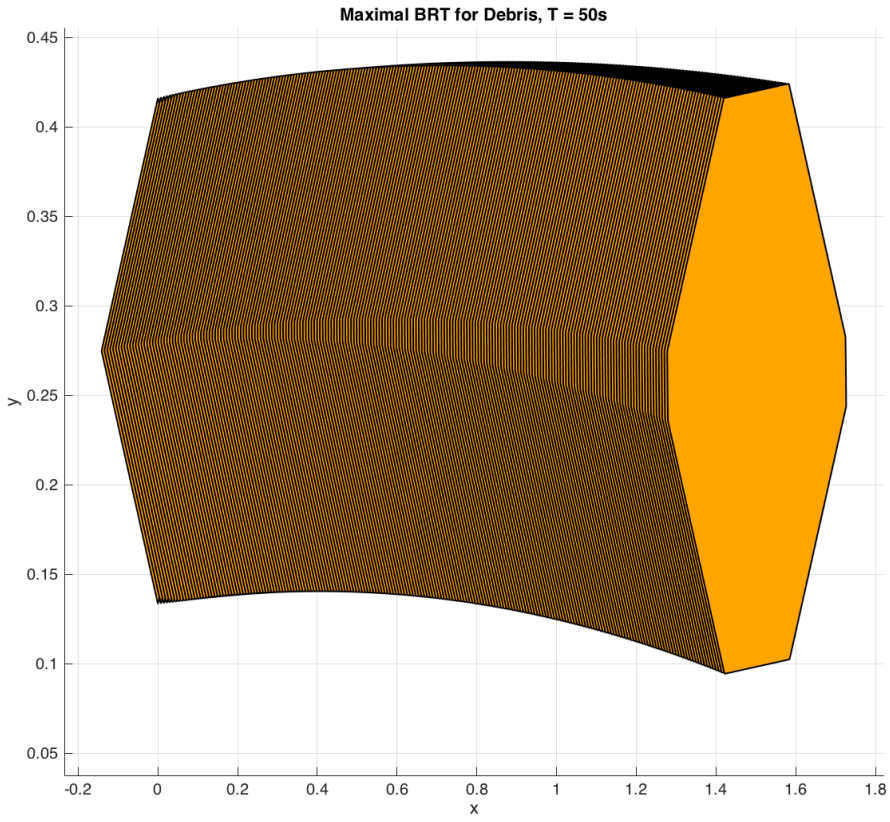

Figure 5.3: Maximal BRT of the avoid set, $\mathbf{Reach}^{\sharp}_{[0,T]}(\mathbf{Avoid}_{d,[0,T]})$, for T = 50s

Finally, Figure 5.4 shows the projection of $\mathbf{RA}^\sharp$ set for the position facets onto the x and y plane in green, the projection of the maximal BRT in orange, the trajectory of the debris in red, and the projection of the $\mathbf{Target}_z$ set in blue. We interpret the position values given in the plot of the $\mathbf{RA}^\sharp$ set (green) as the set of position states which will have corresponding velocity values such that there exists a control which will cause them to end up on the position facets of the $\mathbf{Target}_z$ set at final time T, and for all control the states will not enter the debris for all time over the time horizon.

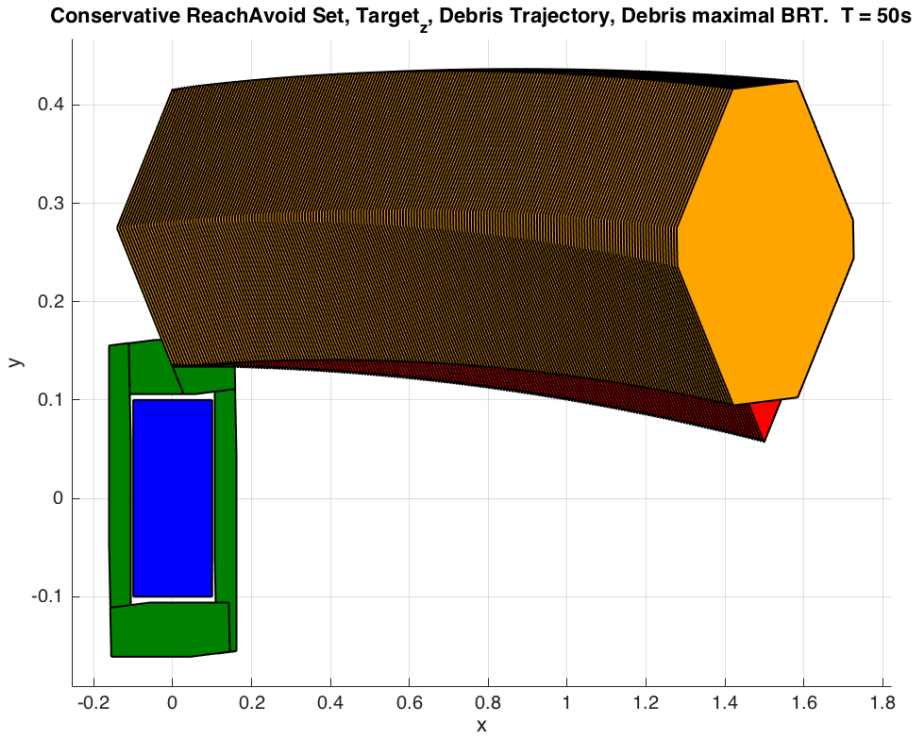

Figure 5.4: $\mathbf{RA}^\sharp$ under-approximation set (green) for single debris, target set (blue), avoid set trajectory (red), maximal BRT of avoid set, $\mathbf{Reach}^\sharp_{[0,T]}(\mathbf{Avoid}_{d,[0,T]})$, (orange), all for time horizon T = 50s

Figure 5.5 shows just the projection of the $\mathbf{RA}^\sharp$ set at the $\mathbf{Target}_z$ set. Compar-

ing this to the plot of the projection of the $\mathbf{Reach}_T^\sharp(\mathbf{Target}_z)$ set in figure 5.1 it is easy to see the removal of states in the maximal BRT of the debris. The discretization of the maximal BRT also becomes more apparent in this plot as you can see the gaps between each of the maximal BRS for the debris represented as triangles of the $\mathbf{RA}^\sharp$ set.

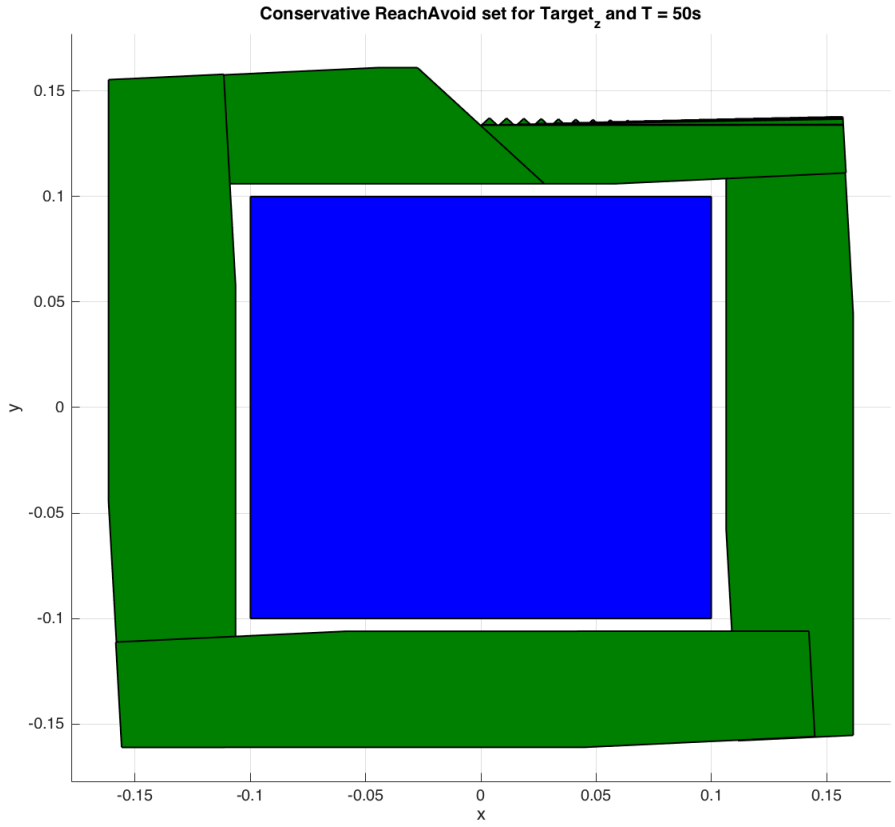

Figure 5.5: $\mathbf{RA}^\sharp$ under-approximation set (green) for single debris, target set (blue), for time horizon T = 50s

5.2 Maximal Reach and Invariance Kernel for Multiple Debris

The following is an example of the computation of the $\mathbf{RA}^\sharp$ set for three pieces of debris. The computation is similar to that for the single debris case and begins with the calculation of the $\mathbf{Reach}_T^\sharp(\mathbf{Target}_z)$ set using the methods from Section 2.4.1. For this calculation, however, we will be using a time horizon of $T = 100$s and a time step of 1s. Figure 5.6 shows the projection of the $\mathbf{Reach}_T^\sharp(\mathbf{Target}_z)$ set onto the x, y plane for the maximum and minimum position facets.

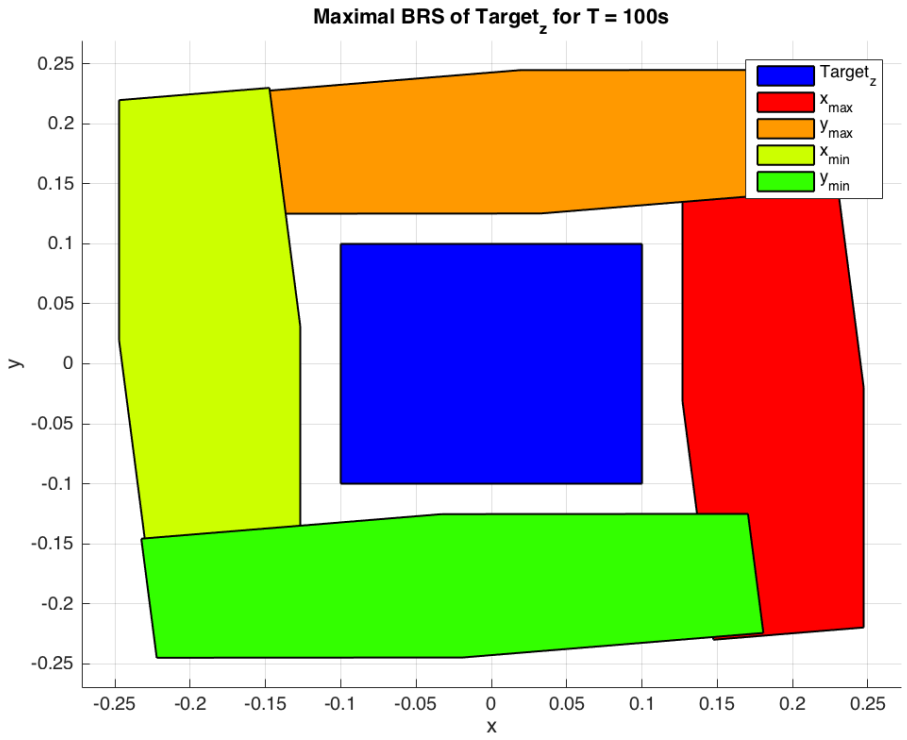

Figure 5.6: Maximal BRS, $\mathbf{Reach}_T^\sharp(\mathbf{Target}_z)$, for the maximum and minimum position facets of the Target for T = 100s

We note that compared to the maximal BRS for the $T = 50$s time horizon shown in Figure 5.1 this maximal BRS for the $T = 100$s time horizon is larger and further away from the $\mathbf{Target}_z$ set.

We will now compute the autonomous evolution of each piece of debris using the dynamics given in (2.3), along with their corresponding maximal BRTs using the methods from Section 2.4.1. The initial conditions for the debris are $z_{d,1}(0) = [-1, 0.03, 0.35, 0]^T$, $z_{d,2}(0) = [-1, 0.02, -1.5, 0.02]^T$, and $z_{d,3}(0) = [0.75, -0.01, -0.7, 0]^T$. Figure 5.7 shows the position of each piece of debris in red and the projection of the maximal BRTs onto the x, y plane for each piece of debris.

Figure 5.7: Multiple avoid set trajectories (red) and corresponding maximal BRTs, $\mathbf{Reach}^{\sharp}_{[0,T]}(\mathbf{Avoid}_{d,i,[0,T]})$, (orange) for T = 100s

The maximal BRTs for each piece of debris are larger than those for the single debris case because of the longer time horizon. Similar to the $\mathbf{Reach}^{\sharp}_{T}(\mathbf{Target}_z)$ set calculation, the longer time horizon leads to larger BRSs for the debris. Furthermore, the maximum and minimum velocity values form the $\mathbf{Reach}^{\sharp}_{T}(\mathbf{Target}_z)$ calculation will have increased in magnitude due to the longer time horizon. This increases the bounds on the initial $\mathbf{Avoid}_{d,i}(s)$ sets which also leads to larger BRTs for the debris.

We then generate the $\mathbf{RA}^{\sharp}$ set by intersecting the maximal BRS for the target with the complement of the maximal BRTs for each piece of debris. Figure 5.8 shows the projection of the $\mathbf{RA}^{\sharp}$ onto the x, y plane in green along with the projection of the $\mathbf{Target}_z$ set in blue and the maximal BRTs for each piece of debris in orange and the trajectories of each piece of debris in red.

Finally, Figure 5.9 shows just the projection of the $\mathbf{RA}^{\sharp}$ set for the multiple debris example onto the x, y plane in green, along with the $\mathbf{Target}_z$ set in blue.

5.3 Minimal Reach and Viability Kernel for Single Debris

This last examples aims to highlight the main differences between the $\mathbf{RA}^{\sharp}$ under-approximation and the $\mathbf{RA}^{\flat}$ under-approximation. As mentioned in the previous chapter we can only generate the $\mathbf{RA}^{\flat}$ set for the single debris case. This following example will use the same initial condition for the debris as was used in Section 5.1, $z_d(0) = [0,\ 0.03,\ 0.275,\ 0]^T$, and the same time horizon of $T = 50$s and time step of 0.25s. The trajectory for the debris is the same as in Section 5.1 and can be seen in Figure 5.2. We then compute the minimal BRS of the target, $\mathbf{Reach}^{\flat}_{T}(\mathbf{Target}_z)$, using the methods in Section 2.4.2. Figure 5.10 shows the projection of the minimal BRS of the target for the maximum and minimum position facets onto the x, y plane.

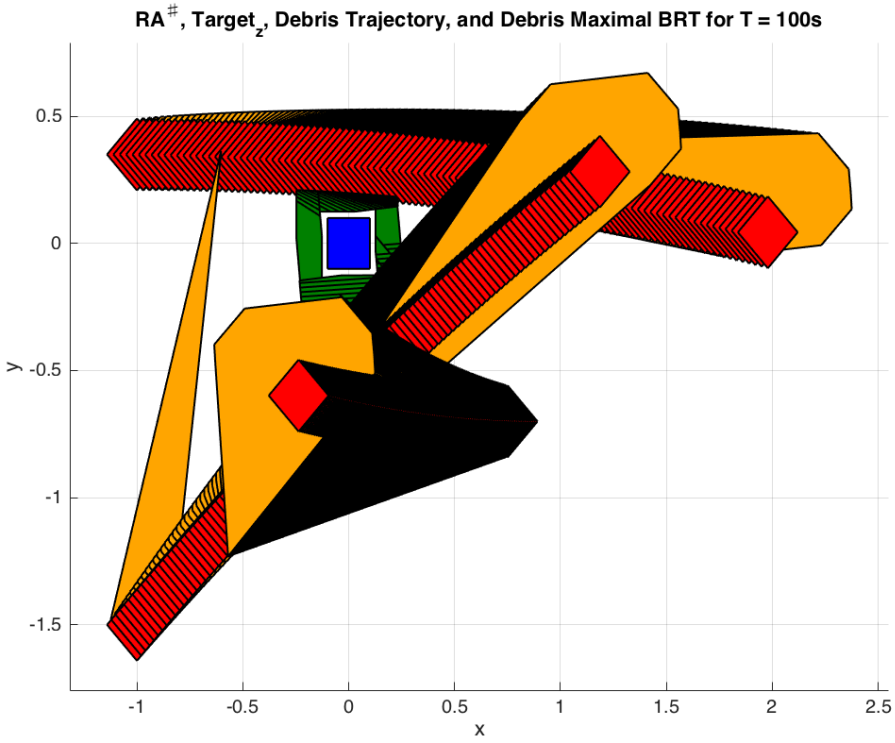

Figure 5.8: Projections of the $\mathbf{RA}^\sharp$ under-approximation set (green), Target (blue), and maximal BRTs, $\mathbf{Reach}^\sharp_{[0,T]}(\mathbf{Avoid}_{d,i,[0,T]})$, of avoid sets (orange), and trajectories of avoid sets (red) for multiple debris example and time horizon T = 100s

We note that, as expected, this set is much smaller than that given in Figure 5.1 for the maximal BRS of the target. Next, we can generate the minimal BRT for the debris by computing the $\mathbf{Reach}^\flat_s(\mathbf{Avoid}_d(s))$, $\forall s \in [0, T]$. Figure 5.11 shows the projection of the minimal BRT for the debris onto the x, y plane.

As with the minimal BRS of the target, the minimal BRT of the debris is also smaller than that of the maximal BRT for the debris given in Figure 5.3. Figures 5.12 shows the projections of the $\mathbf{Target}_z$ set (blue), the minimal BRT of the debris

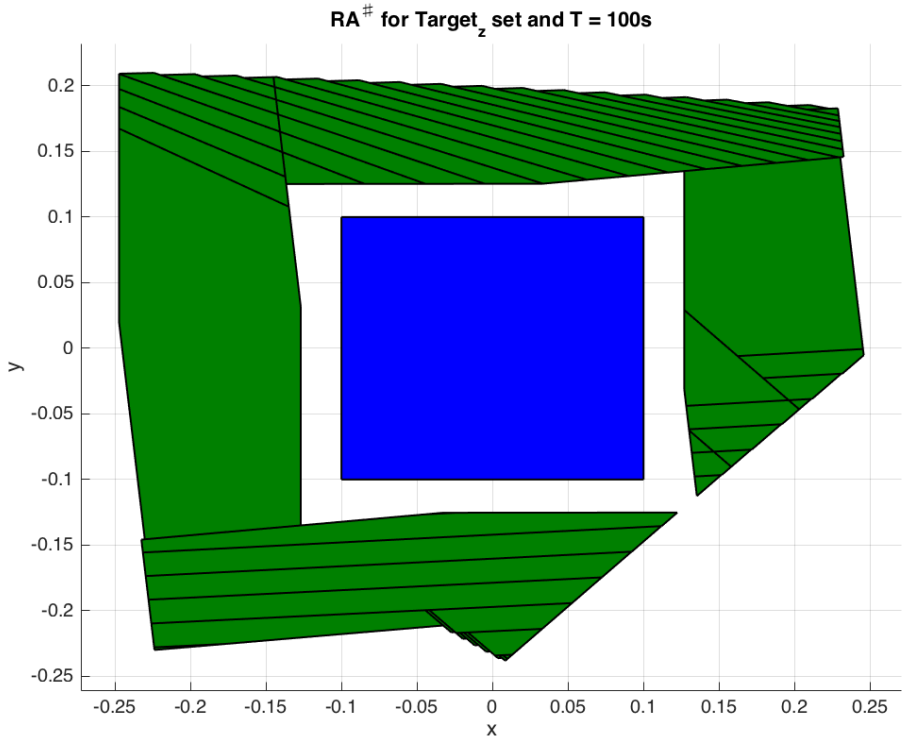

Figure 5.9: Projections of the $\mathbf{RA}^{\sharp}$ under-approximation set (green) and Target (blue) for multiple debris example and time horizon T = 100s

(orange), and the $\mathbf{RA}^{\flat}$ set (green) onto the x, y plane along with the trajectory of the debris (red); while 5.13 shows just the projection of the $\mathbf{RA}^{\flat}$ set (green) and the $\mathbf{Target}_z$ set (blue) onto the x, y plane for a closer view.

5.4 Discussion of Examples

Tables 5.1, 5.2, and 5.3 show the computational times for computing the individual BRS of the target, the BRTs of the avoid set or each avoid set in the multiple debris case, and the computational time associated with generating the under-

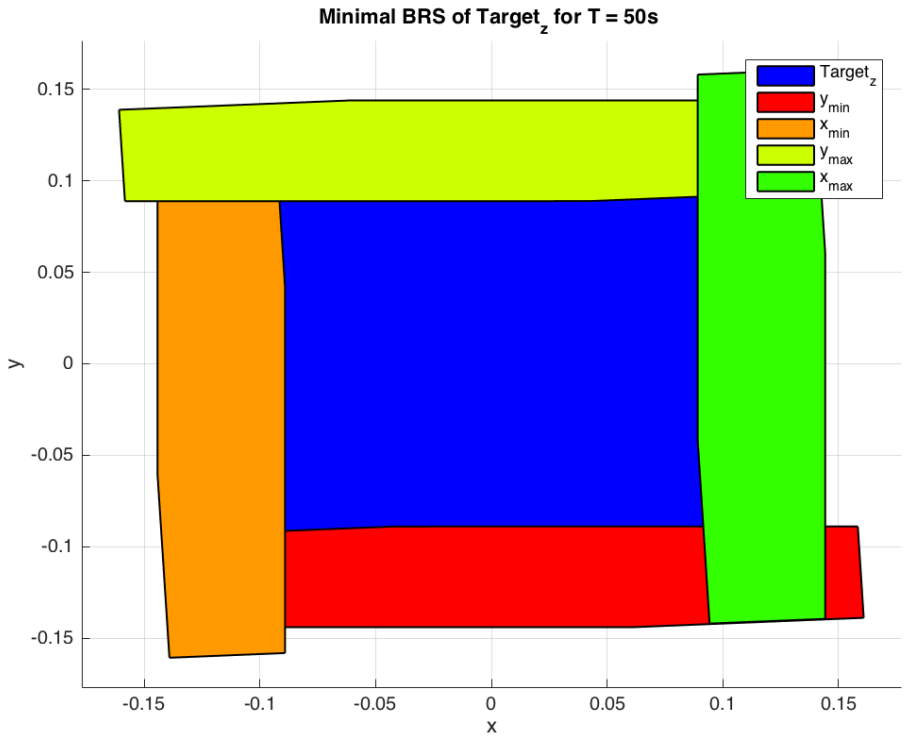

Figure 5.10: Projection of minimal BRS of the target, $\mathbf{Reach}_T^{\flat}(\mathbf{Target}_z)$, for the maximum and minimum position facets for T = 50s

approximation of the full Reach-Avoid set. Table 5.1 shows the computational times for the $\mathbf{RA}^{\sharp}$ single debris example given in Section 5.1, Table 5.2 shows the computational times for the $\mathbf{RA}^{\sharp}$ multiple debris example given in Section 5.2, and Table 5.3 shows the computational times for the $\mathbf{RA}^{\flat}$ single debris example given in Section 5.3. Each BRS and BRT calculation is a four dimension calculation. The examples were run in MATLAB 2015b on a computer with a 2.5 GHz quad-core Intel Core i7 CPU and 16 GB of RAM.

The total computational times for generating either the $\mathbf{RA}^{\sharp}$ or the $\mathbf{RA}^{\flat}$ under approximation, for single or multiple debris, are not fast. However, as previously

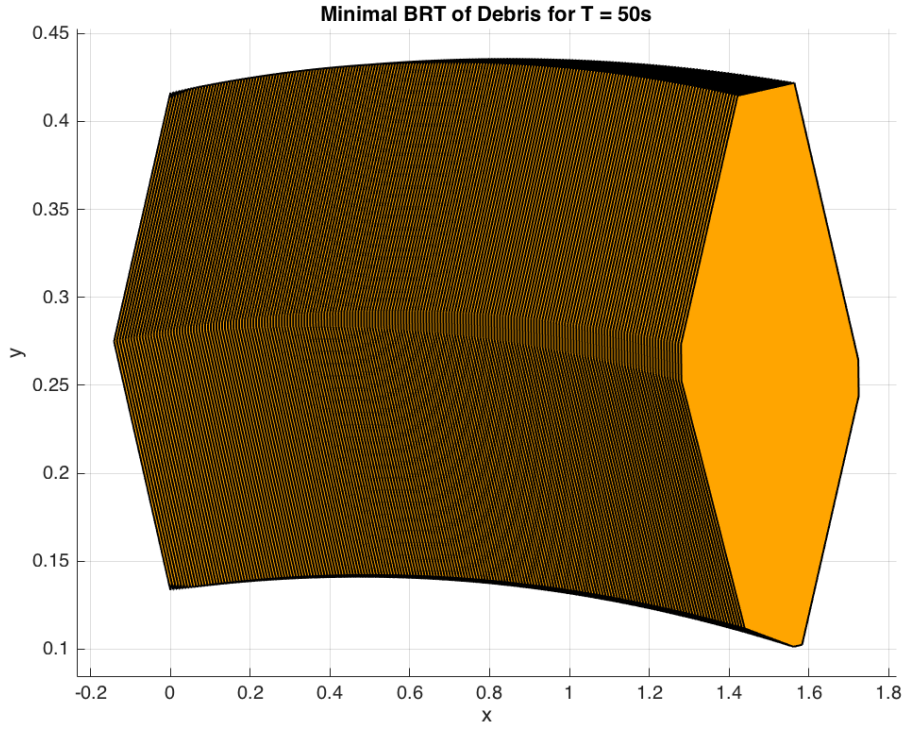

Figure 5.11: Projection of minimal BRT of the avoid set, $\mathbf{Reach}^{\flat}_{[0,T]}(\mathbf{Avoid}_{d,[0,T]})$, for T = 50s

mentioned, numerical solutions using level-set methods in even a four dimension system are not possible, so while these computational times are not quick, they are

Table 5.1: Computational times of maximal reach and invariance kernel under-approximation for single debris example

	Maximal BRS of $\mathbf{Target}_z$	Maximal BRT of z_d	Under-Approximation
Computational Time (s)	0.111	258	2.50

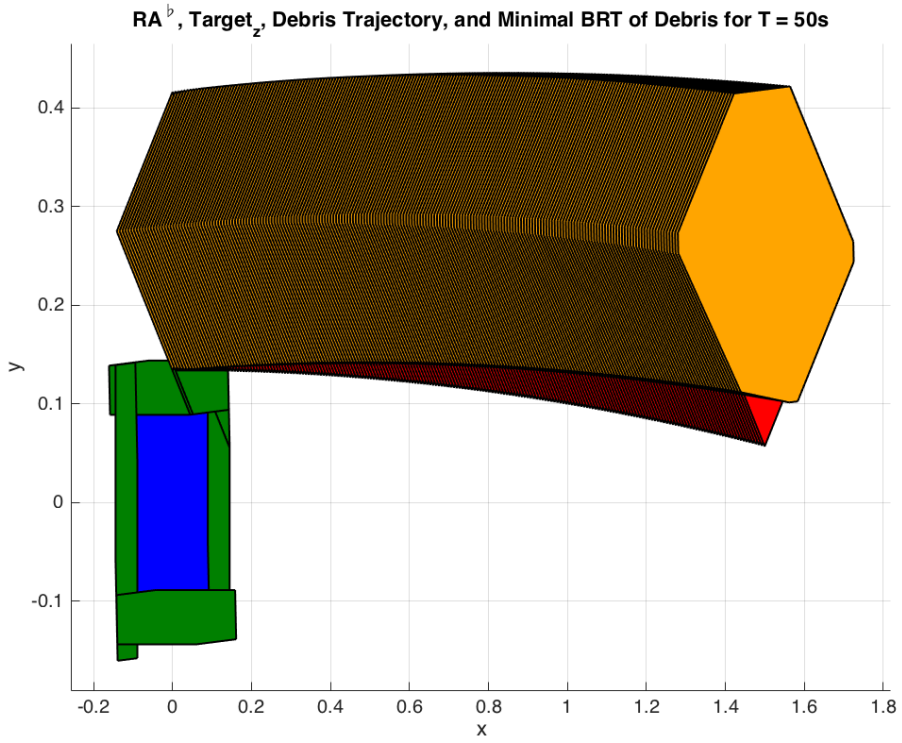

Figure 5.12: Projection of the $\mathbf{RA}^\flat$ under-approximation set (green), target set (blue), and minimal BRT of avoid set, $\mathbf{Reach}^\flat_{[0,T]}(\mathbf{Avoid}_{d,[0,T]})$, (orange), along with the avoid set trajectory (red) for T = 50s

Table 5.2: Computational times of maximal reach and invariance kernel under-approximation for multiple debris example

	Maximal BRS of $\mathbf{Target}_z$	Maximal BRT of $z_{d,1}$	Maximal BRT of $z_{d,2}$	Maximal BRT of $z_{d,3}$	Under-Approximation
Computational Time (s)	0.119	128	129	129	12

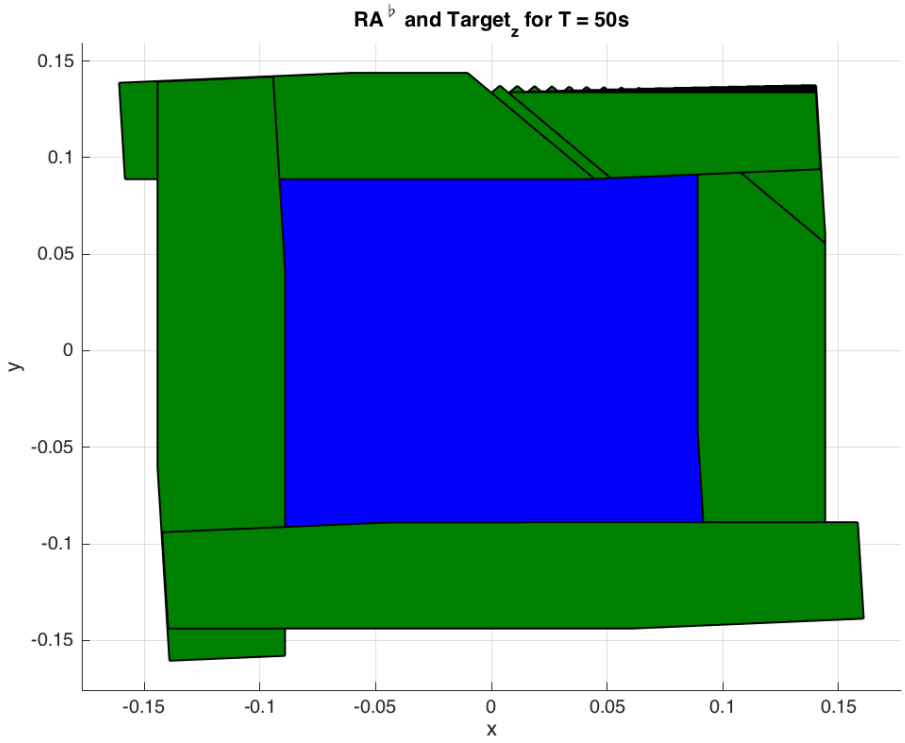

Figure 5.13: Projection of the $\mathbf{RA}^\flat$ under-approximation set (green) and target set (blue) for T = 50s

an improvement. Furthermore, these computational times may be reduced more through a different implementation of the code.

Table 5.3: Computational times of minimal reach and viability kernel under-approximation for single debris example

	Maximal BRS of $\mathbf{Target}_z$	Maximal BRT of z_d	Under-Approximation
Computational Time (s)	0.111	133	2.16

Currently, the code generates an avoid set for each time step in the time horizon. In the single debris examples this corresponds to a total of 200 avoid sets and in the multiple debris case there are 100 avoid sets for each piece of debris for a total of 300 avoid sets. For the $\mathbf{RA}^\sharp$ under-approximation all eight facets will evolve backwards in time where as only four of the facets will for the $\mathbf{RA}^\flat$ under-approximation. The code then generates a BRS for each facet of each of these avoid sets for an increasingly longer time horizon. The time horizon for the avoid set at $s = 20$s is 20s, the time horizon for the avoid set at $s = 30$s is 30s, and so on. If we instead assumed that the debris was *not* moving we would only need to evolve one avoid set over the entire time horizon. Because the dynamics of the system are linear we could then transform the BRS of the avoid set to the corresponding position of the debris at each time step. This would dramatically reduce the computational time of the code. While this more efficient algorithm was developed, there were discrepancies that arose between the two algorithms that could not be addressed in time. Furthermore, the large time difference between the $\mathbf{RA}^\sharp$ under-approximation and the $\mathbf{RA}^\flat$ under-approximation for the single debris case is due to the fact that only four facets in the $\mathbf{RA}^\flat$ calculation evolve backwards in time, opposed to the the $\mathbf{RA}^\sharp$ calculation where all eight facets evolve, leading to a shorter computational time for the $\mathbf{RA}^\flat$ set.

In Figures 5.7 and 5.8 we note an irregularity in one of the maximal BRS of the maximal BRT of the avoid set. We believe this is due to a numerical issue related to the projection of the full four dimension set onto the two dimension position plane. Unfortunately, we did not have the time to fully investigate what was causing this numerical discrepancy or why it only appears to affect just one of the maximal BRSs.

Chapter 6

Conclusion

In this thesis the safe maneuvering of a satellite in the presence of debris was investigated. Reachability techniques were used to guarantee the safety of the satellite with respect to the debris while also guaranteeing that the satellite would be able to reach some new state in its orbit. We assumed that the satellite's and debris' positions and velocities relative to another point near the same orbit were modeled using the LTI CWH equations. We also assumed that the debris had no external input and underwent only autonomous evolution.

First, a computationally tractable solution for the Reach-Avoid set was generated for a system under the CWH dynamics. This method was then further extended to the single maximal or minimal reach problem. Attempting to apply this method directly to the debris avoid problem proved problematic as the complement of the avoid sets were no longer convex. We then investigated the extension of system decomposition reachability techniques to the Reach-Avoid problem. While we found they could not be directly applied to the full Reach-Avoid problem we were able to extend them to the multiple debris avoidance problem. While this would guarantee the safety of the satellite it would not guarantee that we reach any target position.

Finally, we developed two under-approximations for the Reach-Avoid set for the spacecraft rendezvous problem in the presence of debris. This method allowed us to compute the reach and avoid sets independently and construct a conservative Reach-Avoid using a combination of those reach and avoid sets. One under-approximation used the maximal BRS and the invariance kernel for which we could apply our previous methods for computing the reach set of a system with CWH dynamics for multiple pieces of debris. The second method used the minimal BRS and the viability kernel which could not be generated using our reach techniques unless we only considered a single debris case.

References

[1] N. Dennehy and R. Carpenter, "Demonstration of autonomous rendezvous technology mishap investigation board review," Tech. Rep. NASA Engineering and Safety Center Technical Report RP-06-118, January 2007.

[2] I.M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science 4416, A. Bemporad, A. Bicchi, and G. Butazzo, Eds. Berlin, Heidelberg: Springer-Verlag, April 2007, pp. 428-443.

[3] K. Margellos and J. Lygeros, "Hamilton-jacobi formulaiton for reach-avoid differential games," *Transaction on Automatic Control*, vol. 56, no. 8, pp. 1849-1861, August 2011.

[4] C. J. Tomlin, J. Lygeros, and S. S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceeding of the IEEE*, vol. 88, no. 7, pp. 949-970, 2000.

[5] J. Lygeros, "On reachability and minimum cost optimal control," *Automatica*, vol. 40, no. 6, pp. 917-927, June 2004.

[6] I. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947-957, July 2005.

[7] C. Le Guernic and A. Girard, "Reachability analysis of linear systems using support functions," *on linear analysis: Hybrid Systems*, vol. 4, no. 20, pp. 250-262, 2010.

[8] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal toolbox," in *Proceeding of the IEEE Conference on Decision and Control*, San Diego, CA, December 2006, pp. 1498-1503.

[9] J. N. Maidens, S. Kaynama, I. M. Mitchell, M. M. Oishi, and G. A. Dumont, "Lagrangian methods for approximation the viability kernel in high-dimensional systems," *Automatica*, vol. 49, no. 7, pp. 2017-2029, 2013.

[10] M. Kvasnica, P. Grieder, M. Boatic, and M. Morari, "Multi-paramteric toolbox (MPT)," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science 2993, R. Alur and G. Pappas, Eds., April 2004, pp. 448-462.

[11] J. Tschauer, "Elliptic orbit rendezvous," *AIAA Journal*, vol. 5, no. 6, pp. 1110-1113, 1967.

[12] H. Park, S. D. Cairano, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking with a rotating/tumbling platform and for debris avoidance," in *Proceedings of the American Control Conference*, San Francisco, CA, June 2011, pp. 1922-1927.

[13] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies," *IEEE Transaction on Control Systems Technology*, vol. 23, no. 4, pp. 1638-1647, July 2015.

[14] K. Margellos and J. Lygeros, "Viable set computation for hybrid systems," *Nonlinear Analysis: Hybrid Systems*, vol. 10, pp. 45-62, 2013.

[15] C. Tomlin, I. Mitchell, A. Bayen, and M. Oishi, "Computational techniques for the verification of hybrid systems," *Proceeding of the IEEE*, vol. 91, no. 7, pp. 986-1001, 2003.

[16] C. Tomlin, "Hybrid Control of air traffic management systems," Ph.D. dissertation, Berkeley, CA, September 1998.

[17] I. Mithcell, "The flexible, extensible and efficient toolbox of level set methods," *Journal of Scientific Computing*, vol. 35, pp. 300-329, 2008.

[18] M. Chen, S. L. Herbert, M. S. Vashishtha, S. Bansal, and C. J. Tomlin, "Decomposition of reachable sets and tubes for a class of nonlinear systems," submitted to *IEEE Transactions on Automatic Control*