OUTPUT REACH CONTROL PROBLEM WITH APPLICATIONS TO MOTION PLANNING FOR ROBOTIC SYSTEMS

by

Zachary Kroeze

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Graduate Department of Electrical & Computer Engineering
University of Toronto

© Copyright 2019 by Zachary Kroeze
Abstract

Output Reach Control Problem with Applications to Motion Planning for Robotic Systems

Zachary Kroeze
Doctor of Philosophy
Graduate Department of Electrical & Computer Engineering
University of Toronto
2019

As electronic devices become more embedded into everyday products, an increasing number of consumer devices are being equipped with control systems. These control systems are sometimes required to achieve complex specifications such as safety constraints or switching between distinct tasks. Among a wide range of techniques to deal with the control design of complex systems, one is to partition the state space into polytopic regions, and solve a control problem on each region such that the collective behaviour of the hybrid system solves the original complex control specification. In this dissertation we consider such a methodology where the control problem on each polytopic region is called the Reach Control Problem (RCP). The RCP is to find a state feedback to make the closed-loop trajectories of an affine control system defined on a polytope reach and exit a prescribed facet of the polytope in finite time. There is already an extensive literature on the RCP. This dissertation extends this literature by considering the Output Reach Control Problem (ORCP). The ORCP is to control the output trajectory of the control system on a simplex in the output space, in contrast to control the entire state trajectory. We also extend the applications of the RCP by developing a unified framework for control with complex specifications.

This dissertation contains four distinct contributions. The first contribution provides an extension of classical linear regulator theory to affine systems and exosystems. This extension then provides the basis to develop a framework for the output reach control problem with disturbances. The second contribution leverages existing literature on viability theory to convert the ORCP in the output space on a simplex to a RCP in the full state space on a polytope. The third contribution is to propose a unified framework for solving control problems with complex specifications. Our framework is targeted at robotic systems. Finally, our fourth contribution is a set of so called motion primitives which comprise one of the layers of our aforementioned framework. These motion primitives are based on our approach to the ORCP.
Acknowledgements

I would like to begin by expressing gratitude to my supervisor, Prof. Mireille E. Broucke. Your mentorship and guidance has helped me develop in both my academic and personal life. You have taught me that while intuition can help guide the solution of a problem, only rigorous logical reasoning will ensure a solution is correct. I wish you continued success in all your future research directions.

I would like to thank the faculty, staff, and students of the Systems Control Group at the University of Toronto for such a wonderful learning experience. The framework in Chapter 6 was developed in collaboration with Mario Vukosavljev. I would like to thank him for his impeccable attention to detail. I would also like to thank my PhD committee members, Prof. Lacra Pavel and Prof. Manfredi Maggiore, for their useful suggestions.

Additionally, I would like to thank my parents Bill and Lucille, as well as my sister Katrina for always supporting me. Thank you all for being an inspiration in my life, and for motivating me to always strive for greatness.

Finally, I would like to thank my partner Allison who has endured this entire process with me. Thank you for your patience and commitment over these five years. For this and everything else that you have done for me, I will always love you.
Contents

1 Introduction 1
  1.0.1 Organization ................................................. 6
  1.0.2 Main Contributions ............................................. 6

2 Background 9
  2.1 Notation .......................................................... 9
  2.2 Functions ........................................................ 10
  2.3 Geometric Background ............................................ 12
    2.3.1 Simplices .................................................... 12
    2.3.2 Polytopes .................................................... 13
    2.3.3 Triangulation ................................................. 14
    2.3.4 Nonlinear Control Theory .................................... 16

3 Reach Control Problem 18
  3.1 Reach Control Problem .......................................... 18
  3.2 Affine Systems .................................................. 22
  3.3 Affine Feedback ................................................. 24
  3.4 Triangulation and Affine Feedback ............................. 27
  3.5 Reach Control on Polytopes .................................... 30

4 Output Regulation for Affine Systems 35
  4.1 Affine Regulator Theory ......................................... 36
    4.1.1 Output Regulation with Full Information .................. 39
    4.1.2 Output Regulation with Partial Information ................ 43
    4.1.3 Model Reduction ............................................. 46
    4.1.4 Example ..................................................... 49
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2</td>
<td>Reach Control Problem with Disturbance Rejection</td>
<td>51</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Method 1</td>
<td>52</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Method 2</td>
<td>55</td>
</tr>
<tr>
<td>4.3</td>
<td>Well-Posedness and Robust Regulation</td>
<td>61</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Well-Posedness</td>
<td>61</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Structural Stability</td>
<td>65</td>
</tr>
<tr>
<td>5</td>
<td>Output Reach Control Problem</td>
<td>76</td>
</tr>
<tr>
<td>5.1</td>
<td>Problem Statement</td>
<td>76</td>
</tr>
<tr>
<td>5.2</td>
<td>From ORCP to RCP</td>
<td>78</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Computing $\mathcal{P}$</td>
<td>79</td>
</tr>
<tr>
<td>5.3</td>
<td>Viable Polytope for Reach Control</td>
<td>82</td>
</tr>
<tr>
<td>5.4</td>
<td>Main Results</td>
<td>85</td>
</tr>
<tr>
<td>5.5</td>
<td>Example</td>
<td>87</td>
</tr>
<tr>
<td>6</td>
<td>Motion Planning Framework for Integrator Systems</td>
<td>90</td>
</tr>
<tr>
<td>6.1</td>
<td>Related Literature</td>
<td>93</td>
</tr>
<tr>
<td>6.2</td>
<td>Problem Statement</td>
<td>95</td>
</tr>
<tr>
<td>6.3</td>
<td>Modular Framework</td>
<td>96</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Output Transition System</td>
<td>97</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Maneuver Automaton</td>
<td>98</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Product Automaton</td>
<td>102</td>
</tr>
<tr>
<td>6.3.4</td>
<td>High-Level Plan</td>
<td>103</td>
</tr>
<tr>
<td>6.4</td>
<td>Main Results</td>
<td>111</td>
</tr>
<tr>
<td>6.5</td>
<td>Motion Primitives for Integrator Systems</td>
<td>117</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Single Integrator</td>
<td>118</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Double Integrator</td>
<td>119</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Triple Integrator</td>
<td>122</td>
</tr>
<tr>
<td>6.6</td>
<td>Applications</td>
<td>126</td>
</tr>
<tr>
<td>6.6.1</td>
<td>Robotic Manipulator Modeling</td>
<td>126</td>
</tr>
<tr>
<td>6.6.2</td>
<td>Preparing the OTS</td>
<td>126</td>
</tr>
<tr>
<td>6.6.3</td>
<td>Learning Real Time A*</td>
<td>126</td>
</tr>
<tr>
<td>6.6.4</td>
<td>Robotic Manipulator Experimental Results</td>
<td>127</td>
</tr>
<tr>
<td>6.6.5</td>
<td>Quadrocopter Modeling</td>
<td>128</td>
</tr>
<tr>
<td>Chapter</td>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>6.6.6</td>
<td>Interfacing Multiple Quadrocopters</td>
<td>129</td>
</tr>
<tr>
<td>6.6.7</td>
<td>Quadrocopter Experimental Results</td>
<td>130</td>
</tr>
<tr>
<td>7</td>
<td>Motion Primitives for Integrator Systems</td>
<td>134</td>
</tr>
<tr>
<td></td>
<td>7.1</td>
<td>Preliminaries</td>
</tr>
<tr>
<td></td>
<td>7.2</td>
<td>( \mathcal{H} ) Motion Primitive</td>
</tr>
<tr>
<td></td>
<td>7.3</td>
<td>( \mathcal{F} ) and ( \mathcal{B} ) Motion Primitives</td>
</tr>
<tr>
<td>8</td>
<td>Conclusion</td>
<td>151</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>154</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The relationship between the classes of functions used in this dissertation</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>A simplex in $\mathbb{R}^2$</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>A polytope in $\mathbb{R}^2$</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Example for Definition 2.3.2</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>A triangulation of $\mathcal{P}$ with respect to $\mathcal{O}$</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Notation for the reach control problem</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>The invariance conditions state that $y_i := Av_i + Bu_i + a \in C(v_i)$ for $i = 0, \ldots, n$</td>
<td>21</td>
</tr>
<tr>
<td>3.3</td>
<td>Geometric constructs for the reach control problem</td>
<td>21</td>
</tr>
<tr>
<td>3.4</td>
<td>Continuous Assignment Along $\mathcal{O}\mathcal{P}$ of Vectors in $\mathcal{B}$</td>
<td>29</td>
</tr>
<tr>
<td>3.5</td>
<td>Closed-loop vector field on $\mathcal{S}$ for Example 3.4.1</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>Error vs. time</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>The triangulated state space $\mathcal{P}$ with desired closed-loop velocity vectors at the vertices</td>
<td>59</td>
</tr>
<tr>
<td>4.3</td>
<td>Closed-loop phase portrait</td>
<td>60</td>
</tr>
<tr>
<td>5.1</td>
<td>Progression of the viability algorithm</td>
<td>88</td>
</tr>
<tr>
<td>6.1</td>
<td>A two output $(p = 2)$ example of a reach-avoid task. Shown on the left is the feasible space $\mathcal{P}$ consisting of 15 non-obstacle boxes (not red) and the goal region $\mathcal{O}\mathcal{P}$ (green). The output transition system (OTS), which abstracts the box regions and their neighbour connectivity, is shown on the right. Shown below, the possible offsets towards a neighbouring box are labelled using $\Sigma = {-1, 0, 1}^2$, which includes the box edges, vertices, and interior</td>
<td>91</td>
</tr>
<tr>
<td>6.2</td>
<td>Our modular framework consists of five modules</td>
<td>96</td>
</tr>
<tr>
<td>6.3</td>
<td>A fragment of a generic PA, showing a state and its neighbours</td>
<td>103</td>
</tr>
</tbody>
</table>
6.4 At the top, a PA is depicted for a single output system having three motion primitives \( M = \{H, F, B\} \) over three boxes \( L = \{l_j \mid j = 1, 2, 3\} \). The numbers 1, -1 \( \in \Sigma \) on the edges (shown as arrows) are the corresponding labels. The bottom pictures show the reduced set of transitions induced by control polices \( c_1, c_2 \in C \).

6.5 This figure shows a discrete control strategy for the scenario shown in Figure 6.1.

6.6 The maneuver automaton for the single integrator dynamics with \( p = 1 \). There are three motion primitives: Hold \((H)\), Forward \((F)\), and Backward \((B)\).

6.7 The closed-loop vectors at the vertices for the Hold, Forward, and Backward motion primitives for single integrator dynamics.

6.8 The maneuver automaton for the double integrator dynamics with \( p = 1 \). There are three motion primitives: Hold \((H)\), Forward \((F)\), and Backward \((B)\).

6.9 The closed-loop vectors at the vertices for the Hold, Forward, and Backward motion primitives for double integrator dynamics.

6.10 The closed-loop vector fields in the \((x_1, x_2)\) state space for the Hold, Forward, and Backward motion primitives for double integrator dynamics. From the vector field it can be verified that all state trajectories in Hold are stabilized, all state trajectories in Forward reach the right output face (event label 1, with positive velocity), and all state trajectories in Backward reach the left output face (event label -1, with negative velocity).

6.11 The closed-loop vectors at the vertices for the Hold, Forward, and Backward motion primitives for triple integrator dynamics.

6.12 A closer analysis of the Forward mode for triple integrator dynamics.

6.13 A single robotic manipulator avoids a wall to reach its goal location using a LRTA* algorithm.

6.14 Two robotic manipulators attempt to exchange end effector positions while avoiding collisions using a LRTA* algorithm. Since the heuristic function \( \text{Heur}_2(l, m) \) does not take into account collisions between manipulators, the objective is not achieved after 50 seconds.

6.15 Two robotic manipulators exchange end effector positions while avoiding collisions using a LRTA* algorithm. Since the heuristic function \( \text{Heur}_1(l, m) \) takes into account collisions between manipulators, after 11 seconds the goal location is reached.
6.16 Interface between multiple vehicles and the abstract framework with \( p = 3N \) outputs. The internal state is updated via external state measurements (assumed to be given). The hybrid controller internal state consists of the joint state measurement of all the vehicles; this includes the current (joint) box, \( l \), and the current (joint) motion primitive, \( m \). The vehicle state is again used to compute the feedback controls.

6.17 Experimental results for the scenario where two quadrocopters must switch places through a narrow passage, where Vehicle 1 (blue) starts on the left and Vehicle 2 (orange) starts on the right. The grayed out trajectories show the full path followed by the vehicles; the color portion corresponds to the indicated time interval. The maneuver is shown over three time segments, with goals highlighted in green on the right. We observe that since Vehicle 2 gets to the passage first, it makes room and waits for Vehicle 1 to pass first. Noisiness in the trajectories is due to mutual aerodynamic effects, to be contrasted with Figure 6.19. Although these effects are not accounted for, our hybrid feedback controller safely completes the task.

6.18 The same scenario as in Figure 6.17 above, but now an unmodelled wind disturbance causes a delay in Vehicle 2. To compensate, Vehicle 1 makes room and waits for Vehicle 2 to pass first, opposite to the nominal case. The same hybrid controller was used as in the nominal run of Figure 6.17.

6.19 A single-vehicle run for a similar scenario as considered in Figures 6.17 and 6.18. The trajectory is smooth and efficient, which illustrates that the noisy trajectories in the multi-vehicle case arise from mutually caused aerodynamic effects. The blue trajectory represents the motion for the given time range from 0 to 15 seconds.

7.1 The maneuver automaton for \( n \)-th order integrator dynamics. There are three motion primitives: Hold \( (H) \), Forward \( (F) \), and Backward \( (B) \).

7.2 A polytope of dimension \( \kappa = 2 \) which satisfies Properties 7.2.1.

7.3 A polytope of dimension \( \kappa = 3 \) which satisfies Properties 7.2.1.
List of Symbols and Acronyms

General:

\[ x := y \quad x \text{ is defined as } y \]

\[ \forall \quad \text{for all} \]

\[ \exists \quad \text{there exists} \]

\[ \Rightarrow \quad \text{implies} \]

\[ \Leftrightarrow \quad \text{equivalent} \]

\[ C^1 \quad \text{continuously differentiable} \]

\[ U \quad \text{a control class such as open-loop controls, continuous state feedback, affine feedback, etc} \]

Sets and Fields:

\[ \mathbb{R} \quad \text{field of real numbers} \]

\[ \mathbb{R}^n \quad \text{the set of } n \text{ tuples of the real numbers} \]

\[ \mathbb{R}^{n \times m} \quad \text{the set of } n \times m \text{ matrices with real elements} \]

\[ \mathbb{R}_+ \quad \text{the set of non-negative real numbers} \]

\[ 2^{\mathbb{R}^n} \quad \text{the power set of } \mathbb{R}^n \text{ (the set of all subsets of } \mathbb{R}^n) \]

\[ C \quad \text{field of complex numbers} \]

\[ C^- \quad \text{open left-half complex plane} \]

\[ \emptyset \quad \text{the empty set} \]

\[ 0 \quad \text{the subset of } \mathbb{R}^n \text{ containing only the zero vector} \]
\( x \in \mathcal{K} \)  \( x \) is an element of the set \( \mathcal{K} \)

\( \dim(\mathcal{K}) \)  affine dimension of the set \( \mathcal{K} \) (dimension of the affine hull of \( \mathcal{K} \))

\( \mathcal{K}_1 \subset \mathcal{K}_2 \)  the set \( \mathcal{K}_1 \) is contained in the set \( \mathcal{K}_2 \)

\( \mathcal{K}_1 \subseteq \mathcal{K}_2 \)  the set \( \mathcal{K}_1 \) is contained in or it is equal to the set \( \mathcal{K}_2 \)

\( \mathcal{K}_1 \cap \mathcal{K}_2 \)  the intersection of the sets \( \mathcal{K}_1 \) and \( \mathcal{K}_2 \)

\( \mathcal{K}_1 \cup \mathcal{K}_2 \)  the union of the sets \( \mathcal{K}_1 \) and \( \mathcal{K}_2 \)

\( \mathcal{K}_1 \setminus \mathcal{K}_2 \)  elements of the set \( \mathcal{K}_1 \) not contained in the set \( \mathcal{K}_2 \)

\( \mathcal{K}^c \)  the complement of \( \mathcal{K} \) (\( \mathcal{K}^c := \mathbb{R}^n \setminus \mathcal{K} \))

\( \overline{\mathcal{K}} \)  the closure of \( \mathcal{K} \)

\( \mathcal{K}^\circ \)  the interior of \( \mathcal{K} \)

\( \partial \mathcal{K} \)  the boundary of \( \mathcal{K} \) (\( \overline{\mathcal{K}} \setminus \mathcal{K}^\circ \))

\( \text{ri} (\mathcal{K}) \)  the relative interior of \( \mathcal{K} \)

\( \text{rb} (\mathcal{K}) \)  the relative boundary of \( \mathcal{K} \) (\( \overline{\mathcal{K}} \setminus \text{ri} (\mathcal{K}) \))

\( \text{co} (\mathcal{K}) \)  the convex hull of the set \( \mathcal{K} \)

\( \text{co} \{v_1, v_2, \ldots\} \)  the convex hull of a set of points \( v_i \in \mathbb{R}^n \)

\( \text{aff} (\mathcal{K}) \)  the affine hull of the set \( \mathcal{K} \)

\( \text{aff} \{v_1, v_2, \ldots\} \)  the affine hull of a set of points \( v_i \in \mathbb{R}^n \)

\( T_{\mathcal{K}}(x) \)  the Bouligand tangent cone to a set \( \mathcal{K} \subset \mathbb{R}^n \) at point \( x \)

\( \mathcal{B} \)  the open ball of radius 1 centered at the origin

\( \mathcal{B}_\delta(x) \)  the open ball of radius \( \delta \) centered at \( x \)

Matrices:

\( A : \mathcal{X} \mapsto \mathcal{Y} \)  \( A \) maps a vector in domain \( \mathcal{X} \) to a vector in codomain \( \mathcal{Y} \)

\( A^T \)  transpose of \( A \)

\( A^{-1} \)  inverse of a square matrix \( A \)
\( \sigma(A) \) spectrum of a square matrix \( A \)

\( I \) identity

**Subspaces:**

\( \text{Im } A \) image or range of \( A \)

\( \text{Ker } A \) kernel or null space of \( A \)

**Vectors and Scalars:**

\( x \cdot y \) the dot product of two vectors \( x, y \in \mathbb{R}^n \)

\( x^T \) transpose of the vector \( x \)

\( L_f V(x) \) the Lie derivative of function \( V : \mathbb{R}^n \to \mathbb{R} \) with respect to function \( f : \mathbb{R}^n \to \mathbb{R}^n \)

**Units:**

\( ^\circ C \) degree Celsius

\( s \) seconds

**Acronyms:**

LMI Linear Matrix Inequality

PWA Piecewise Affine

PWL Piecewise Linear

RCP Reach Control Problem

MRCP Monotonic Reach Control Problem

LP Linear Programming problem

NP Nondeterministic Polynomial-time

w.l.o.g. without loss of generality
Chapter 1

Introduction

Stabilization techniques for control systems have been studied since the early sixties, and there is little left to be done on the subject. Modern control problems instead tend to deal with complex control specifications. Some examples of such control specifications are to perform several reach-avoid tasks in succession, or to perform synchronized motion between several agents. These control tasks may also need to satisfy certain safety constraints, such as avoiding collisions with the environment. Additionally, there may be state restrictions which take into account maximum or minimum velocity requirements. Such control specifications are difficult to address directly by standard control techniques.

The operating premise of this dissertation is that dedicated control techniques are required for dealing with complex control specifications. We are particularly informed by recent advancements in reach control theory. The central problem of reach control theory, called the Reach Control Problem (RCP), is to design a feedback controller to make the trajectories of an affine control system reach a prespecified facet in finite time. The RCP is distinct from standard control techniques in that the focus is on the transient behaviour of the system. The desired transient behaviour is characterized through linear inequalities, which encode state constraints, and through a qualitative description of temporal events. In particular, these linear inequalities form the polytope $\mathcal{P} \subset \mathbb{R}^n$ bounded over which the affine system evolves. By having the affine system evolve over a collection of polytopes, the complex control specifications discussed above can be achieved.

We now present the progression of the RCP since its inception. The reach control problem was first presented by Habets and van Schuppen in 2001 [25]. Their 2004 paper [6] studied the solvability of the RCP by continuous piecewise affine feedbacks over a polytope $\mathcal{P}$. The paper introduced the invariance conditions which ensure that trajectories of the system do not exit the polytope $\mathcal{P}$ through facets which
are not exit facets. The paper also proposed an elegant method for computing affine feedbacks on a simplex, given control values at the vertices. In 2006 Habets, Collins, and van Schuppen [7], as well as Roszak and Broucke [9] presented the foundation for the modern formulation of the RCP. The difficult problem of the RCP on polytopes, was converted into the more tractable problem of the RCP on simplices. Necessary and sufficient conditions were obtained to solve the RCP by affine feedbacks. The authors in [9] introduced a flow condition, which guarantees that trajectories originating within the simplex, leave the simplex in finite time. It was shown that the flow condition was equivalent to the lack of closed-loop equilibria in the simplex. The flow condition in [9] was first presented in terms of bilinear inequalities, but an algorithm was presented to convert the bilinear inequalities to a series of linear programming problems. Unfortunately, the complexity of the linear programming solution increases exponentially with the system dimension.

The progression of the RCP from here tends in three different directions. The first direction began with Broucke in [16] and was concerned with replacing the computational task of finding a flow condition with a more geometric approach to finding equilibria. In [16], Broucke developed a preferred triangulation of the state space which made the location of the equilibria explicit. Moarref, Ornik and Broucke studied the case where the equilibria could not be removed from the simplex, the study of the so called topological obstruction to solvability of the RCP, in [20]. A more general result on the topological obstruction was presented by Ornik and Broucke in [18].

The second direction for the progression of the RCP was to study various classes of controllers which solve the RCP. Continuous controllers were studied in [16], discontinuous feedbacks in [17, 1], output feedbacks in [22] and time varying feedbacks in [1]. While the study of the various feedback methods helped advance the RCP theory, the most definitive results are focused on the RCP by affine feedbacks [7, 2, 92, 1].

The third direction was to study the RCP on polytopes, and not on simplices. While the original formulation of the RCP was on polytopes, there were no necessary and sufficient conditions for its solvability. The monotonic reach control problem (MRCP) on a polytope was formulated in [8]. The MRCP differs from the RCP in that it adds a flow condition into the problem statement. Necessary and sufficient conditions are given to solve the MRCP, and it is shown that solvability of the MRCP does not depend on the triangulation of the polytope.

In addition to the vast theoretical developments of the RCP since 2001, there have also been meaningful applications. Some of the most recent applications for the RCP include control of the Canadarm [26], parallel parking [27], and adaptive cruise control [28]. In the case of the Canadarm, the end effector is feedback linearized to obtain a linear system. This combination of a preliminary nonlinear feedback
transformation with a secondary RCP controller to achieve the control objective will also be used in this dissertation. In contrast, in both the parallel parking and adaptive cruise control applications, a linearization about specific regions of the state space is done. This method is shown to be sufficient to achieve the control objective of the nonlinear system. These applications show the use of the RCP beyond its original formulation on piecewise affine systems.

The RCP assumes that the polytopic state space has been triangulated. There are both benefits and drawbacks to this approach. There are two primary benefits of working with a triangulated state space, namely working with simplices. First, control specifications based on linear inequality constraints on the states lead to a state space that is a polytope. Any polytope can be triangulated to simplices. Second, the simplex is a canonical object to represent bounded $n$-space. Hence, a design method for one simplex works for all. This trivial observation lead to the use of simplices to formulate algebraic topology in terms of simplices by Poincaré. Thanks to the canonical structure of simplices, it has been possible to develop theory for solving certain reachability problems in terms of favorable classes of feedback controls such as affine feedbacks. None of these developments would have been as efficacious or elegant using other bounded geometric sets.

An alternate strategy to solve control problems with complex specifications is to use abstractions of the system [43]. These methods generally rely on computations of the reachable set of states from some initial state set. Unfortunately, computation of reachable sets either in forward or backward time is highly computationally complex, particularly if accuracy in the calculation is necessary to meet safety requirements. Discretization of both states and inputs leads to a computationally infeasible representation of the dynamics of the control system.

Using reach control theory, no approximations or discretizations of the state space or input space are introduced. Transient behaviour is guaranteed to remain within the simplex. Finally the reachable set is intrinsic in the problem statement. Once a series of RCP problems is solved, a discrete abstraction is formed. From this, any method based on discrete abstractions can be used to solved the high-level problem.

Despite these benefits of working with simplices, there are also drawbacks. The need to triangulate the state space increases the computational complexity. In addition, the flow condition must be satisfied at each of the vertices of a simplex, and this may be hard to achieve in higher dimensions. Furthermore, not all states of the system may have constraints on them. Given that solving the RCP has exponential complexity in the state dimension, a formulation in the output space can potentially significantly reduce complexity. These considerations led us to the development of the Output Reach Control Problem. We introduce for the first time the notion of an output for the RCP, and are concerned with the reachability
of the output trajectory of an affine system to a prespecified facet of the simplex. We now highlight the problems addressed in this dissertation.

The overarching theme of this dissertation is to propose various strategies to tackle the complexity problem inherent in the RCP. In Chapters 4-5, our strategy is to formulate variants of the RCP focused on the output space, rather than the full state space. This is the first time that an output has been considered in the RCP. In Chapter 4 we use regulator theory to solve output regulation problems on simplices by two methods. In Chapter 5, we formulate and solve the Output Reach Control Problem (ORCP) using methods inspired by viability theory. A different strategy to deal with the complexity of the RCP is adopted in Chapters 6 and 7. Here we use the notion of motion primitives to solve a highly complex motion planning problem. In so doing, we completely bypass the inherent difficulty of triangulating the full state space. Below we provide more details on the methods introduced in the dissertation.

In Chapter 4 we approach solving the RCP in the output space using regulator theory, a theory naturally structured to deal with specifications on the outputs of a control system. There are two variants of the problem. In the first method, we consider an affine control system defined on a set of simplices in a polytopic state space. For each simplex, we have in hand a reach controller that captures some desirable behavior either on the simplex or on the projection of the simplex to the output space. The reach controller does not necessarily solve the RCP on its associated simplex (because the problem may not be solvable). However it should have some useful behaviors that are worthy of tracking in the output space. As such, this formulation of the RCP in the output space gives the extra flexibility that the RCP may not be solved in the full state space with the proposed reach controller; yet, the behavior of specific closed-loop trajectories may be effective to provide reasonable behavior in the output space. By setting up a tracking problem, some desired behavior that approximately solves the RCP in the output space is achieved asymptotically. To formulate this problem as a problem of regulator theory, we define an exosystem that models one specific reference trajectory of the closed-loop system obtained using the family of reach controllers. The regulation problem is then to make the output of the control system track the desired reference trajectory generated by the first exosystem while rejecting disturbances, which are modelled by a second exosystem. This problem is solved using classic regulator theory.

In the second method, we consider again an affine control system defined on a set of simplices in a polytopic state space. For each simplex, we have a reach controller that captures some desirable behavior either on the full simplex or on the projection of the simplex to the output space. As before, the reach controllers do not necessarily solve the RCP on their respective simplices. However, the behavior elicited by the reach controllers is sufficiently effective that we simply adopt these controllers. Also as
before, there is a persistent unmeasurable disturbance acting on the closed-loop system. We model the disturbance by an exosystem. Since there is no reference trajectory to track, there is no need for another exosystem. The control problem is to design a regulator to reject the disturbance in the output of the closed-loop system formed by applying the proposed reach controllers. This formulation of the problem makes it a pure disturbance rejection problem in the output space. Of course, if we were to select the output equal to the state, then this problem has independent value not only a solution of the RCP in the output space, but also as a solution of the RCP with disturbances.

To properly formulate the problems just described, we require two extensions of classic regulator theory. First, we require a regulator theory for affine control systems. Second, we require the ability to track exosystems that are stable, not only those that are marginally stable. Both extensions are developed in the first part of Chapter 4.

In Chapter 5 we formulate and solve the ORCP by using methods inspired by viability theory [15, 21]. The main idea is to construct a maximal (or approximately maximal) polytope $P$ in the state space and an associated piecewise affine controller defined on $P$, such that the closed-loop system solves the ORCP (when solutions are projected to the output space). A procedure to construct piecewise affine controllers in the full state space on a polytope is available in [8]. Therefore, we do not focus on the control synthesis aspect of the problem. Rather, we focus our attention on the necessary conditions, called invariance conditions, that make the procedure of [8] feasible.

Our methodology to find a (close to) maximal polytope in the state space on which the invariance conditions are solvable is to adapt the algorithm in [23] for finding maximal positively invariant polytopes under linear dynamics. The main idea of our adaptation is that, instead of beginning with an initial positively invariant polytope, we begin with an exit facet in the state space that is determined by the exit facet in the output space of the ORCP that we want to solve. Then we iteratively “grow” a polytope such that, at each iteration, the invariance conditions are solvable for the current iterate.

Chapters 6 and 7 take a completely different approach, yet the theme remains the same. Rather than considering general affine control systems, we focus on integrator systems. We specify atomic behaviors in the output space, called motion primitives. Because we are dealing with a specific class of systems and specific closed-loop behaviors, one can readily apply the viability theory algorithm in Chapter 7 to obtain polytopes on which piecewise affine feedbacks can be defined to achieve the goal of each motion primitive.

In essence, the idea of motion primitives is to devise closed-loop behavior using reach controllers with the state space triangulation already “built in” to the control design. The difficulty of constructing a triangulation is completely bypassed by this approach. On the other hand, the price to be paid is
Chapter 1. Introduction

that we only work with integrator systems. Such systems have received considerable attention in the multi-agent systems community. Many highly complex robotic models, such as the quadrocopter model, can be reduced to a collection of double integrator models by using feedback linearization. This fact has inspired our choice of the integrator model on which to design motion primitives.

1.0.1 Organization

This dissertation is organized as follows. The next chapter provides a mathematical background. In Chapter 3 we review the basic results for solvability of the RCP on simplices and solvability of the MRCP on polytopes. In Chapter 4 we extend the study of regulator theory to affine systems. We then formulate a disturbance rejection problem for the RCP using this extension. In Chapter 5 we formulate the ORCP on a simplex in the output space, and we lift it to a polytope in the full state space using a modified viability algorithm. Using existing RCP techniques, we solve the RCP on the full dimensional polytope, and show that this solves the ORCP. In Chapter 6 we formulate a modular motion planning framework for integrator systems. We provide specific motion primitives for single and double integrator systems. We apply this framework to reach-avoid planning problems for quadrocopters and robotic manipulators. In Chapter 7 we develop an algorithm to construct motion primitives for \( n \)-th order integrator systems. Finally in Chapter 8 we summarize the dissertation and show future research directions.

1.0.2 Main Contributions

The dissertation develops techniques to solve the ORCP, and formulates a motion planning framework for integrator systems. We first summarize the direct contributions and then discuss how the development fits into the larger framework of control problems with complex specifications:

1. Chapter 4

- In Theorem 4.1.4 we extend the study of regulator systems to affine systems
- In Theorem 4.2.1 and Theorem 4.2.2 we provide two methods for applying this extension to solving a disturbance rejection problem for the RCP
- We apply the above theory to the control of the end effector of the Canadarm

2. Chapter 5

- In Problem 5.1.2 we formulate the ORCP on a simplex
• In Section 5.2.1 we convert the ORCP on a simplex to a RCP on a polytope
• In Algorithm 5.3.1 a viability algorithm is presented to achieve a polytope that satisfies certain constraints on the vector field.
• In Theorem 5.4.1 we show that solving the RCP on the refined polytope solves the ORCP

3. Chapter 6
• In Section 6.3 we formulate a motion planning framework for robotic systems
• In Theorem 6.3.1 we develop a one step cost function for the high level control strategy on the PA
• In Theorem 6.4.1 we solve a reach-avoid problem for robotic systems
• We apply our framework to reach-avoid problems for quadrocopters and robotic manipulators

4. Chapter 7
• We provide two algorithms, Algorithms 2 and 3, to construct high order motion primitives in a recursive manner.
• In Theorems 7.2.6 and 3 we prove the correctness of these algorithms.

We now mention how our work fits into a larger framework for control problems with complex specifications. The most popular framework at this time is control problems with linear temporal logic (LTL) specifications. Also, typically it is assumed that the LTL specifications are formulated in terms of linear inequality constraints in the state space. The collection of linear inequality constraints induce a partition of the state space space consisting of a collection of polytopes. In the standard formulation, a discrete transition system is defined on these polytopes. The transitions encode for which contiguous polytopes it is possible to find a feedback controller that drives all trajectories starting in one polytope into the other. Finally, model checking or any other LTL synthesis method is applied to the transition system to verify that the LTL specification can be met. The resulting low-level controllers defined over the polytopes then form a hybrid system that solves the original control problem.

Our work in Chapters 4 and 5 fits directly within this methodology. We present methods for designing the low-level controllers over each polytope. In Chapter 4 we consider the presence of disturbances, and in Chapter 5 we consider a problem where only the outputs are in interest in the control specification. In these chapters, we do not deal with the high level problem posed on the transition system.

In contrast, Chapters 6 and 7 put forward a completely different method of solving a control problem with complex specifications. We do away with the standard method to construct a discrete transition
system based on a partition of the state space, which itself arises from linear inequality constraints on the states. In fact, in Chapters 6 and 7, no partition of the state space is obtained. Rather, we focus on a generic partition of the output space into boxes. Then we restrict the dynamics of the control system to a finite set of behaviours called motion primitives, with associated feedback controllers. Once we have these finite behaviours in the output space, then polytopes in the state space are devised to guaranteed correct behaviour in the output space. These polytopes no longer fit together to form a partition of the state space.

Our approach bypasses the very difficult problem of how to partition a high dimensional state space to ensure that a specification will be met. Of course, our method does not work for all systems. We focus on systems with special structure, such as integrator systems.
Chapter 2

Background

In this chapter we review the preliminary mathematical definitions and results needed for the RCP. In Section 2.1 we introduce the notations used in the dissertation. Section 2.2 reviews functions. Section 2.3 provides a geometric background including simplices, polytopes, and triangulation.

2.1 Notation

The notation \( C^1 \) denotes continuously differentiable, and the symbol \( U \) represents a control class such as open-loop controls, continuous state feedback, affine feedback, etc.

The notation \( \mathbb{R}^{n \times m} \) denotes the set of \( n \times m \) matrices with real numbers, and the notation \( \mathbb{R}_+ \) denotes the set of nonnegative real numbers. The notation \( 2^{\mathbb{R}^n} \) denotes the power set of \( \mathbb{R}^n \), the set of all subsets of \( \mathbb{R}^n \). The notation \( \mathbb{C} \) denotes the field of complex numbers, and the notation \( \mathbb{C}^- \) denotes the open left-half complex plane. Let \( \alpha > 0 \), \( \mathbb{C}^-_\alpha := \{ x \in \mathbb{C} : \text{Re} \, x < -\alpha \} \) is the \( \alpha \)-shifted open left-half plane, and \( \mathbb{C}^+_\alpha := \{ x \in \mathbb{C} : \text{Re} \, x \geq -\alpha \} \) is the \( \alpha \)-shifted closed right-half plane. The notation \( \emptyset \) denotes the empty set, while the notation \( 0 \) denotes the subset of \( \mathbb{R}^n \) containing only the zero vector. The notation \( B \) denotes the open ball of radius 1 centered at the origin, and the notation \( B_\delta(x) \) denotes the open ball of radius \( \delta \) centered at \( x \).

Let \( K \subset \mathbb{R}^n \) be a set. The notation \( \dim(K) \) denotes the affine dimension of the set \( K \), which is the dimension of the affine hull of the set \( K \) (the smallest affine set containing \( K \)). For the set \( K \), the complement of \( K \) is \( K^c := \mathbb{R}^n \setminus K \), the closure is \( \overline{K} \), the interior is \( K^\circ \), and the boundary is \( \partial K := \overline{K} \setminus K^\circ \). The relative interior is denoted \( \text{ri} \, K \) and the relative boundary of \( K \), denoted \( \text{rb} \, K \) is \( \overline{K} \setminus \text{ri} \, K \). The notation \( \text{co} \, K \) denotes the convex hull of the set \( K \) (the smallest convex set containing \( K \)), and \( \text{aff} \, K \) denotes the affine hull of the set \( K \). The notation \( \text{co} \, \{ v_1, v_2, \ldots \} \) denotes the convex hull of a set of
points \( v_i \in \mathbb{R}^n \), while \( \text{aff} \{ v_1, v_2, \ldots \} \) denotes the affine hull of a set of points \( v_i \in \mathbb{R}^n \).

Given \( \Omega \subset \mathbb{R}^n \), and \( x \in \Omega \), let \( T_\Omega(x) \) denote the Bouligand tangent cone of to the set \( \Omega \) at \( x \).

Let \( A : X \rightarrow Y \) denote a matrix that maps a vector in domain \( X \) to a vector in codomain \( Y \). The notation \( A^T \) denotes the transpose of the matrix \( A \), and \( \|A\| \) denotes the induced norm of the matrix \( A \). If \( A \) is a square matrix, then \( A^{-1} \) denotes the inverse of \( A \), and \( \sigma(A) \) denotes the spectrum of \( A \). Let \( \text{Im} A \) denote the image or range of \( A \), and let \( \text{Ker} A \) denote the kernel or null space of \( A \). The notation \( I \) denotes the identity matrix.

For vectors \( x, y \in \mathbb{R}^n \), \( x^T \) denotes the transpose of \( x \), \( \|x\| \) denotes the Euclidean norm of \( x \), and \( x \cdot y \) denotes the dot product of the two vectors. The notation \( x \prec y \) (\( x \preceq y \)) means \( x_i < y_i \) (\( x_i \leq y_i \)) for all \( 1 \leq i \leq n \). Let \( X, Y \) be vector spaces. If \( f : X \rightarrow Y \) is a surjective mapping, and \( W \subset Y \), then \( f(f^{-1}(W)) = W \), where \( f^{-1}(W) = \{ x \in X \mid f(x) \in W \} \).

### 2.2 Functions

In this section we review the definitions of some classes of functions that will be used in the feedback laws solving RCP throughout the dissertation.

**Definition 2.2.1** ([109]). A function \( f : X \rightarrow Y \) is Lipschitz continuous at \( x_0 \in X \) if

\[
(\exists \delta, L > 0) \ (\forall x \in X) \ x \in B_\delta(x_0) \Rightarrow \|f(x) - f(x_0)\| \leq L \|x - x_0\|.
\]

We call \( L \) a Lipschitz constant at \( x_0 \). If \( f \) is Lipschitz continuous at every \( x \in X \), then \( f \) is said to be locally Lipschitz on \( X \).

Now we define affine functions, which are used to solve RCP on simplices (Chapter 3).

**Definition 2.2.2** ([76, 109]). A function \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is affine if for every \( x, y \in \mathbb{R}^n \) and \( \alpha \in \mathbb{R} \), we have:

\[
f(\alpha x + (1 - \alpha)y) = \alpha f(x) + (1 - \alpha)f(y).
\]

Any linear function is affine, but the converse is not true.

**Remark 2.2.1.** If \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is affine, then there exist a matrix \( F \in \mathbb{R}^{m \times n} \) and a vector \( g \in \mathbb{R}^m \) such that \( f(x) = Fx + g \), \( x \in \mathbb{R}^n \).

Now we review piecewise affine (PWA) functions that are used to solve RCP on polytopes in Chapter 3.
Figure 2.1: The relationship between the classes of functions used in this dissertation

**Definition 2.2.3** ([103]). A function $f: \mathbb{R}^n \to \mathbb{R}^m$ is a piecewise affine function if there exist finite number of sets $\Omega_1, \cdots, \Omega_L$ such that $\bigcup_{i=1}^L \Omega_i = \mathbb{R}^n$, and $f(x)$ is affine on each $\Omega_i$. In particular, for each $i = 1, \cdots, L$, there exist $F_i \in \mathbb{R}^{m \times n}$ and $g_i \in \mathbb{R}^m$ such that $f(x) = F_i x + g_i$, $x \in \Omega_i$. Also, $f$ is a continuous piecewise affine function if additionally:

$$F_i x + g_i = F_j x + g_j, \quad x \in \Omega_i \cap \Omega_j, \quad i, j \in \{1, \cdots, L\}.$$

Now we define multi-affine functions.

**Definition 2.2.4** ([69]). A function $f: \mathbb{R}^n \to \mathbb{R}^m$ is multi-affine if it is a polynomial in the indeterminates $x_1, \cdots, x_n$ with the property that the degree of $f$ in any of the indeterminates $x_1, \cdots, x_n$ is either 0 or 1. Equivalently, $f$ has the form:

$$f(x_1, \cdots, x_n) = \sum_{i_1, \cdots, i_n \in \{0, 1\}} c_{i_1, \cdots, i_n} x_1^{i_1} \cdots x_n^{i_n}$$

with $c_{i_1, \cdots, i_n} \in \mathbb{R}^m$ for all $i_1, \cdots, i_n \in \{0, 1\}$ and using the convention that if $i_k = 0$, then $x_k^{i_k} = 1$.

Figure 2.1 shows the relationship between the different classes of functions defined above.
2.3 Geometric Background

In this dissertation we study the reach control problem on simplices and on polytopes (Chapter 3). Hence, it is important to review their geometric properties.

2.3.1 Simplices

A simplex is a generalization of the notion of a triangle to arbitrary dimension. Let $V := \{v_0, \cdots, v_n\}$ be a set of $n + 1$ points in $\mathbb{R}^n$. We say $\{v_0, \cdots, v_n\}$ are affinely independent if they do not lie in an $(n - 1)$-dimensional plane in $\mathbb{R}^n$. Equivalently, $\{v_0, \cdots, v_n\}$ are affinely independent if $\{v_1 - v_0, \cdots, v_n - v_0\}$ are linearly independent. An $n$-dimensional simplex is the convex hull of $n + 1$ affinely independent points in $\mathbb{R}^n$. Suppose $\{v_0, \cdots, v_n\}$ are affinely independent, and define the $n$-dimensional simplex $S := \text{co} \ \{v_0, \cdots, v_n\}$.

Figure 2.2 shows a simplex in $\mathbb{R}^2$. A face of $S$ is any sub-simplex of $S$ which makes up its boundary. A facet is an $(n - 1)$-dimensional face of $S$. We denote the facets of $S$ by $F_0, \cdots, F_n$. Our numbering convention is that each facet is indexed by the vertex it does not contain. Let $h_i$ denote the unit normal vector to $F_i$ pointing outside $S$. It is possible to define the simplex $S$ using these normal vectors. In particular, there exist $\alpha_0, \cdots, \alpha_n \in \mathbb{R}$ such that

$$S = \{x \in \mathbb{R}^n \mid h_i \cdot x \leq \alpha_i, \forall i \in \{0, \cdots, n\}\}.$$

The following lemma summarizes useful properties of simplices.

**Lemma 2.3.1 ([84]).** Let $S$ be an $n$-dimensional simplex. Then the following hold:
Chapter 2. Background

2.3.2 Polytopes

An $n$-dimensional polytope is the convex hull of a finite set of points in $\mathbb{R}^n$ [74]. In particular, let $\{v_1, \cdots, v_p\}$ be a set of points in $\mathbb{R}^n$, where $p > n$, and suppose that $\{v_1, \cdots, v_p\}$ contains $(n + 1)$ affinely independent points. We define the $n$-dimensional polytope

$$P := \mathrm{co} \ \{v_1, \cdots, v_p\}.$$ 

Clearly, a simplex is a special case of a polytope in which $p = n + 1$. Figure 2.3 shows a polytope in $\mathbb{R}^2$. A face of $P$ is any sub-polytope of $P$ which makes up its boundary. The polytope $P$ itself is considered as a trivial face, and all other faces (of dimension less than $n$) are called proper faces. An edge of $P$ is

(i) If $x \in \mathrm{co} \ \{v_1, \cdots, v_k\}$, then $x \in F_j$ for $k + 1 \leq j \leq n$;

(ii) $h_j \cdot (v_i - v_0) = 0$, for all $1 \leq i, j \leq n$ and $j \neq i$;

(iii) $h_j \cdot (v_i - v_k) = 0$, for all $0 \leq i, k \leq n$ and $j \neq i, k$;

(iv) $h_i \cdot (v_i - v_0) < 0$, for all $1 \leq i \leq n$;

(v) $h_j \cdot (v_i - x) > 0$, for all $x \in S \setminus F_j$ and $1 \leq i, j \leq n$, $i \neq j$;

(vi) $h_0 \cdot (v_i - v_0) > 0$, for all $1 \leq i \leq n$;

(vii) The vectors $\{v_1 - v_0, \cdots, v_n - v_0\}$ are a basis for $\mathbb{R}^n$;

(viii) The vectors $\{h_1, \cdots, h_n\}$ are a basis for $\mathbb{R}^n$;

(ix) There exist $\gamma_1 > 0, \cdots, \gamma_n > 0$ such that $h_0 = -\gamma_1 h_1 - \cdots - \gamma_n h_n$.
a 1-dimensional face of \( P \). A facet of \( P \) is an \((n-1)\)-dimensional face of \( P \). We denote the facets of \( P \) by \( F_0, \ldots, F_r \). Let \( h_i \) denote the unit normal vector to \( F_i \) pointing outside \( P \). An implicit description of \( P \) can be obtained using the normal vectors. Precisely, there exist \( \alpha_1, \ldots, \alpha_r \) such that

\[
P = \{ x \in \mathbb{R}^n \mid h_i \cdot x \leq \alpha_i, \forall i \in \{0, \ldots, r\} \}.
\]

In the following part, we review special types of polytopes that we will use in Chapter 3. First, a simplicial polytope is a polytope whose proper faces are simplices. We review generic polytopes. A set of \( p > n \) points in \( \mathbb{R}^n \) are in general position if any \((n+1)\) points of them are affinely independent (form an \( n \)-dimensional simplex). A generic polytope is the convex hull of a set of points in general position in \( \mathbb{R}^n \) [75]. For a generic polytope, all proper faces are simplices. Notice that any generic polytope is simplicial, while the converse is not true. Third, an \( n \)-dimensional polytope \( P \) is said to be simple if each \( k \)-dimensional face of \( P \) is contained in exactly \( n-k \) facets.

**Remark 2.3.1.** If \( P \) is an \( n \)-dimensional simple polytope, then \( P \) has the following properties [74]:

(i) Each vertex of \( P \) is contained in exactly \( n \) edges.

(ii) Let \( F \) be a facet of \( P \) and \( v \) a vertex of \( P \) in \( F \). Then there are exactly \( n-1 \) edges in \( F \) containing \( v \).

### 2.3.3 Triangulation

Triangulation plays a key role in the control synthesis of PWA feedbacks to solve the RCP on polytopes.

**Definition 2.3.1** (\([97]\)). A triangulation \( T \) of an \( n \)-dimensional polytope \( P \) is a finite collection of \( n \)-dimensional simplices \( S_1, \ldots, S_L \) such that (i) \( P = \bigcup_{i=1}^L S_i \); (ii) For all \( i, j \in \{1, \ldots, L\} \) with \( i \neq j \), the intersection \( S_i \cap S_j \) is either empty or a common face of \( S_i \) and \( S_j \).

Let \( P \) be an \( n \)-dimensional polytope, and let \( O \) be an affine space with dimension less than \( n \). Also, suppose \( O_P := P \cap O \) is a polytope with vertices \( V_O = \{o_1, \ldots, o_r\} \). In the following part, we review the placing triangulation, which is used to triangulate \( P \) with respect to \( O \) - that is, to triangulate \( P \) such that \( O_P \) is the union of lower dimensional simplices of the triangulation. In the context of RCP, the placing triangulation was used in [16] to triangulate the polytopic state space with respect to the set of possible equilibria, \( O \).

Next we define the placing triangulation method. To that end, suppose \( V \) is a finite set of points such that \( P = \text{co} (V) \) is a \( k \)-dimensional polytope. A subdivision of \( V \) is a finite collection \( S = \{P_1, \cdots, P_q\} \)
of $k$-dimensional polytopes such that: (i) The vertices of each $P_i$ are drawn from $V$ (though not every point in $V$ need be used), (ii) $P = \bigcup_i P_i$, (iii) If $i \neq j$, then $P_i \cap P_j$ is a common (possibly empty) face of $P_i$ and $P_j$.

**Definition 2.3.2 ([97]).** Let $x \in \mathbb{R}^n$, $P$ an $n$-dimensional polytope, and $F$ a facet of $P$. The hyperplane $H = \text{aff}(F)$ defines an open half-space containing $P^\circ$, the interior of $P$. If $x$ is contained in the opposite open half-space, then $F$ is said to be visible from $x$.

For instance, in Figure 2.4 the facet $F$ is visible from $x_1$, but not visible from $x_2$ or $x_3$. Now we describe what it means to place a vertex. Let $S = \{P_1, \cdots, P_q\}$ be a subdivision of $V$ and $v \in \mathbb{R}^n$ be such that $v \notin V$. Placing the vertex $v$ means that $v$ is adjoined to the point set $V$ after which the following subdivision of $V \cup \{v\}$ is computed.

**Definition 2.3.3 ([97]).** The subdivision $T$ of $V \cup \{v\}$ that results from placing $v$ is obtained as

1. If $v \notin \text{aff}(V)$, then for each $P_i \in S$, include $\text{co}(P_i \cup \{v\})$ in $T$.

2. If $v \in \text{aff}(V)$, then for each $P_i \in S$, $P_i \in T$ and if $F$ is a facet of $P_i$ that is contained in a facet of $\text{co}(V)$ visible from $v$, then $\text{co}(F \cup \{v\}) \in T$.

For example, in Figure 2.5 let $V = \{o_1, o_2\}$ and $S = \{P_1\}$, where $P_1 = \text{co}\{o_1, o_2\}$. The subdivision $T$ of $V \cup \{v_4\}$ that results from placing $v_4$ is obtained as follows. Since $v_4 \notin \text{aff}(V)$, then by Definition 2.3.3(i), $T = \{S_1\}$, where $S_1 = \text{co}\{o_1, o_2, v_4\}$. As another example, let $V = \{o_1, o_2, v_4\}$ and $S = \{P_1\}$, where $P_1 = \text{co}\{o_1, o_2, v_4\}$. The subdivision $T$ of $V \cup \{v_1\}$ that results from placing $v_1$ is obtained as
follows. Since \( v_1 \in \text{aff}(V) \), then by Definition 2.3.3(ii), we have \( P_1 = \text{co}\{o_1, o_2, v_4\} \in T \). Also, since \( \text{co}\{o_1, o_2\} \) is a facet of \( P_1 = \text{co}(V) \) visible from \( v_1 \), we include \( \text{co}\{o_1, o_2, v_1\} \) in \( T \). We conclude that \( T = \{S_1, S_2\} \), where \( S_1 = \text{co}\{o_1, o_2, v_4\} \) and \( S_2 = \text{co}\{o_1, o_2, v_1\} \).

**Theorem 2.3.2** ([97]). Suppose \( V_\mathcal{O} \) and \( V \) are finite sets of points such that \( V_\mathcal{O} \subset V \) and \( P := \text{co}(V) \) is an \( n \)-dimensional polytope. Let \( \mathcal{O}_P := \text{co}(V_\mathcal{O}) \). If the points of \( V \) are ordered such that the points of \( V_\mathcal{O} \) are listed first and if \( T \) is the subdivision obtained by placing the points of \( V \) in order, then \( T \) is a triangulation of \( V \) such that for every \( n \)-dimensional simplex \( S \in T \), \( S^c \cap \mathcal{O}_P = \emptyset \) and if \( S \cap \mathcal{O}_P \neq \emptyset \), then \( S \cap \mathcal{O}_P \) is a face of \( S \).

Figure 2.5 shows a triangulation of \( P \) with respect to \( \mathcal{O} \) that results from placing the points of the set \( V = \{o_1, o_2, v_4, v_3, v_2\} \) in order.

### 2.3.4 Nonlinear Control Theory

We use some notions of nonlinear control theory to prove results on motion primitives for integrator systems in Chapter 7.

**Definition 2.3.4.** Given the system \( \dot{x}(t) = f(x(t)) \), the set \( \mathcal{P} \) is said to be positively invariant if for all \( x(0) \in \mathcal{P} \) the solution \( x(t) \in \mathcal{P} \) for all \( t \geq 0 \).
The Bouligand tangent cone (or simply tangent cone) to $S$ at $x$, denoted $T_S(x)$, is defined by

$$T_S(x) := \left\{ v \in \mathbb{R}^n \mid \liminf_{t \to 0^+} \frac{d_S(x + tv)}{t} = 0 \right\}.$$ 

**Theorem 2.3.3.** [82] Suppose $f \in C^1$. A closed set $\Omega \subset \mathbb{R}^n$ is positively invariant for (7.1) if and only if for all $x \in \partial \Omega$, $f_n(x, u) \in T_\Omega(x)$. 


Chapter 3

Reach Control Problem

This chapter presents the reach control problem on simplices and polytopes. The problem is for trajectories of an affine system defined on a simplex to exit a prespecified facet of the simplex in finite time without first leaving the simplex. In this chapter we review basic principles which shape the features of the problem. These principles are derived from the geometry of the simplex and from convexity properties of affine systems. Most of the proofs in this chapter are suppressed since these results have appeared in previous theses or papers. Those proofs that are included will be highlighted or referenced in later chapters.

3.1 Reach Control Problem

We study an $n$-dimensional simplex defined by

$$
\mathcal{S} := \text{co} \{v_0, \ldots, v_n\}
$$

with vertices $\{v_0, \ldots, v_n \mid v_i \in \mathbb{R}^n\}$ that are affinely independent. Define the vertex set

$$
V := \{v_0, \ldots, v_n\}.
$$

We denote the $(n - 1)$-dimensional facets by $\mathcal{F}_0, \ldots, \mathcal{F}_n$, where the index of each facet is determined by the vertex it does not contain. Let $h_j, j = 0, \ldots, n$ be the unit normal vector to each facet $\mathcal{F}_j$ pointing outside of the simplex. Facet $\mathcal{F}_0$ is called the exit facet of $\mathcal{S}$. Define the index set

$$
I := \{1, \ldots, n\}.
$$
For $x \in \mathcal{S}$ defined the closed, convex cone 

$$\mathcal{C}(x) := \{ y \in \mathbb{R}^n \mid h_j \cdot y \leq 0, j \in I, x \in \mathcal{F}_j \}.$$ 

We introduce a more evocative notation 

$$\text{cone}(\mathcal{S}) := \mathcal{C}(v_0)$$ 

because $\mathcal{C}(v_0)$ is the tangent cone to $\mathcal{S}$ at $v_0$. Instead $\mathcal{C}(v_i)$ for $i \in I$ are not tangent cones to $\mathcal{S}$.

Consider the affine control system defined on $\mathcal{S}$:

$$\dot{x} = Ax + a + Bu, \quad x \in \mathcal{S}, \quad (3.1)$$

where $A \in \mathbb{R}^{n \times n}$, $a \in \mathbb{R}^n$, $B \in \mathbb{R}^{n \times m}$, and rank$(B) = m$. Let $B = \text{Im} (B)$, the image of $B$. Define 

$$\mathcal{O} := \{ x \in \mathbb{R}^n \mid Ax + a \in B \}$$ 

and 

$$\mathcal{O}_P := \mathcal{S} \cap \mathcal{O}.$$ 

(3.3)

We also associate with $\mathcal{O}_P$ its vertex index set 

$$I_{\mathcal{O}_P} := \{ i : v_i \in V \cap \mathcal{O}_P \}.$$
Chapter 3. Reach Control Problem

Let \( \phi_u(t, x_0) \) denote the trajectory of (3.1) starting at \( x_0 \) under a control \( u \).

**Example 3.1.1.** Consider Figure 3.1 where we illustrate the notation in a 2D example. We have a full-dimensional simplex in \( \mathbb{R}^2 \) given by \( S = \text{co} \{ v_0, v_1, v_2 \} \) with vertex set \( V = \{ v_0, v_1, v_2 \} \) and \((n-1)\)-dimensional facets \( F_0, F_1, \) and \( F_2 \). Each facet \( F_j \) has an outward normal vector \( h_j \). The only vertex not in facet \( F_j \) is vertex \( v_j \). \( F_0 \) is the exit facet. If we assume that \( v_0 = 0 \), then subspace \( B \) is shown passing through \( v_0 \). The set \( O \) is an affine space shown passing through \( F_0 \). Notice in this case \( O \cap \{ v_1, v_2 \} \).

The problem we consider is one where closed-loop trajectories of (3.1) are driven out of \( S \) through the exit facet \( F_0 \) only. For this, conditions are required that restrict trajectories from exiting the remaining facets \( F_i, i \in I \). In particular, it is said that the invariance conditions are solvable at vertex \( v_i \in V \) if there exists \( u_i \in \mathbb{R}^m \) such that

\[
Av_i + Bu_i + a \in C(v_i) .
\]  

Moreover, the invariance conditions are solvable if (3.4) is solvable at each \( v_i \in V \). The inequalities (3.4) are called invariance conditions. They guarantee trajectories cannot exit from the facets \( F_i, i \in I \), and they are used to construct affine feedbacks [6]. For general state feedbacks, stronger conditions (also called invariance conditions) are needed. A state feedback \( u = f(x) \) then satisfies the invariance conditions if for all \( x \in \mathcal{S} \),

\[
Ax + Bf(x) + a \in \mathcal{C}(x) .
\]  

**Example 3.1.2.** Consider Figure 3.2. Attached at each vertex is a velocity vector \( y_i := Av_i + Bu_i + a \), \( i \in \{ 0 \} \cup I \). The invariance conditions (3.4) require that \( y_i \in \mathcal{C}(v_i) \), as illustrated. Notice that velocity vectors at \( v_i \in F_0 \) may or may not point out of \( S \). If the control is an affine feedback \( u = Kv_i + g \) such that \( u_i = Kv_i + g \), then by convexity of the closed-loop vector field, (3.5) holds at every \( x \in F_i, i \in I \). If the input is a continuous state feedback \( u = f(x) \), then invariance conditions for every \( x \in F_i, i \in I \), must be explicitly stated, since convexity is not guaranteed; hence (3.5).

**Problem 3.1.1 (Reach Control Problem (RCP)).** Consider system (3.1) defined on \( S \). Find a state
feedback $u = f(x)$ such that for every $x \in \mathcal{S}$ there exist $T \geq 0$ and $\gamma > 0$ such that $\phi_u(t, x) \in \mathcal{S}$ for all $t \in [0, T]$, $\phi_u(T, x) \in \mathcal{F}_0$, and $\phi_u(t, x) \notin \mathcal{S}$ for all $t \in (T, T + \gamma)$.

In the sequel we will use the shorthand notation $\mathcal{S} \xrightarrow{\mathcal{S}} \mathcal{F}_0$ to denote that (i)-(iii) of Problem 3.1.1 hold under some control law.

We now focus our attention on the set $\mathcal{O}$, which is critical in RCP. The most important observation
is that the vector field $Ax + a + Bu$ can vanish at any $x \in \mathcal{O}$ for an appropriate choice of $u \in \mathbb{R}^m$, so $\mathcal{O}$ is interpreted as the set of all possible equilibrium points of (3.1). Thus, if $x_0$ is an equilibrium of (3.1) under feedback control, then $x_0 \in \mathcal{O}$. Similarly the set of possible equilibrium points of (3.1) on $\mathcal{S}$ is given by $\mathcal{O}_P := \mathcal{S} \cap \mathcal{O}$.

The following lemma provides a characterization of $\mathcal{O}$.

**Lemma 3.1.1.**

(i) If $\text{Im } (A) \subseteq \mathcal{B}$ and $a \not\in \mathcal{B}$, then $\mathcal{O} = \emptyset$;

(ii) If $\text{Im } (A) \subseteq \mathcal{B}$ and $a \in \mathcal{B}$, then $\mathcal{O} = \mathbb{R}^n$;

(iii) Otherwise, $\mathcal{O}$ is an affine space with $m \leq \text{dim}(\mathcal{O}) < n$.

**Example 3.1.3.** We conclude this section by giving a second example of the geometric constructs required for RCP for the the case $n = 3$ and $m = 2$. Consider Figure 3.3. We have a simplex $\mathcal{S}$ with normal vectors $h_i$ to each facet $\mathcal{F}_i$. Depicted by a shaded section is cone($\mathcal{S}$), the tangent cone at $v_0$. The space $\mathcal{B}$ is copied to $v_0$, and in this view we see that $\mathcal{B} \cap \text{cone}(\mathcal{S}) = 0$. That is, $\mathcal{B}$ does not “dip” into the tangent cone at $v_0$. This geometric feature will be further analyzed in ensuing chapters. The affine set $\mathcal{O}$ intersects $\mathcal{S}$ along the face $\mathcal{F}_1\mathcal{F}_2$, and this forms $\mathcal{O}_P$. It is interpreted as the set of possible equilibria of the system. We know that in $\mathcal{O}_P$, the only velocity vectors available to the closed loop system are vectors in $\mathcal{B}$. This is depicted by placing copies of $\mathcal{B}$ at each of the vertices of $\mathcal{O}_P$. Two velocity vectors $b_1$ and $b_2$ are shown, and these clearly satisfy the invariance conditions at $v_1$ and $v_2$, respectively. At vertices not in $\mathcal{O}_P$, the drift term $Ax + a$ becomes relevant, and the figure depicts closed-loop velocity vectors at $v_0, v_3 \not\in \mathcal{O}_P$ which satisfy their respective invariance conditions. The invariance conditions can be interpreted in terms of the cones $\mathcal{C} (v_i)$. Consider vertex $v_3$ where $\mathcal{C} (v_3)$ is depicted by a shaded region. This cone is shaped like an open book whose spine is parallel to the face $\mathcal{F}_1\mathcal{F}_2$ and whose cover and back cover lie in $\mathcal{F}_2$ and $\mathcal{F}_1$, respectively. The invariance condition at $v_3$ is satisfied if the closed-loop velocity vector $Av_3 + Bu_3 + a$ lies in $\mathcal{C} (v_3)$.

### 3.2 Affine Systems

In this section well-known properties of affine systems on compact, convex sets are presented for the case where there is no control input. First, results regarding the relationship between positively invariant sets, existence of equilibria, and trajectories leaving the set in finite time are discussed. Second, the way in which trajectories exit a compact, convex set when invariance conditions such as (3.4) hold is shown.
Proofs in this section are also not included because the results follow from well-known arguments in convex analysis.

Let \( P \subset \mathbb{R}^n \) be a compact, convex set and consider the affine system defined on \( P \):

\[
\dot{x} = Ax + a, \quad x \in P, \quad (3.6)
\]

where \( A \in \mathbb{R}^{n \times n} \) and \( a \in \mathbb{R}^n \). Let \( \phi(t, x_0) \) denote the unique trajectory of (3.6) starting from \( x_0 \in P \).

**Lemma 3.2.1.** Consider the affine system (3.6) defined on a compact, convex set \( P \). Suppose there exists \( x_0 \in P \) such that \( \phi(t, x_0) \in P \) for all \( t \geq 0 \). Then the set

\[
\Phi := \text{co} \{ \phi(t, x_0) \mid t \geq 0 \} \subset P
\]

is a positively invariant set. Moreover, \( \Phi \) contains an equilibrium of (3.6), i.e. there exists \( \overline{x} \in \Phi \) such that \( A\overline{x} + a = 0 \).

**Lemma 3.2.2.** Consider the affine system (3.6) defined on a compact, convex set \( P \). We have \( Ax + a \neq 0 \) for all \( x \in P \) if and only if there exists \( \xi \in \mathbb{R}^n \) such that

\[
\xi \cdot (Ax + a) < 0, \quad x \in P. \quad (3.7)
\]

Equation (3.7) is called a *flow condition* for system (3.6) on \( P \). A consequence of the existence of a flow condition on a compact, convex set is that all trajectories of (3.6) originating in the set eventually leave it.

**Lemma 3.2.3.** Consider the affine system (3.6) defined on a compact, convex set \( P \). Suppose that for all \( x \in P \), \( Ax + a \neq 0 \). Then for each \( x_0 \in P \), the trajectory starting at \( x_0 \) eventually leaves \( P \), i.e. there exists \( t_1 > 0 \) such that \( \phi(t_1, x_0) \notin P \).

The existence of a flow condition is also related to the set \( \mathcal{O} \) for an affine control system. To see this, consider again the control system (3.1). The next result shows that a flow condition naturally arises on any compact, convex set that does not intersect \( \mathcal{O} \). Importantly, the control input plays no role and, moreover, the origin of the vector \( \xi \) can be made explicit.

**Lemma 3.2.4.** Consider the affine control system (3.1) defined on a compact, convex set \( P \). If \( P \cap \mathcal{O} = \emptyset \), then there exists \( \xi \in \text{Ker} \ B^T \) such that

\[
\xi \cdot (Ax + a) < 0, \quad \forall x \in P.
\]
Lemma 3.2.4 allows us to return to Lemma 3.2.1 and make a statement about trajectories that do not exit $\mathcal{P}$ under an open-loop control for the control system (3.1). Informally, if a trajectory of an affine control system does not leave $\mathcal{P}$, it is because it encircles $\mathcal{P} \cap \mathcal{O}$, approaches $\mathcal{P} \cap \mathcal{O}$, or remains on $\mathcal{P} \cap \mathcal{O}$.

**Lemma 3.2.5.** Consider the affine control system (3.1) defined on a compact, convex set $\mathcal{P}$. Suppose there exists $x_0 \in \mathcal{P}$ and an open-loop control $\mu(t)$ such that the (unique) solution $\phi_{\mu}(t, x_0)$ satisfies $\phi_{\mu}(t, x_0) \in \mathcal{P}$ for all $t \geq 0$. Then the set

$$\Phi := \text{co} \{ \phi(t, x_0) \mid t \geq 0 \} \subset \mathcal{P}$$

satisfies

$$\Phi \cap \mathcal{O} \neq \emptyset.$$ 

Finally, a result is presented that examines the way that trajectories of an affine system exit a compact, convex set when invariance conditions such as (3.4) or (3.5) hold.

**Lemma 3.2.6.** Consider the affine system (3.6) defined on a compact, convex set $\mathcal{P}$. Suppose additionally that $\mathcal{P}$ is a polytope with facets $\{ \mathcal{F}_0, \ldots, \mathcal{F}_k \}$. Let $h_i$ be the outward normal vector of $\mathcal{F}_i$. Suppose that for some facet $\mathcal{F}_i$ the following conditions hold:

$$h_i \cdot (Ax + a) \leq 0, \quad \forall x \in \mathcal{F}_i. \quad (3.8)$$

Then all trajectories originating in $\mathcal{P}$ that leave $\mathcal{P}$ do so via a facet $\mathcal{F}_j$, $j \neq i$.

The next result shows that for affine systems, if trajectories exit from the proper exit facet, then the invariance conditions hold.

**Lemma 3.2.7.** Consider the affine system (3.6) defined on a compact, convex set $\mathcal{P}$. Suppose additionally that $\mathcal{P}$ is a polytope with facets $\{ \mathcal{F}_0, \ldots, \mathcal{F}_k \}$. Let $h_i$ be the outward normal vector of $\mathcal{F}_i$. Suppose that $\mathcal{P} \overset{\rho}{\rightarrow} \mathcal{F}_0$. Then

$$h_i \cdot (Ax + a) \leq 0, \quad \forall x \in \mathcal{F}_i, i = 1, \ldots, k. \quad (3.9)$$

### 3.3 Affine Feedback

In this section we present results from the literature [6]-[9], in which the properties of affine systems given in the previous section are used to produce a solution of RCP by affine feedback. The solution
relies on a control synthesis procedure described next.

**Lemma 3.3.1** ([6]). Consider two sets of points $V = \{v_0, \ldots, v_n \mid v_i \in \mathbb{R}^n \}$ and $\{u_0, \ldots, u_n \mid u_i \in \mathbb{R}^m \}$. Suppose $V$ is affinely independent. Then there exist unique matrices $K \in \mathbb{R}^{m \times n}$ and $g \in \mathbb{R}^m$ such that

$$u_i = Kv_i + g, \quad i \in \{0, \ldots, n\}.$$ 

**Proof.** We must show there exist matrices $K$ and $g$ such that

$$
\begin{bmatrix}
v_0^T & 1 \\
\vdots & \vdots \\
v_n^T & 1
\end{bmatrix}
\begin{bmatrix}
K^T \\
g^T
\end{bmatrix}
= 
\begin{bmatrix}
u_0^T \\
\vdots \\
u_n^T
\end{bmatrix}.

(3.10)
$$

If the $(n + 1) \times (n + 1)$ left-hand matrix is full rank, then multiplying by its inverse yields the unique solutions $K$ and $g$. However,

$$
\text{rank}
\begin{bmatrix}
v_0^T & 1 \\
\vdots & \vdots \\
v_n^T & 1
\end{bmatrix}
= 1 + \text{rank}
\begin{bmatrix}
v_1^T - v_0^T \\
v_2^T - v_0^T \\
\vdots \\
v_n^T - v_0^T
\end{bmatrix}
= 1 + n.
$$

The last equality follows since the points $\{v_0, \ldots, v_n\}$ are affinely independent if and only if $\{v_1 - v_0, \ldots, v_n - v_0\}$ are linearly independent. 

Using the results in Section 3.2 and the synthesis procedure of Lemma 3.3.1, the necessary and sufficient conditions for solvability of RCP by affine feedback are given in the following theorem.

**Theorem 3.3.1** ([7, 9]). Given the system (3.1) and an affine feedback $u(x) = Kx + g$, where $K \in \mathbb{R}^{m \times n}$, $g \in \mathbb{R}^m$, and $u_0 = u(v_0), \ldots, u_n = u(v_n)$, the closed-loop system satisfies $\mathcal{S} \xrightarrow{S} \mathcal{F}_0$ if and only if

(a) The invariance conditions (3.4) hold.

(b) There is no equilibrium in $\mathcal{S}$.

**Proof.**

$(\Rightarrow)$ If $\mathcal{S} \xrightarrow{S} \mathcal{F}_0$ by affine feedback, then clearly the closed-loop system does not have equilibria in $\mathcal{S}$, for otherwise trajectories starting at an equilibrium would not leave $\mathcal{S}$. The invariance conditions (3.4) hold by Lemma 3.2.7.
(⇐) By assumption, for vertex set $V$ there exist inputs $\{u_0, \ldots, u_n\}$ satisfying the invariance conditions (3.4). Invoking Lemma 3.3.1, there exists an affine feedback $u = Kx + g$ such that the invariance conditions are satisfied at the vertices. The resulting closed-loop system is

$$\dot{x} = (A + BK)x + (Bg + a) = \tilde{A}x + \tilde{a}.$$ 

By assumption, $\tilde{A}x + \tilde{a} \neq 0$ for all $x \in S$, so by Lemma 3.2.3, all trajectories leave $S$ in finite time. From (3.4),

$$h_j \cdot (\tilde{A}v_i + \tilde{a}) \leq 0, \quad i \in \{0, \ldots, n\}, \quad j \in I_i.$$ 

By convexity

$$h_i \cdot (\tilde{A}x + \tilde{a}) \leq 0, \quad \forall x \in F_i, \quad i \in I.$$ 

By Lemma 3.2.6 trajectories cannot leave $S$ via $F_1, \ldots, F_n$. This proves condition (i) of RCP. For condition (ii), since $\|\tilde{A}x + \tilde{a}\| \neq 0$ for all $x \in S$, $S$ is compact, and $x \mapsto \|\tilde{A}x + \tilde{a}\|$ is continuous, there exists $\varepsilon > 0$ such that $\|\tilde{A}x + \tilde{a}\| > \varepsilon$ for all $x \in S$. 

Theorem 3.3.1 gives conditions for the solvability of RCP which are primarily of theoretical interest. In practice, these conditions do not realize a synthesis procedure, for one cannot guarantee that a proposed affine feedback satisfying the invariance conditions does not place closed-loop equilibria in $S$. On the other hand, existence of a flow condition guarantees there are no equilibria in $S$ by Lemma 3.2.2, so one can replace the requirement of no closed-loop equilibria with existence of a flow condition. This observation leads to an alternative set of necessary and sufficient conditions for solvability of RCP.

**Theorem 3.3.2** ([9]). $S \xrightarrow{\xi} S_0$ by affine feedback if and only if there exist $u_0, \ldots, u_n \in \mathbb{R}^m$ and $\xi \in \mathbb{R}^n$ such that

(a) The invariance conditions (3.4) hold.

(b) The flow condition holds: $\xi \cdot (Av_i + Bu_i + a) < 0, \quad i \in \{0, \ldots, n\}.$

Theorem 3.3.2 can be viewed as a computational solution to the problem. If the invariance and flow conditions can be solved simultaneously for the unknowns $\xi \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^m$, then an affine feedback can be constructed by the procedure of Lemma 3.3.1. Unfortunately, this approach relies on solving bilinear inequalities, which are known to be $NP$-hard [113].
We now present results for a variation of Problem 3.1.1 whereby it is desired to keep trajectories within a simplex \( S \) for all time, and to stabilize the system to a point \( \bar{x} \in S \). That is, a state feedback \( u = f(x) \) is to be found such that:

(i) For every \( x \in S \) then \( \varphi_u(t, x) \in S, t \geq 0 \).

(ii) Given \( \bar{x} \in S \), solve (i) and achieve \( \varphi_u(t, x) \to \bar{x} \) as \( t \to \infty \).

The solution to (i) amounts to the solvability of the invariance conditions where \( I := \{0, \ldots, n\} \) such that there is no exit facet.

**Theorem 3.3.3 ([7])**. For every \( x \in S \), \( \varphi_u(t, x) \in S, t \geq 0 \) by affine feedback if and only if there exist \( u_0, \ldots, u_n \in \mathbb{R}^m \) such that \( \forall i, j \in \{0, \ldots, n\}, i \neq j : h_j^T \cdot (Av_i + Bu_i + a) \leq 0 \).

The solution to (ii) additionally requires the geometric condition that equilibria may exist in \( S \), i.e. that \( O_P = S \cap O \neq \emptyset \), and that \( \bar{x} \) lies on the set of possible equilibria. Moreover the closed-loop velocities at the vertices of \( S \) must span \( \mathbb{R}^n \).

**Theorem 3.3.4 ([7])**. Let \( \bar{x} \in S \). Condition (i) is solved and \( \varphi_u(t, x) \to \bar{x} \) as \( t \to \infty \) by affine feedback if and only if there exist \( u_0, \ldots, u_n \in \mathbb{R}^m \) such that

(i) \( \forall i, j \in \{0, \ldots, n\}, i \neq j : h_j^T \cdot (Av_i + Bu_i + a) \leq 0 \)

(ii) \( \bar{x} \in O \)

(iii) \( \text{span}(\{Av_i + Bu_i + a| i = 0, \ldots, n\}) = \mathbb{R}^n \).

### 3.4 Triangulation and Affine Feedback

In this section we assume a specific triangulation of the state space which makes possible the use of various results in the literature for such triangulations [16]. Specifically, the selected triangulation is attractive because it yields more geometric necessary and sufficient conditions for solvability of RCP. Note that a similar discussion of the material to come was presented in [65].

We retain our declaration of a simplex \( S = \text{co} \{v_0, v_1, \ldots, v_n\} \) and the affine system defined on \( S \)

\[
\dot{x} = Ax + Bu + a, \quad x \in S, \quad (3.13)
\]

where \( A \in \mathbb{R}^{n \times n}, a \in \mathbb{R}^n, B \in \mathbb{R}^{n \times m}, \) and \( \text{rank}(B) = m. \)
We present first the geometric sufficient conditions for existence of affine feedbacks solving RCP with respect to the sets $\mathcal{O}$ and $\mathcal{O}_P$, defined in (3.2) and (3.3), respectively. We remark that the invariance conditions by themselves are generally not enough to produce a solution to RCP by affine feedback, as Theorem 3.3.1 indicates. However, there is one extreme case when the invariance conditions sufficiently solve the problem. This idea depends on combining Theorem 3.3.1 with the notion that equilibria may only appear on the set $\mathcal{O}$.

**Theorem 3.4.1** ([16]). Suppose $\mathcal{O}_P = \emptyset$. If the invariance conditions are solvable, then $\mathcal{S} \xrightarrow{S} \mathcal{F}_0$ by affine feedback.

In general it is difficult to extend results such as Theorem 3.4.1. Instead, a preferred triangulation of the state space is used, presented in Section 2.3.3, which expedites the problem of obtaining geometric necessary and sufficient conditions for a solution to RCP. Imperative to the approach is the following assumption.

**Assumption 3.4.1.** Simplex $\mathcal{S}$ and system (3.13) satisfy the following condition: if $\mathcal{O}_P \neq \emptyset$, then $\mathcal{O}_P$ is a $\kappa$-dimensional face of $\mathcal{S}$, where $0 \leq \kappa \leq n$.

**Remark 3.4.2.** There are three possible forms for $\mathcal{O}$, and hence three possible approaches to a solution. If $\mathcal{O} = \emptyset$, then one applies Theorem 3.4.1. If $\mathcal{O}$ is the entire state space then it will be shown that there are easily derived necessary and sufficient conditions for solvability. The only interesting case is when $\mathcal{O}$ is a $\kappa$-dimensional affine subspace with $\kappa < n$. This case arises, for example, when $(A, B)$ is controllable, and then the placing triangulation can be applied.

With this result, several new sufficient conditions for existence of an affine feedback that will solve RCP have been shown to exist, which are presented below. In particular, the preferred triangulation of the state space with respect to $\mathcal{O}$ has been exploited to obtain such results. In particular, application of Lemma 3.2.4, along with additional conditions, has been found to achieve on $\mathcal{S}$ a flow condition satisfying Theorem 3.3.2.

**Theorem 3.4.3** ([16]). Suppose Assumption 3.4.1 holds and $\mathcal{O}_P \neq \emptyset$. Suppose the following conditions hold.

1. The invariance conditions (3.4) are solvable.

2. $\mathcal{B} \cap \text{cone} (\mathcal{S}) \neq \emptyset$.

Then $\mathcal{S} \xrightarrow{S} \mathcal{F}_0$ by affine feedback.
Sufficient conditions for the existence of affine feedback are also feasible when \( B \cap \text{cone}(S) = 0 \); this may only occur if \( v_0 \not\in O_P \). This relies on the idea that there are enough degrees of freedom in \( B \) with respect to \( O_P \). Consider then the following assumptions.

**Assumption 3.4.2.**

(A1) \( O_P = \text{co} \{v_1, \ldots, v_{\kappa+1}\} \), with \( 0 \leq \kappa < m \).

(A2) \( B \cap \text{cone}(S) = 0 \).

(A3) There exists a linearly independent set \( \{b_i \in B \cap C_i \mid i \in I_{O_P}\} \).

The important new assumption is (A3) which says that \( B \) and \( O_P \) are arranged with respect to each other so that enough degrees of freedom exist in \( B \) both to span a \( \kappa + 1 \)-dimensional subspace of \( B \) and at the same time satisfy all the invariance conditions for the vertices of \( O_P \), indexed by \( I_{O_P} \). To achieve this we require \( \kappa < m \).

![Figure 3.4: Continuous Assignment Along \( O_P \) of Vectors in \( B \).](image_url)

**Theorem 3.4.4** ([16]). *Suppose Assumption 3.4.1 holds and \( O_P = \text{co} \{v_1, \ldots, v_{\kappa+1}\} \), with \( 0 \leq \kappa < m \). Suppose the following conditions hold.*

1. The invariance conditions (3.4) are solvable.

2. There exists a linearly independent set \( \{b_i \in B \cap C_i \mid i \in I_{O_P}\} \).

Then \( S \rightarrow F_0 \) by affine feedback.
Example 3.4.1. Consider the 2D simplex $\mathcal{S} = \text{co} \{v_0, v_1, v_2\}$ with $v_0 = (0, 0)$, $v_1 = (1, 2)$ and $v_2 = (3, 2)$. Consider the affine system on $\mathcal{S}$

$$\dot{\xi} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 4 \\ 2 \end{bmatrix}.$$ 

The normal vectors of the facets of $\mathcal{S}$ are $h_0 = (0, 0.5)$, $h_1 = (0.5, -0.75)$ and $h_2 = (-0.5, 0.25)$. The objective is to solve the reach control problem on $\mathcal{S}$ with respect to exit facet $\mathcal{F}_0$. First, we calculate the set of possible equilibria $\mathcal{O} = \{\xi \in \mathbb{R}^2 \mid \xi_1 - \xi_2 - 4 = 0\}$. Then, we compute $\mathcal{O}_\mathcal{F} = \mathcal{S} \cap \mathcal{O} = \emptyset$. Since no equilibria can exist on $\mathcal{S}$, then by Theorem 3.4.1, if the invariance conditions are solvable, then $\mathcal{S} \rightarrow \mathcal{F}_0$ by affine feedback. To that end, let $y_i := Av_i + Bu_i + a$, $i = 0, 1, 2$. From (3.4) we have the following invariance conditions for $\mathcal{S}$:

- $v_0: \ h_1 \cdot y_0 \leq 0 \implies (0.5, -0.75) \cdot (4, 2 + u_0) \leq 0 \implies \frac{2}{3} \leq u_0$
- $h_2 \cdot y_0 \leq 0 \implies (-0.5, 0.25) \cdot (4, 2 + u_0) \leq 0 \implies u_0 \leq 6$
- $v_1: \ h_2 \cdot y_1 \leq 0 \implies (-0.5, 0.25) \cdot (5, 3 + u_1) \leq 0 \implies u_1 \leq 7$
- $v_2: \ h_1 \cdot y_2 \leq 0 \implies (0.5, -0.75) \cdot (3, 5 + u_2) \leq 0 \implies -3 \leq u_2$.

We choose $u_0 = 4$, $u_1 = 2$ and $u_2 = -1$ as the control values, which satisfy the inequalities above. With the chosen controls and the $v_i$ we generate the feedback parameters $K$ and $g$ using (3.10) and construct

$$u = [-1.5 - 0.25] \xi + 4, \quad \xi \in \mathcal{S}.$$ 

In Figure 3.5 the simplex $\mathcal{S}$ and the closed-loop vector field under $u$ are depicted. Clearly, for any initial condition in $\mathcal{S}$ trajectories exit through $\mathcal{F}_0$ without exiting other facets.

### 3.5 Reach Control on Polytopes

In this thesis there will prove to be scenarios in which a solution to RCP using simplex methods fails to materialise, due to the restrictive nature of the simplex. In particular, the inherent structure of the system may render it difficult to guarantee that the invariance conditions hold in order to allow trajectories to pass through some facets but not others. In this instance we turn to polytopes, which are effectively a collection of simplices, for which a desired system behaviour may be achieved using a similar
Chapter 3. Reach Control Problem

Figure 3.5: Closed-loop vector field on $S$ for Example 3.4.1.

design methodology. The material to come skims the surface of the problem of RCP on polytopes which is dealt with in full in [89]. The reader is referred to [89] for an illustration of the proofs for the cited results, as well as an extensive comparison of simplex and polytope solutions.

Consider an $n$-dimensional polytope

$$P := \text{co} \{ v_1, \ldots, v_p \}$$

with vertex set $V := \{ v_1, \ldots, v_p \}$ and facets $F_0, F_1, \ldots, F_r$. Denote $F_0$ the exit facet of $P$, and $F_1, \ldots, F_r$ the restricted facets of $P$. Let $h_j$ be the outward-pointing unit normal vector to each facet $F_i$. Define the index sets $I_P := \{ 1, \ldots, p \}$, $J := \{ 1, \ldots, r \}$, and $J(x) := \{ j \in J \mid x \in F_j \}$. That is, $J(x)$ is the set of indices of the restricted facets in which $x$ is a point. As in Section 3.1, for each $x \in P$, define the closed, convex cone

$$C(x) := \{ y \in \mathbb{R}^n \mid h_j \cdot y \leq 0, \ j \in J(x) \}.$$

Note that $F_0$ never appears in this definition of $C(x)$ since $F_0$ is the exit facet. We consider the affine control system defined on $P$:

$$\dot{x} = Ax + Bu + a, \quad x \in P,$$

where $A \in \mathbb{R}^{n \times n}$, $a \in \mathbb{R}^n$, $B \in \mathbb{R}^{n \times m}$, and $\text{rank}(B) = m$. Let $\phi_u(t, x_0)$ be the trajectory of (3.14) under a control law $u$ starting from $x_0 \in P$, and let $B = \text{Im}(B)$, the image of $B$. We are interested in studying reachability of the exit facet $F_0$ from $P$ by feedback control.
Also as in Section 3.1 define the set of possible equilibrium points of (3.1) on \( P \) by

\[
O_P := P \cap O,
\]

(3.15)

where \( O \) is defined in (3.2). Since \( O \) is an affine space, either \( O_P = \emptyset \) or \( O_P \) is a \( \kappa \)-dimensional polytope in \( P \). If \( O_P \neq \emptyset \), we define the vertex set of \( O_P \) to be \( V_O := \{o_1, \ldots, o_q\} \), where \( o_i \) are the vertices of \( O_P \) (not necessarily vertices of \( P \)). Also define the index set \( I_O := \{1, \ldots, q\} \). We say \( T \) is a triangulation of \( P \) with respect to \( O \) if \( T \) is a triangulation of \( P \) such that \( O_P \) is a union of simplices of the triangulation. We specify further criteria for the triangulation in order to obtain and harness specific geometric properties that are helpful in developing a solution to RCP. In particular, in Section 3.4 we used the assumption that \( O_P \) is a face of the simplex. We adopt the same type of assumption for polytopes.

**Assumption 3.5.1.** Polytope \( P \) and system (3.14) satisfy the following condition: \( O_P \) is a \( \kappa \)-dimensional face of \( P \), where \( 0 \leq \kappa \leq n \). In particular,

\[
O_P = \text{co} \{v_1, \ldots, v_q\},
\]

where \( v_i \) is a vertex of \( P_i \).

The procedure of triangulating \( P \) in such a manner is detailed in Section 2.3.3. We now repeat the notion of the invariance conditions in the context of polytopes.

**Definition 3.5.1.** We say the invariance conditions are solvable if for each \( v \in V \) there exists \( u \in \mathbb{R}^m \) such that

\[
Av + Bu + a \in C(v).
\]

(3.16a)

Equivalently,

\[
h_j \cdot (Av + Bu + a) \leq 0, \quad j \in J(v).
\]

(3.16b)

Also define the following closed-loop velocity vectors associated with the set \( O_P \):

\[
b_i := Ao_i + Bu(o_i) + a \in B \cap C(o_i), \quad i \in I_O.
\]

(3.17)

We now present a variation of the reach control problem for polytopes which is called the Monotonic Reach Control Problem (MRCP).

**Problem 3.5.1 (Monotonic Reach Control Problem (MRCP)).** Consider system (3.14) defined on a convex
polytope $\mathcal{P}$. Find a state feedback $u(x)$ such that for each initial condition $x_0 \in \mathcal{P}$, there exist $T \geq 0$ and $\gamma > 0$ such that

(i) $\phi_u(t, x_0)$ remains in $\mathcal{P}$ for all time $t \in [0, T]$; $\phi_u(t, x_0)$ reaches $\mathcal{F}_0$ at time $T$; $\phi_u(t, \xi_0)$ leaves $\mathcal{P}$ during an interval of time $t \in (T, T + \gamma)$.

(ii) There exists $\xi \in \mathbb{R}^n$ such that for all $x \in \mathcal{P}$, $\xi \cdot (Ax + Bu(x) + a) < 0$.

Again, the goal is to construct a piecewise affine (PWA) state feedback law that will ensure that any closed-loop trajectory that starts in $\mathcal{P}$ may only exit the polytope through the exit facet $\mathcal{F}_0$. Condition (ii) is new with respect to Problem 3.1.1, and is called a flow condition. Moreover, the term ‘monotonic” refers to the evolution of trajectories through the polytope in a common sense with respect to consecutive, parallel hyperplanes characterized by the normal vector $\xi$. We write $\mathcal{P} \xrightarrow{\text{P}} \mathcal{F}_0$ monotonically where conditions (i)-(ii) from above hold.

In general, numerical algorithms are required to determine the existence of $\xi$, however a set of geometric arguments may be made that connect the flow condition to the existence of equilibria on $\mathcal{P}$. In particular, the first result addresses the case where $\mathcal{O}_\mathcal{P} = \emptyset$, to which we refer the reader to Lemma 3.2.2. In effect, the lack of equilibria on $\mathcal{P}$ implies that trajectories must flow out of $\mathcal{P}$. Consequently, a result analogous to that of Theorem 3.3.1 for simplices follows, where the lack of equilibria simply requires the satisfaction of the invariance conditions for MRCP to be solved.

**Theorem 3.5.1 ([89]).** Consider the system (3.14) defined on a polytope $\mathcal{P}$, and suppose $\mathcal{O}_\mathcal{P} = \emptyset$. Then $\mathcal{P} \xrightarrow{\text{P}} \mathcal{F}_0$ if and only if the invariance conditions (3.16) are solvable.

Commonly, $\mathcal{O}_\mathcal{P} \neq \emptyset$, and thus a trivial argument guaranteeing no equilibria in $\mathcal{P}$ may not be made. However, additional geometric conditions associated with the closed-loop velocities at the vertices $V_\mathcal{O}$ (as a result of an affine feedback) can be used to determine when there are no equilibria in $\mathcal{P}$.

**Theorem 3.5.2 ([89]).** Consider the system (3.14) defined on a polytope $\mathcal{P}$. Let $\mathcal{T}$ be a triangulation of $\mathcal{P}$ with respect to $\mathcal{O}$, $u(x)$ be a piecewise affine feedback defined on $\mathcal{T}$, and $b_i$ be as in (3.17). If $0 \notin \text{co} \{b_1, \ldots, b_q\}$, then the closed-loop system has no equilibrium in $\mathcal{P}$.

The condition above equivalently implies the lack of equilibria on $\mathcal{P}$ and the existence of a flow-like vector $\xi$ from (ii) of Problem 3.5.1. With these results in hand, the following theorem provides the necessary and sufficient conditions for a solution to MRCP as used in this thesis.

**Theorem 3.5.3 ([89]).** Consider the system (3.14) defined on $\mathcal{P}$ and suppose Assumption 3.5.1 holds. Then $\mathcal{P} \xrightarrow{\text{P}} \mathcal{F}_0$ monotonically by continuous piecewise affine feedback if and only if
(i) The invariance conditions (3.16) hold.

(ii) There exists \( \{b_1, \ldots, b_q | b_i \in B \cap C(v_i) \} \) such that \( 0 \notin \text{co} \{b_1, \ldots, b_q\} \).

Finally, the methodology for obtaining an affine feedback that solves MRCP is as follows. For an affine system defined on a polytope we first select control values at the vertices of \( \mathcal{P} \) to satisfy the invariance conditions (3.16). Then the polytope is triangulated into a number of simplices \( \mathcal{S}_k \), \( k > 1 \) such that \( \mathcal{O} \cap \mathcal{S}_k \) is a \( \kappa \)-dimensional face of \( \mathcal{S}_k \), where \( 0 \leq \kappa \leq n \). A continuous affine feedback of the form \( u = Kx + g \) is generated for each \( \mathcal{S}_k \subset \mathcal{P} \) by applying (3.10) with the \( u_i \) and \( v_i \) belonging to \( \mathcal{S}_k \). A hybrid switching system is then constructed on \( \mathcal{P} \) with feedback \( u_{\mathcal{P}}(x) \) defined as

\[
\begin{align*}
 u_{\mathcal{P}}(x) &:= K^k x + g^k, \quad x \in \mathcal{S}_k. 
\end{align*}
\] (3.18)
Chapter 4

Output Regulation for Affine Systems

In this chapter we develop our ideas on the output reach control problem. We propose two methods to solve the RCP in the output space, both of which rely on regulator theory. To properly formulate our two approaches, we first extend classic regulator theory in two directions. First we consider affine control systems, rather than linear systems. Second, we consider exosystems that are stable, not only marginally stable as is typical in the literature.

In the first method to solve the RCP in the output space, we model both the desired reference behavior and persistent unmeasurable disturbances using two exosystems. Then the problem is to track one of the reference trajectories generated by the first exosystem, while rejecting the disturbance modeled by the second exosystem.

In the second method, we assume we have available a family of reach controllers defined on simplices in the state space to yield desirable behavior in the output. Then the problem is one of pure disturbance rejection - to reject a disturbance entering the output of the system.

The methods of this chapter make no claim that trajectories remain in a given polytope or simplex. Therefore, these methods offer no guarantees that the RCP is solved on any individual simplex. In practice, there may be initial conditions such that a disturbance is rejected in sufficient time and the requirements of the RCP are in fact solved for these trajectories. A rigorous theory of disturbance rejection for the RCP could be formulated, by placing bounds on the size of disturbances; this extension is beyond the scope of this thesis. Instead, we found a more fruitful area of research to be the RCP in the output space, as it had the potential to address dimensionality problems with the RCP.
4.1 Affine Regulator Theory

We develop an extension of classic regulator theory to the case of affine systems. Strictly speaking affine systems are nonlinear systems, so one could in principle use nonlinear regulator theory to obtain these results [13]. We are interested in explicitly deriving regulator theory for affine systems, since the additional structure of affine systems over general nonlinear systems is helpful in the design problems we treat in Chapter 5. See also Remark 4.1.3 after Theorem 4.1.1. Our extension includes exosystems with stable dynamics; thus further motivating a redevelopment of regulator theory for the case affine systems.

We first provide some preliminary definitions and results which will be used later in proving our main results. Particularly, we extend what is meant by stabilizability and detectability for a shifted complex half plane. This allows us to extend current results in regulator theory to stable exosystems. Let $\alpha > 0$. Recall, the $\alpha$-shifted open left-half plane is

$$\mathbb{C}_\alpha^- := \{ x \in \mathbb{C} : \text{Re} \, x < -\alpha \},$$

and the $\alpha$-shifted closed right-half plane is

$$\bar{\mathbb{C}}_\alpha^+ := \{ x \in \mathbb{C} : \text{Re} \, x \geq -\alpha \}.$$

**Definition 4.1.1.** Let $\alpha > 0$. The pair $(A, B)$ is $\alpha$-stabilizable if there exists a $K$ such that $\sigma(A + BK) \subset \mathbb{C}_\alpha^-$. The pair $(C, A)$ is $\alpha$-detectable if there exists an $L$ such that $\sigma(A + LC) \subset \bar{\mathbb{C}}_\alpha^-$.

The following lemma allows us to approximate the rate of decay of a matrix exponential by the rate of decay of its largest eigenvalue.

**Lemma 4.1.1.** Let $A \in \mathbb{R}^{n \times n}$ and let $\sigma(A)$ denote the spectrum of $A$. Then for any $\lambda^* > \max_{\lambda \in \sigma(A)} \text{Re}(\lambda)$, there exists $\beta > 0$ such that $\|e^{At}\| \leq \beta e^{\lambda^* t}$ for all $t \geq 0$.

**Proof.** Since $\lim_{t \to \infty} e^{-\lambda^* t} e^{At} = 0$, and $t \mapsto e^{-\lambda^* t} e^{At}$ is continuous, we see that $\beta = \sup_{t \geq 0} \| e^{-\lambda^* t} e^{At} \| < \infty$. Hence we have that

$$\| e^{-\lambda^* t} e^{At} \| \leq \beta, \quad \forall t \geq 0$$

$$e^{-\lambda^* t} \| e^{At} \| \leq \beta, \quad \forall t \geq 0$$

$$\| e^{At} \| \leq \beta e^{\lambda^* t}, \quad \forall t \geq 0.$$
The following lemma extends Hautus’ result on detectability to provide a test for $\alpha$-detectability.

**Lemma 4.1.2.** The pair $(C, A)$ is $\alpha$-detectable if and only if
\[
\text{rank } \begin{bmatrix} A - \lambda I & C \end{bmatrix} = n, \quad \text{for all } \lambda \in \bar{C}_\alpha^+.
\]

**Proof.** We begin with necessity. Supposed $\exists \lambda \in \bar{C}_\alpha^+$ such that
\[
\text{rank } \begin{bmatrix} A - \lambda I & C \end{bmatrix} < n.
\]
Then $\exists v \in \mathbb{C}^n$ such that
\[
\begin{bmatrix} A - \lambda I & C \end{bmatrix} v = 0.
\]
But then for any $L$, $(A - LC - \lambda I)v = 0$, hence $\lambda$ is an eigenvalue of $(A - LC)$ for any $L$. This implies that $(C, A)$ is not $\alpha$-detectable.

For sufficiency, we know that if $(C, A)$ is not $\alpha$-detectable, $\exists$ a nonsingular $T$ such that
\[
TAT^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad CT = \begin{bmatrix} C_1 \\ 0 \end{bmatrix}
\]
with $(C_1, A_{11})$ observable, and hence $\alpha$-detectable. Using our assumption that for all $\lambda \in \bar{C}_\alpha^+$
\[
\text{rank } \begin{bmatrix} A - \lambda I & C \end{bmatrix} = n.
\]
Then we must also have

\[
\text{rank} \begin{pmatrix}
T^{-1} & 0 \\
0 & I
\end{pmatrix}
\begin{pmatrix}
A - \lambda I \\
C
\end{pmatrix} T
= \text{rank} \begin{pmatrix}
T^{-1}AT - \lambda I \\
CT
\end{pmatrix}
= \text{rank} \begin{pmatrix}
A_{11} - \lambda I & A_{12} \\
0 & A_{22} - \lambda I
\end{pmatrix}
= n,
\]

which implies that \( \lambda \notin \sigma(A_{22}) \), which implies that \((C, A)\) is \(\alpha\)-detectable.

A similar result holds for \(\alpha\)-stabilizability. We now use the above result to show that \(\alpha\)-stabilizability of system parameters \((A, B)\) is invariant to state feedback. From Hautus’ test for \(\alpha\)-stabilizability, \((A, B)\) is \(\alpha\)-stabilizable if and only if

\[
\text{rank} \begin{pmatrix}
A - \lambda I & B
\end{pmatrix} < n,
\]

which leads to the following result.

**Lemma 4.1.3.** \((A, B)\) is \(\alpha\)-stabilizable if and only if for all \(K \in \mathbb{R}^{m \times n}\), \((A + BK, B)\) is \(\alpha\)-stabilizable.

**Proof.** Suppose \((A + BK, B)\) is not \(\alpha\)-stabilizable for all \(K \in \mathbb{R}^{m \times n}\). Then there exists a \(\bar{K}\) such that \((A + B\bar{K}, B)\) is not \(\alpha\)-stabilizable. By the above formulation, that implies that there exists an eigenvalue \(\lambda \in \bar{C}_\alpha^+\) of \(A + B\bar{K}\) such that

\[
\text{rank} \begin{pmatrix}
A + B\bar{K} - \lambda I & B
\end{pmatrix} < n.
\]

Therefore, there exists \(v\) such that

\[
v^* \begin{pmatrix}
A + B\bar{K} - \lambda I & B
\end{pmatrix} = 0
\]

where \(v^*\) denotes the conjugate transpose of \(v\). But then

\[
v^* (A + B\bar{K} - \lambda I + B(-\bar{K})) = v^* (A - \lambda I) = 0.
\]
This implies that \( \text{rank} \left[ A - \lambda I \ B \right] < n \), a contradiction.

For the other direction setting \( K = 0 \) yields the desired result. \( \square \)

We now extend the results of typical regulator theory to accommodate two changes for RCP. First, instead of studying purely linear systems, we extend the theory to affine systems: the state equation, exosystem, and error will all have affine terms. Second, we allow for asymptotically stable exosystems.

Consider the system

\[
\begin{align*}
\dot{x} &= Ax + Bu + Ew + a, \\
\dot{w} &= Sw + s, \\
y &= C_1x + C_2w, \\
e &= D_1x + D_2w + d,
\end{align*}
\]

where \( A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, a \in \mathbb{R}^n, E \in \mathbb{R}^{n \times q}, S \in \mathbb{R}^{q \times q}, s \in \mathbb{R}^q, C_1 \in \mathbb{R}^{p \times n}, C_2 \in \mathbb{R}^{p \times q}, D_1 \in \mathbb{R}^{r \times n}, D_2 \in \mathbb{R}^{r \times q}, d \in \mathbb{R}^r. \) Equation (4.1a) describes the plant, (4.1b) describes the exosystem, (4.1c) is the measurement, and (4.1d) is the error to be regulated to zero. The exosystem models two behaviours: desired reference behaviour for the plant and disturbances or external dynamics acting on the plant.

### 4.1.1 Output Regulation with Full Information

We introduce the problem of output regulation with full information.

**Problem 4.1.1 (Regulation with Full Information).** Consider the system (4.1) and let \( 0 < \alpha < \alpha^* \), for some \( \alpha^* \in \mathbb{R} \). Find \( u = H_1x + H_2w + h \) such that the following conditions hold:

- (AS) \( \sigma(A + BH_1) \subset \mathbb{C}_{\alpha}^- \).

- (R) There exists \( \beta > 0 \), such that for all \( (x(0), w(0)) \) and for all \( t \geq 0 \), the closed-loop system satisfies

\[
\|e(t)\| \leq \beta e^{-\alpha^* t} \|e(0)\|.
\]

The regulation requirement for this problem statement is different from typical regulator theory. We provide a guaranteed rate of decay on the error bound, as opposed to simply guaranteeing that the error decays to zero. To solve Problem 4.1.1 we require the following assumptions.

**Assumption 4.1.1.** The system (4.1) satisfies the following:

1. (A1) \( (A, B) \) is \( \alpha \)-stabilizable.
2. (A2) \( \sigma(S) \subset \mathbb{C}_{\alpha}^+ \).
Before we can prove Theorem 4.1.1, we introduce a preliminary lemma that will be used in the proof.

**Lemma 4.1.4.** Suppose Assumption 4.1.1 holds and consider the system \( \dot{w} = Sw + s \) where \( w(t) \in \mathbb{R}^q \) and \( \sigma(S) \subset \bar{C}_+^{\alpha} \). Let \( G \in \mathbb{R}^{p \times q} \) and \( g \in \mathbb{R}^p \). If for all initial conditions \( w(0) \in \mathbb{R}^q \), \( e^{\alpha t} (Gw(t) + g) \to 0 \) as \( t \to \infty \), then \( G = 0 \) and \( g = 0 \).

**Proof.** Let \( w_{01} \) and \( w_{02} \) be two different initial conditions. Then we have that

\[
0 = \lim_{t \to \infty} e^{\alpha t} \left( G \left[ e^{St}w_{01} + \int_0^t e^{S(t-\tau)}s \right] + g \right)
\]

Let \( w_0 = w_{01} - w_{02} \neq 0 \). Then

\[
0 = \lim_{t \to \infty} e^{\alpha t} \left( G \left[ e^{St}w_0 + \int_0^t e^{S(t-\tau)}s \right] + g \right)
\]

\[
0 = \lim_{t \to \infty} e^{\alpha t} \left( G \left[ e^{St}w_0 + \int_0^t e^{S(t-\tau)}s \right] + g \right) - \lim_{t \to \infty} e^{\alpha t} (Ge^{St}w_{02})
\]

\[
0 = - \lim_{t \to \infty} e^{\alpha t} (Ge^{St}w_{02})
\]

\[
0 = - \lim_{t \to \infty} \left( Ge^{(S+1\alpha)t}w_{02} \right)
\]

Since \( \sigma(S + \alpha I) = \sigma(S) + \alpha, \sigma(S + I\alpha) \subset \bar{C}_+^{\alpha} \). Also, since \( w_{02} \) is arbitrary, we have that \( G = 0 \). It immediately follows that \( g = 0 \). \( \square \)

The next result provides regulator equations for the affine extension of regulator theory. The first and third regulator equations are the usual ones, whereas the second and fourth arise specifically to address the affine nature of the problem.

**Theorem 4.1.1.** Consider the system (4.1) and suppose Assumption 4.1.1 holds. Then Problem 4.1.1 is solvable if and only if there exist \((\Pi, \Gamma, p, \gamma)\) such that

\[
\Pi S = A\Pi + B\Gamma + E \tag{4.2a}
\]

\[
\Pi s = Ap + B\gamma + a \tag{4.2b}
\]

\[
0 = D_1\Pi + D_2 \tag{4.2c}
\]

\[
0 = D_1p + d \tag{4.2d}
\]

where \( \Pi \in \mathbb{R}^{n \times q}, \Gamma \in \mathbb{R}^{m \times q}, p \in \mathbb{R}^n, \) and \( \gamma \in \mathbb{R}^m \). Moreover a suitable state feedback solving Prob-
where $K$ is any matrix such that $\sigma(A + BK) \subset C^-\alpha$.

Proof. First we prove that under Assumptions 4.1.1 if there exists a state feedback law $u = H_1x + H_2w + h$ such that $\sigma(A + BH_1) \subset C^-\alpha$ and for all $(x(0), w(0))$, the closed-loop system satisfies $\|e(t)\| \leq \beta e^{-\alpha^* t} \|e(0)\|$, then the affine FBI equations (4.2a)-(4.2d) hold. We first note that

$$II S = (A + BH_1)I + (E + BH_2)$$

$$p \cdot 0 = (A + BH_1)p + (a + Bh - IS)$$

are Sylvester equations. Since by assumption $\sigma(A + BH_1) \cap \sigma(S) = \emptyset$, and $\sigma(A + BH_1) \cap \sigma(0) = \emptyset$, there exists a unique solution $(I, p)$ satisfying the above two Sylvester equations [41]. Alternatively, $p$ can be solved by inverting $A + BH_1$. Setting $\Gamma = H_1I + H_2$ and $\gamma = H_1p + h$, we obtain (4.2a)-(4.2b). Now consider the coordinate transformation $z = x - (Iw + p)$. Then using (4.2a)-(4.2b) we derive that

$$\dot{z} = (A + BH_1)z := \tilde{A}z.$$

Consider the error signal, we have that

$$e(t) = D_1x(t) + D_2w(t) + d$$

$$= D_1x(t) - D_1(Iw(t) + p) + D_1(Iw(t) + p) + D_2w(t) + d$$

$$= D_1\tilde{z}(t) + (D_1I + D_2)w(t) + (D_1p + d).$$

We define the transformations $\tilde{e}(t) := e^{\alpha t}e(t)$, $\tilde{z}(t) := e^{\alpha t}z(t)$, and $\tilde{w}(t) := e^{\alpha t}w(t)$. Considering $\tilde{z}(t)$ we have $\|\tilde{z}(t)\| \leq e^{\alpha t} \|e^{\alpha t} \| \|z(0)\|$. By Problem 4.1.1 (AS), $\sigma(\tilde{A}) \subset C^-\alpha$ so there exists $\lambda^* > \max_{\lambda \epsilon \sigma(\tilde{A})} \{\text{Re}(\lambda)\}$ such that $-\alpha > \lambda^* > \max_{\lambda \epsilon \sigma(\tilde{A})} \{\text{Re}(\lambda)\}$. By Lemma 4.1.2 there exists $\beta > 0$ such that $\|\tilde{z}(t)\| \leq \beta e^{(\alpha + \lambda^*) t} \|z(0)\|$. Since $\alpha + \lambda^* < 0$, we have that $\tilde{z}(t) \rightarrow 0$ as $t \rightarrow \infty$. Similarly it can be shown that $\tilde{e}(t) \rightarrow 0$ as $t \rightarrow \infty$. Since $\tilde{e}(t) = \begin{bmatrix} D_1 & 0 \end{bmatrix} \tilde{z}(t) + (D_1I + D_2)\tilde{w}(t) + e^{\alpha t}(D_1p + d)$, it must be that $e^{\alpha t}[(D_1I + D_2)w(t) + (D_1p + d)] \rightarrow 0$ as $t \rightarrow \infty$. By Lemma 4.1.4, this implies that $D_1I + D_2 = 0$ and $D_1p + d = 0$, which give (4.2c) and (4.2d).

We now assume that there exists solutions $(\Pi, \Gamma, p, \gamma)$ of (4.2a)-(4.2d) and show that Problem 4.1.1 is solvable. Let $u$ be given by (4.3). By Assumption 4.1.1(A1), $(A, B)$ is $\alpha$-stabilizable. Therefore we
can find a $K$ such that $\sigma(A + BK) \subset \mathbb{C}_-\alpha$, meeting our stability requirement for Problem 4.1.1. Define $z = x - (\Pi w + p)$, it can be shown that $\dot{z} = (A + BK)z := \bar{A}z$.

We have as above

$$e(t) = D_1x(t) + D_2w(t) + d$$

$$= D_1z(t) + (D_1\Pi + D_2)w(t) + (D_1p + d)$$

$$= D_1z(t) \text{ by (4.2c),(4.2d).}$$

Therefore $\|e(t)\| \leq \|D_1\| \|e^{\bar{A}t}\| \|z(0)\| \leq \beta e^{\lambda^* t} \|e(0)\|$, where $-\alpha > \lambda^* > \max_{\sigma(A)} \{\text{Re}(\lambda)\}$ and $\beta > 0$ by Lemma 4.1.1. This proves our regulation requirement for Problem 4.1.1.

\[\square\]

Remark 4.1.2. The affine regulator equations (4.2) can also be derived from the standard linear regulator equations by defining an extended exosystem

$$\dot{x} = Ax + Bu + \begin{bmatrix} E & a \end{bmatrix} w_e$$

$$\dot{w}_e = \begin{bmatrix} S & s \\ 0 & 0 \end{bmatrix} w_e$$

$$y = C_1 x + C_2 w_e$$

$$e = D_1 x + \begin{bmatrix} D_2 & d \end{bmatrix} w_e,$$

and setting $w_e(0) = \begin{bmatrix} w_{e1}(0) \\ 1 \end{bmatrix}^T$.

Remark 4.1.3. The regulator equations (4.2a)-(4.2d) can also be derived by starting with the nonlinear regulator equations. Consider the nonlinear system with some slight abuse of notation

$$\dot{x} = f(x, u, w) \quad (4.7)$$

$$e = h(x, w) \quad (4.8)$$

$$\dot{w} = S(w). \quad (4.9)$$

We must find a pair of mappings $\pi(w), c(w)$ that solve

$$\frac{\partial \pi}{\partial w} S(w) = f(\pi(w), c(w), w) \quad (4.10)$$

$$0 = h(\pi(w), w). \quad (4.11)$$
Substituting \( f(x,u,w) = Ax + Bu + Ew + a, h(x,w) = C_1 x + C_2 w, S(w) = Sw + s, \pi(w) = \Pi w + p, \) and \( c(w) = \Gamma w + \gamma \) we recover the equation (4.2a),(4.2c). Performing the same substitution with \( w = 0 \) we recover (4.2b),(4.2d).

### 4.1.2 Output Regulation with Partial Information

In typical engineering systems it is not common to have knowledge of the exosystem’s initial conditions or even the initial conditions of the full state. This brings us to the problem of output regulation with partial state information. We consider dynamic feedback of the form

\[
\begin{align*}
\dot{\xi} &= F\xi + Gy + f, \\
u &= H\xi + h,
\end{align*}
\] (4.12a, 4.12b)

where \( F \in \mathbb{R}^{n_c \times n_c}, G \in \mathbb{R}^{n_c \times p}, f \in \mathbb{R}^{n_c}, H \in \mathbb{R}^{m \times n_c}, \) and \( h \in \mathbb{R}^m. \)

**Problem 4.1.2 (Regulation with Measurement Feedback).** Consider the system (4.1) and let \( 0 < \alpha < \alpha^* \).

Find a dynamic feedback (4.12) such that the following conditions hold:

\[(AS) \quad \sigma \left( \begin{bmatrix} A & BH \\ GC_1 & F \end{bmatrix} \right) \subset \mathbb{C}_-^\alpha.\]

\[(R) \quad \text{For all } (x(0), \xi(0), w(0)) \text{ and for all } t \geq 0, \text{ there exists } \beta > 0 \text{ such that the closed-loop system satisfies } ||e(t)|| \leq \beta e^{-\alpha t} ||e(0)||.\]

**Assumption 4.1.2.** System (4.1) satisfies the following:

\[(A3) \quad \text{The pair } \begin{bmatrix} C_1 & C_2 \end{bmatrix}, \begin{bmatrix} A & E \\ 0 & S \end{bmatrix} \text{ is } \alpha\text{-detectable.}\]

**Theorem 4.1.4.** Consider the system (4.1) and suppose Assumptions 4.1.1 and 4.1.2 hold. Then Problem 4.1.2 is solvable if and only if there exist \((\Pi, \Gamma, p, \gamma)\) such that (4.2) hold. Moreover a suitable dynamic feedback solving Problem 4.1.2 is given by

\[
\begin{align*}
\dot{\xi}_1 &= A \xi_1 + B u + G_1 (y - \hat{y}) + a, \\
\dot{\xi}_2 &= C_1 \xi_1 + C_2 \xi_2, \\
\hat{y} &= C_1 \xi_1 + C_2 \xi_2, \\
u &= \Gamma \xi_2 + K (\xi_1 - (\Pi \xi_2 + p)) + \gamma,
\end{align*}
\] (4.13a, 4.13b)
where $K$ and $G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$ are any matrices such that

\[
\sigma(A + BK) \subset \mathbb{C}_-^ \alpha, \sigma\left( \begin{bmatrix} A - G_1C_1 & E - G_1C_2 \\ -G_2C_1 & S - G_2C_2 \end{bmatrix} \right) \subset \mathbb{C}_-^ \alpha.
\]

Proof. First we prove that, under Assumptions 4.1.1 and 4.1.2, if there exists a state feedback law of the form (4.12) such that (AS) and (R) hold, then (4.2) are solvable. With the regulator of the form (4.12), and $w(t) = 0$, the linearisation of the closed-loop system becomes

\[
\begin{bmatrix}
\dot{x} \\
\dot{\xi}
\end{bmatrix} = \begin{bmatrix} A & BH \\ GC_1 & F \end{bmatrix} \begin{bmatrix} x \\ \xi \end{bmatrix}
\]

which by assumption is stable. Now consider the Sylvester equations

\[
\begin{bmatrix} \Pi \\ \Sigma \end{bmatrix} S = \begin{bmatrix} A & BH \\ GC_1 & F \end{bmatrix} \begin{bmatrix} \Pi \\ \Sigma \end{bmatrix} + \begin{bmatrix} E \\ GC_2 \end{bmatrix} \tag{4.14}
\]

\[
\begin{bmatrix} p \\ \sigma \end{bmatrix} [0] = \begin{bmatrix} A & BH \\ GC_1 & F \end{bmatrix} \begin{bmatrix} p \\ \sigma \end{bmatrix} + \begin{bmatrix} Bh + a - \Pi s \\ f - \Sigma s \end{bmatrix} \tag{4.15}
\]

Since the eigenvalues of $S$ and of the closed-loop system matrix are disjoint, and the eigenvalues of the closed-loop system matrix are disjoint from zero, by Sylvester’s theorem there exists a unique solution for $(\Sigma, p, \sigma)$ satisfying the above two equations. Alternatively, $(p, \sigma)$ can be solved by inverting the matrix in condition (AS). Setting $\Gamma = H^T \Sigma$ and $\gamma = Hp + h$, we obtain (4.2a)-(4.2b).

Next let $z_1 := x - (\Pi w + p)$ and $z_2 := \xi - (\Sigma w + \sigma)$. Then using (4.14) - (4.15) we derive that

\[
\dot{z} = \begin{bmatrix} A & BH \\ GC_1 & F \end{bmatrix} z =: \tilde{A} z.
\]
Considering the error signal, we have that
\[
e(t) = D_1 x(t) + D_2 w(t) + d \]
\[
= D_1 x(t) - D_1 (\Pi w(t) + p) + D_1(\Pi w(t) + p) +
\]
\[
= \begin{bmatrix} D_1 & 0 \end{bmatrix} z(t) + (D_1 \Pi + D_2) w(t) + (D_1 p + d).
\]

We define the transformations \( \tilde{e}(t) := e^{\alpha t} e(t), \tilde{z}(t) := e^{\alpha t} z(t), \) and \( \tilde{w}(t) = e^{\alpha t} w(t). \) Considering \( \tilde{z}(t) \) we have \( \| \tilde{z}(t) \| \leq e^{\alpha t} \left\| e^{\tilde{A} t} \right\| \| z(0) \|. \) By (AS), \( \sigma(\tilde{A}) \subset \mathbb{C}^{-\alpha} \) so there exists \( \lambda^* > \max_{\lambda \in \sigma(\tilde{A})} \{ \Re(\lambda) \} \) such that \( -\alpha > \lambda^* > \max_{\lambda \in \sigma(\tilde{A})} \{ \Re(\lambda) \}. \) By Lemma 4.1.1 there exists \( \beta > 0 \) such that \( \| \tilde{z}(t) \| \leq \beta e^{(\alpha + \lambda^*) t} \| z(0) \|. \) Since \( \alpha + \lambda^* < 0, \) we have that \( \tilde{z}(t) \to 0 \) as \( t \to \infty. \) Similarly it can be shown that \( \tilde{e}(t) \to 0 \) as \( t \to \infty. \) Since \( \tilde{e}(t) = \begin{bmatrix} D_1 & 0 \end{bmatrix} \tilde{z}(t) + (D_1 \Pi + D_2) \tilde{w}(t) + e^{\alpha t}(D_1 p + d), \) it must be that \( e^{\alpha t}(D_1 \Pi + D_2) w(t) + (D_1 p + d) \to 0 \) as \( t \to \infty. \) By Lemma 4.1.4, this implies that \( D_1 \Pi + D_2 = 0 \) and \( D_1 p + d = 0, \) which give (4.2c) and (4.2d).

We now assume that there exist solutions \( (\Pi, \Gamma, p, \gamma) \) of (4.2a)-(4.2d), and show that Problem 4.1.2 is solvable. By Assumption 4.1.1, we can create \( u \) given by (4.13a) - (4.13b), where \( K \) is chosen such that \( \sigma(A + BK) \subset \mathbb{C}^{-\alpha}. \) The main idea is to construct an observer for the composite state \( x_c = (x, w). \) Then the composite system is
\[
\dot{x}_c = A_c x_c + B_c u + a_c, \quad y = C_c x_c, \quad e = D_c x_c + d,
\]
where
\[
A_c = \begin{bmatrix} A & E \\ 0 & S \end{bmatrix}, \quad B_c = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad a_c = \begin{bmatrix} a \\ s \end{bmatrix}, \quad C_c = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, \quad D_c = \begin{bmatrix} D_1 & D_2 \end{bmatrix}.
\]

An observer for the composite system is
\[
\dot{\hat{x}}_c = A_c \hat{x}_c + B_c u + G(y - \hat{y}) + a_c
\]
\[
\dot{\hat{y}} = C_c \hat{x}_c
\]
\[
\dot{\hat{e}} = D_c \hat{x}_c + d.
\]
The estimator error $\hat{x}_c = x_c - \hat{x}_c$ has dynamics $\dot{\hat{x}}_c = (A_c - GC_c)\hat{x}_c$. Since $(C_c, A_c)$ is $\alpha$-detectable, there exists a $G$ such that $A_c - GC_c \subset \mathcal{C}_\alpha^\circ$. We'll show that the above is a regulator with $\xi = \hat{x}_c$. First we'll check the asymptotic stability requirement. Suppose $w(t) = 0$. Then the dynamics of the linearised closed-loop system are given by

$$
\begin{bmatrix}
\dot{x} \\
\dot{\hat{x}}_c
\end{bmatrix} =
\begin{bmatrix}
(A + BK) & -B \begin{bmatrix} K & \Gamma - K\Pi \end{bmatrix} \\
0 & (A_c - GC_c)
\end{bmatrix}
\begin{bmatrix}
x \\
\hat{x}_c
\end{bmatrix},
$$

which satisfies our asymptotic stability requirement. Next consider the regulation requirement. Define $z = x - (\Pi w + p)$. Using (4.2a)-(4.2b), we obtain

$$
\dot{z} = (A + BK)z - B \begin{bmatrix} K & \Gamma - K\Pi \end{bmatrix} \hat{x}_c.
$$

Combining with the dynamics of $\hat{x}_c$ we have the composite dynamics

$$
\begin{bmatrix}
\dot{z} \\
\dot{\hat{x}}_c
\end{bmatrix} =
\begin{bmatrix}
(A + BK) & -B \begin{bmatrix} K & \Gamma - K\Pi \end{bmatrix} \\
0 & (A_c - GC_c)
\end{bmatrix}
\begin{bmatrix}
z \\
\hat{x}_c
\end{bmatrix} =: \bar{A}
\begin{bmatrix}
z \\
\hat{x}_c
\end{bmatrix}.
$$

We have as above

$$
e(t) = D_1 x(t) + D_2 w(t) + d
$$

$$
= \begin{bmatrix} D_1 & 0 \end{bmatrix}
\begin{bmatrix}
z(t) \\
\hat{x}_c(t)
\end{bmatrix} + (D_1 \Pi + D_2)w(t) + (D_1 p + d)
$$

$$
= \begin{bmatrix} D_1 & 0 \end{bmatrix}
\begin{bmatrix}
z(t) \\
\hat{x}_c(t)
\end{bmatrix}
$$

by (4.2c),(4.2d).

Therefore $\|e(t)\| \leq \left\|\begin{bmatrix} D_1 & 0 \end{bmatrix}\right\| \|e^{\bar{A}t}\| \left\|\begin{bmatrix} z(0) \\
\hat{x}_c(0) \end{bmatrix}\right\| \leq \beta e^{\lambda_t} \|e(0)\|$, where $-\alpha > \lambda^* > \max_{\sigma(\bar{A})}\{Re(\lambda)\}$ and $\beta > 0$ by Lemma 4.1.1. This proves our regulation requirement for Problem 4.1.2.

4.1.3 Model Reduction

In this section, we discuss necessary conditions for solving Problem 4.1.2 and how they relate to the assumptions that we have made. It is clear that to achieve the requirement (AS), $(A, B)$ must be $\alpha$-stabilizable. Also $(C_1, A)$ must be $\alpha$-detectable since the measurement $y$ is used in the feedback
controller. On the other hand, to achieve the regulation requirement, it must be that every eigenvalue
that is observable from \(e\) which lies in \(\mathbb{C}_\alpha^+\) is also observable from \(y\). This is a necessary condition, since
otherwise it would not be possible to observe if the regulation requirement is satisfied. We state these
two new necessary conditions next.

**Assumption 4.1.3.** The system (4.1) satisfies the following:

(A1) \((C_1, A)\) is \(\alpha\)-detectable.

(A2) For all \(\lambda \in \mathbb{C}_\alpha^+\)

\[
\ker \begin{bmatrix}
A - \lambda I & E \\
0 & S - \lambda I \\
C_1 & C_2
\end{bmatrix} = \ker \begin{bmatrix}
A - \lambda I & E \\
0 & S - \lambda I \\
C_1 & C_2 \\
D_1 & D_2
\end{bmatrix}.
\]

We first show that these two assumptions are weaker than Assumption 4.1.2, and then show through
a proposition that there is no loss in using Assumption 4.1.2. First, to see that Assumption 4.1.2 implies
Assumption 4.1.3 (A1), we use Hautus’ test for detectability. To that end, suppose Assumption 4.1.3
(A1) does not hold. Then by Lemma 4.1.2 there exists \(\lambda \in \mathbb{C}_\alpha^+\) such that \(\begin{bmatrix}
A - \lambda I \\
C_1
\end{bmatrix}\) is not full column
rank. But then with the same \(\lambda\) the matrix \(\begin{bmatrix}
A - \lambda I & E \\
0 & S - \lambda I \\
C_1 & C_2
\end{bmatrix}\) fails to be full column rank as well, so
Assumption 4.1.2 does not hold. Second, we show that Assumption 4.1.2 implies Assumption 4.1.3 (A2).
Observe that for all \(\lambda \in \mathbb{C}_\alpha^+\) we have

\[
\ker \begin{bmatrix}
A - \lambda I & E \\
0 & S - \lambda I \\
C_1 & C_2
\end{bmatrix} \supseteq \ker \begin{bmatrix}
A - \lambda I & E \\
0 & S - \lambda I \\
C_1 & C_2 \\
D_1 & D_2
\end{bmatrix}.
\]

Since Assumption 4.1.2 and Hautus’ test implies that for all \(\lambda \in \mathbb{C}_\alpha^+\) we have \(\begin{bmatrix}
A - \lambda I \\
C_1
\end{bmatrix} = 0\),
we arrive at Assumption 4.1.3(A2).
Assumption 4.1.2, while not a necessary condition for Problem 4.1.1, implies that an observer can be built for both \( x \) and \( w \). The reasoning why this assumption does not involve a loss of generality is based on the fact that we can decompose the exosystem into a part that is observable from \( e \) and a part not observable from \( e \). The exosystem can be reduced by eliminating states that are not observable from \( e \), and then apply Theorem 4.1.4 with the reduced exosystem \([4, 14]\).

**Proposition 4.1.1.** Suppose that Assumptions 4.1.3 hold, but not Assumption 4.1.2. Consider the composite system

\[
\dot{x}_c = A_c x_c + B_c u + a_c, \quad y = C_c x_c,
\]

where

\[
A_c = \begin{bmatrix} A & E \\ 0 & S \end{bmatrix}, \quad B_c = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad a_c = \begin{bmatrix} a \\ s \end{bmatrix}, \quad C_c = \begin{bmatrix} C_1 & C_2 \\ D_1 & D_2 \end{bmatrix}.
\]

Then there exists a coordinate transformation \( \tilde{x}_c = T x_c \) such that, in the new coordinates

\[
\begin{align*}
\tilde{A}_c &= TA_c T^{-1} = \begin{bmatrix} A & \tilde{E} \\ 0 & \tilde{S} \end{bmatrix}, \\
\tilde{B}_c &= TB_c = \begin{bmatrix} B \\ 0 \end{bmatrix}, \\
\tilde{a}_c &= Ta_c = \begin{bmatrix} a \\ \tilde{s} \end{bmatrix}, \\
\tilde{C}_c &= CT^{-1} = \begin{bmatrix} C_1 & \tilde{C}_2 \\ D_1 & \tilde{D}_2 \end{bmatrix},
\end{align*}
\]

with a partitioned structure \( \tilde{S} = \begin{bmatrix} S_{11} & 0 \\ S_{21} & S_{22} \end{bmatrix} \), \( \tilde{E} = \begin{bmatrix} \tilde{E}_1 & 0 \end{bmatrix} \), \( \tilde{C}_2 = \begin{bmatrix} \tilde{C}_{21} & 0 \end{bmatrix} \), \( \tilde{D}_2 = \begin{bmatrix} \tilde{D}_{21} & 0 \end{bmatrix} \), and \( \tilde{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \). Moreover, we have that \( \begin{bmatrix} C_1 & \tilde{C}_2 \\ D_1 & \tilde{D}_2 \end{bmatrix} \begin{bmatrix} A & \tilde{E}_1 \\ 0 & S_{11} \end{bmatrix} \) is \( \alpha \)-detectable.

**Sketch of proof.** If the pair \((C_c, A_c)\) is not \( \alpha \)-detectable, by an appropriate change of coordinates, one can transform this pair into \((\tilde{C}_c, \tilde{A}_c)\) with the structure \( \tilde{C}_c = \begin{bmatrix} \tilde{C}_{c_1} & 0 \end{bmatrix} \) and \( \tilde{A}_c = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} \), where the pair \((\tilde{C}_{c_1}, A_{11})\) is \( \alpha \)-detectable. Since we have that \((C_1, A)\) is \( \alpha \)-detectable by assumption, we can pick the transformation to obtain \( A_{11} = \begin{bmatrix} A & \tilde{E}_1 \\ 0 & S_{11} \end{bmatrix} \), \( A_{21} = \begin{bmatrix} 0 & S_{21} \end{bmatrix} \), \( A_{22} = S_{22} \), and \( \tilde{C}_{c_1} = \begin{bmatrix} C_1 & \tilde{C}_{21} \\ D_1 & \tilde{D}_{21} \end{bmatrix} \).

We shall see that, with the help of Proposition 4.1.1, Assumption 4.1.2 is without any loss of gen-
Chapter 4. Output Regulation for Affine Systems

Let \[
\begin{bmatrix}
\dot{x} \\
\dot{w}
\end{bmatrix} = T \begin{bmatrix} x \\
w \end{bmatrix},
\] where \( T \) is defined as in Proposition 4.1.1. Then \( \tilde{w} \) may be partitioned as
\[
\tilde{w} = \begin{bmatrix}
\tilde{w}_1 \\
\tilde{w}_2
\end{bmatrix}.
\] In these coordinates we have
\[
\begin{align*}
\dot{x} &= A\tilde{x} + Bu + \tilde{E}_1 \tilde{w}_1 + a \\
\dot{\tilde{w}}_1 &= S_{11} \tilde{w}_1 + s_1 \\
\dot{\tilde{w}}_2 &= S_{21} \tilde{w}_1 + S_{22} \tilde{w}_2 + s_2 \\
y &= C_1 x + \tilde{C}_2 \tilde{w}_1 \\
e &= D_1 \tilde{x} + \tilde{D}_2 \tilde{w}_1 + d.
\end{align*}
\]
We observe that the only terms affecting \( e \) are \( \tilde{x} \) and \( \tilde{w}_1 \). This means solving the regulation problem of the original system is equivalent to solving the regulation problem with the reduced exosystem
\[
\dot{\tilde{w}}_1 = S_{11} \tilde{w}_1 + s_1.
\]
For the new plant and exosystem, we have that \( \tilde{x} \) and \( \tilde{w}_1 \) are \( \alpha \)-detectable from \( y \) and \( e \). Applying Assumption 4.1.3(A2), we have that \( \tilde{x} \) and \( \tilde{w}_1 \) are \( \alpha \)-detectable from just \( y \). Therefore we have that Assumption 4.1.2 holds.

4.1.4 Example

We give an application of the foregoing theory to the control of a robotic manipulator. We consider the problem of the Canadarm tracking a region of the space station, called the environment, which is in a constant drift relative to the Canadarm. Suppose the arm also experiences an external sinusoidal disturbance. Both the drift and the disturbance are modeled by the exosystem. The error to be regulated is the distance between the Canadarm and the environment. The dynamics of the manipulator after feedback linearisation become
\[
\begin{align*}
\dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} w \\
y &= x_1 - w_1 \\
\dot{w} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} w + \begin{bmatrix} 1 \\ 0 \\ -1 & 0 \end{bmatrix} \\
e &= x_1 - w_1
\end{align*}
\]

Here \( w_1 \) provides the reference trajectory, while \((w_2, w_3)\) generate the sinusoidal disturbance. For this example the detectability assumption of Theorem 4.1.4 fails, so we must invoke the model reduction method of the previous section. We obtain a reduced model which satisfies the detectability assumption. The reduced model is

\[
\begin{align*}
A_r &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \\
B_r &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\
E_r &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \\
a_r &= \begin{bmatrix} -1 \\ 0 \end{bmatrix} \\
S_r &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \\
s_r &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
D_{1r} &= \begin{bmatrix} 1 & 0 \end{bmatrix}, \\
D_{2r} &= \begin{bmatrix} 0 & 0 \end{bmatrix}
\end{align*}
\]

The redundancy between the plant and the exosystem was the reason for the failed detectability assumption. The reduced model accounts for the tracking by changing the affine term in the original model, and removing the reference from the exosystem. The reduced model now satisfies the detectability assumption and thus we can apply Theorem 4.1.4. We begin by solving the affine regulator equations which yield

\[
\begin{align*}
\Pi &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\
p &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\
\Gamma &= \begin{bmatrix} 0 & -1 \end{bmatrix}, \\
\gamma &= 0.
\end{align*}
\]
With these matrices, and by placing \( \sigma(A + BK) = \{-1, -2\} \) and \( \sigma(A_c - GC_c) = \{-1, -2, -3, -4\} \), we obtain the following regulator

\[
\dot{\xi} = \begin{bmatrix}
-10 & 1 & 0 & 0 \\
-36 & -3 & 0 & 0 \\
-10 & 0 & 0 & 1 \\
-40 & 0 & -1 & 0
\end{bmatrix} \xi + \begin{bmatrix}
10 \\
34 \\
10 \\
40
\end{bmatrix} y + \begin{bmatrix}
-1 \\
3 \\
0 \\
0
\end{bmatrix}
\]

\[
u = \begin{bmatrix}
-2 \\
-3 \\
0 \\
-1
\end{bmatrix} \xi + 3.
\]

We now apply this regulator to our original model to solve the regulation problem. Figure 4.1 shows the plot of the error evolving over time.

### 4.2 Reach Control Problem with Disturbance Rejection

Now that a regulator theory for affine systems has been put in place, we apply this theory to the problem of disturbance rejection combined with reach control on a simplex. Our approach is to encode the desired behaviour on each simplex in an exosystem.

We present two approaches to solving the disturbance rejection problem with partial state information. In the first method, we introduce a family of reach controllers, one for each simplex. These reach
controllers do not necessarily solve the RCP on their respective simplices, but they capture sufficiently desirable behavior, or they have certain desirable trajectories that emulate the requirements of the RCP sufficiently closely. Two exosystems are defined for this method. The first exosystem models the closed-loop behavior using the assigned reach controller for a simplex. By setting the initial condition of this (known) exosystem, the output of the exosystem is the desired reference trajectory to be tracked. A second exosystem is used to model unknown persistent disturbances acting on the affine control system. The combination of these two exosystems with the tracking and disturbance rejection specifications sets up a classic problem of regulator theory. the only difference is that we use regulator theory for affine systems, as developed above.

In the second method, we again assume that we have available reach controllers, one for each simplex. Rather than using an exosystem to characterize a single reference trajectory of the closed-loop system, instead, we first apply the reach controller of each simplex to the affine control system. Then to perform disturbance rejection, an controller is derived using regulator theory. The disturbance is modeled by an exosystem, and the second controller is regarded as an add-on to the reach controller. Because the vector field in essence is already closed-loop, we forego the usual regulation requirement in regulator theory to track a specific reference trajectory. Instead, the regulation requirement is simply to reject the disturbance.

4.2.1 Method 1

Let $\mathcal{P} \in \mathbb{R}^n$ be a full dimensional polyhedron and let $\mathcal{T} = \{S_1, \ldots, S_l\}$ be a triangulation of $\mathcal{P}$. Consider the system defined on $\mathcal{P}$

\[
\begin{align*}
\dot{x} &= Ax + Bu + Ew_2 + a \\
\dot{w}_1 &= (A + BK_{\kappa(w_1)})w_1 + (a + Bg_{\kappa(w_1)}) \\
\dot{w}_2 &= Sw_2 + s \\
y &= C_1 x + C_2 w_2 \\
e &= x - w_1.
\end{align*}
\]

Observe that the exosystem has been split into an exosystem (4.16b) that describes the desired behavior on each simplex and (4.16c) that generates the disturbance. In (4.16b) it is assumed that reach controllers $u_{rcp}^i = K^i x + g^i, i = 1, \ldots, l$ are available to generate the desired behaviour on each simplex. The index $\kappa(w_1) = i$ when $w_1 \in S_i$. We require the following assumptions.
Assumption 4.2.1. The system \((4.16)\) satisfies the following:

(A1) \((A, B)\) is \(\alpha\)-stabilizable.

(A2) For each \(i \in \{1, \ldots, l\}\), \(\sigma(A + BK^i) \subset \bar{C_\alpha}^+\) and \(\sigma(S) \subset \bar{C_\alpha}^+\).

(A3) \[
\begin{bmatrix}
C_1 & C_2
\end{bmatrix},
\begin{bmatrix}
A & E \\
0 & S
\end{bmatrix}
\] is \(\alpha\)-detectable.

(A4) \(w_1(0)\) is known.

We have included the third assumption since the desired reference behaviour will be generated by the designer, and therefore would be known. The assumption can be removed with a slight modification to the proof.

Problem 4.2.1. Consider the system \((4.16)\), RCP controllers \(u_{rcp}^1, \ldots, u_{rcp}^l\), and let \(0 < \alpha < \alpha^*\). Find a dynamic feedback of the form

\[
\dot{\xi} = F\xi + Gy + f \tag{4.17}
\]

\[
u^\kappa(w_1) = H\xi + h + K^\kappa(w_1)w_1 + g^\kappa(w_1), \tag{4.18}
\]

such that the following conditions hold:

(AS) \(\sigma\left(\begin{bmatrix}
A & BH \\
GC_1 & F
\end{bmatrix}\right) \subset \mathbb{C_\alpha}^-\),

(R) For all \((x(0), \xi(0), w_1(0), w_2(0))\) and for all \(t \geq 0\), there exists \(\beta > 0\) such that the closed loop system satisfies \(\|e(t)\| \leq \beta e^{-\alpha^* t}\|e(0)\|\).

Theorem 4.2.1. Consider the system \((4.16)\) and suppose Assumption 4.2.1 holds. Then Problem 4.2.1 is solvable if and only if there exists \(\Gamma\) such that \(B\Gamma + E = 0\). Moreover a suitable dynamic feedback solving Problem 4.2.1 is given by

\[
\begin{bmatrix}
\xi_1 \\
\xi_2
\end{bmatrix} = \begin{bmatrix}
A & E \\
0 & S
\end{bmatrix} \begin{bmatrix}
\xi_1 \\
\xi_2
\end{bmatrix} + \begin{bmatrix}
B \\
0
\end{bmatrix} u + \begin{bmatrix}
G_1 \\
G_2
\end{bmatrix}(y - \hat{y}) + \begin{bmatrix}
a \\
s
\end{bmatrix} \tag{4.19a}
\]

\[
u^\kappa(w_1) = K^\kappa(w_1)w_1 + g^\kappa(w_1) + K(\xi_1 - w_1) + \Gamma \xi_2 \tag{4.19b}
\]

where \(K\) and \(G = \begin{bmatrix}
G_1^T & G_2^T
\end{bmatrix}^T\) are any matrices such that \(\sigma\left(\begin{bmatrix}
A - G_1C_1 & E - G_1C_2 \\
-G_2C_1 & S - G_2C_2
\end{bmatrix}\right) \subset \mathbb{C_\alpha}^-\) and \(\sigma(A + BK) \subset \mathbb{C_\alpha}^-\).
Proof. We begin with necessity. With the regulator of the form (4.17), and \( w_1(t) = w_2(t) = 0 \), the linearisation of the closed-loop system becomes

\[
\begin{bmatrix}
\dot{x} \\
\dot{\xi}
\end{bmatrix} =
\begin{bmatrix}
A & BH \\
GC_1 & F
\end{bmatrix}
\begin{bmatrix}
x \\
\xi
\end{bmatrix}
\]

which by assumption is stable. Now consider the Sylvester equations for each \( i \in \{1, \ldots, l\} \)

\[
\begin{bmatrix}
\Pi_1 & \Pi_2 \\
\Sigma_1 & \Sigma_2
\end{bmatrix}
\begin{bmatrix}
A + BK^i & 0 \\
0 & S
\end{bmatrix}
= \begin{bmatrix}
A & BH \\
GC_1 & F
\end{bmatrix}
\begin{bmatrix}
\Pi_1 & \Pi_2 \\
\Sigma_1 & \Sigma_2
\end{bmatrix} + \begin{bmatrix}
BK^i & E \\
0 & GC_2
\end{bmatrix}
\]

(4.20)

\[
\begin{bmatrix}
p \\
\sigma
\end{bmatrix}
= \begin{bmatrix}
A & BH \\
GC_1 & F
\end{bmatrix}
\begin{bmatrix}
p \\
\sigma
\end{bmatrix} + \begin{bmatrix}
Bg^i + B\alpha - \Pi_1(a + Bg^i) - \Pi_2s \\
f - \Sigma_1(a + Bg^i) - \Sigma_2s
\end{bmatrix}
\]

(4.21)

By Assumption (A2) and Sylvester’s theorem there exists a unique solution for \((\Pi_1, \Pi_2, \Sigma_1, \Sigma_2, p, \sigma)\) satisfying the above two equations. Alternatively, \((p, \sigma)\) can be solved by inverting the matrix in condition (AS).

Next let \( z_1 := x - (\Pi_1 w_1 + \Pi_2 w_2 + p) \) and \( z_2 := \xi - (\Sigma_1 w_1 + \Sigma_2 w_2 + \sigma) \). Then using (4.20) - (4.21) we derive that

\[
\dot{z} = \begin{bmatrix}
A & BH \\
GC_1 & F
\end{bmatrix} z =: \tilde{A} z.
\]

Considering the error signal, we have that

\[
e(t) = x(t) - w_1(t) = x(t) - (\Pi_1 w_1(t) + \Pi_2 w_2(t) + p) + (\Pi_1 w_1(t) + \Pi_2 w_2(t) + p) - w_1(t) = \begin{bmatrix}
I & 0
\end{bmatrix} z(t) + \begin{bmatrix}
\Pi_1 - I & \Pi_2
\end{bmatrix} \begin{bmatrix}
w_1(t) \\
w_2(t)
\end{bmatrix} + p.
\]

We define the transformations \( \hat{e}(t) := e^{\alpha t} e(t) \), \( \hat{z}(t) := e^{\alpha t} z(t) \), \( \hat{w}_1(t) := e^{\alpha t} w_1(t) \), and \( \hat{w}_2(t) := e^{\alpha t} w_2(t) \). Considering \( \tilde{z}(t) \) we have \( \|\tilde{z}(t)\| \leq e^{\alpha t} \|e^{\tilde{A} t}\| \|z(0)\| \). By (AS), \( \sigma(\tilde{A}) \subset C_\alpha \) so there exists \( \lambda^* > \max_{\lambda \in \sigma(\tilde{A})} \{\text{Re}(\lambda)\} \) such that \(-\alpha > \lambda^* > \max_{\lambda \in \sigma(\tilde{A})} \{\text{Re}(\lambda)\} \). By Lemma 4.1.1 there exists \( \beta > 0 \) such that \( \|\tilde{z}(t)\| \leq \beta e^{(\alpha + \lambda^*) t} \|z(0)\| \). Since \( \alpha + \lambda^* < 0 \), we have that \( \tilde{z}(t) \to 0 \) as \( t \to \infty \). Similarly it can be shown that \( \hat{e}(t) \to 0 \) as \( t \to \infty \). Since \( \hat{e}(t) = \begin{bmatrix}
I & 0
\end{bmatrix} \tilde{z}(t) + \begin{bmatrix}
\Pi_1 - I & \Pi_2
\end{bmatrix} \begin{bmatrix}
\hat{w}_1(t) \\
\hat{w}_2(t)
\end{bmatrix} + e^{\alpha t} p \), it must...
be that $e^{\alpha t} \begin{bmatrix} \Pi_1 - I & \Pi_2 \\ w_1(t) & w_2(t) \end{bmatrix} + p \rightarrow 0$ as $t \rightarrow \infty$. By Lemma 4.1.4, this implies that $\Pi_1 = I$, $\Pi_2 = 0$ and $p = 0$. Looking back at (4.20), this implies that with $\Gamma = H\Sigma_2$, $B\Gamma + E = 0$.

For sufficiency we construct an observer for $x$ and $w_2$ of the form (4.19a). Define the estimator error states $\hat{\xi}_1 = x - \xi_1$ and $\hat{\xi}_2 = w_2 - \xi_2$. Then we verify

$$\dot{\hat{\xi}} = \begin{bmatrix} A - G_1C_1 & E - G_1C_2 \\ -G_2C_1 & S - G_2C_2 \end{bmatrix}\hat{\xi}.$$

By (A2) we can choose $G_1$ and $G_2$ such that the estimator error dynamics have poles in $\mathbb{C}_\alpha^-$. Let $u$ be given by (4.19b). Then the linearised closed-loop system with $w_1 = 0$ is given by

$$\begin{bmatrix} \dot{x} \\ \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} A +BK & -BK & -B\Gamma \\ 0 & A - G_1C_1 & E - G_1C_2 \\ 0 & -G_2C_1 & S - G_2C_2 \end{bmatrix}\begin{bmatrix} x \\ \xi_1 \\ \xi_2 \end{bmatrix}.$$

The system matrix has its spectrum in $\mathbb{C}_\alpha^-$ by the choice of $K$ and $G$. Then we can verify that with $H_1 := [B \Gamma]$, $F := \begin{bmatrix} A + BK - G_1C_1 & E + B\Gamma - G_1C_2 \\ -G_2C_1 & S - G_2C_2 \end{bmatrix}$, and $T := \begin{bmatrix} I & 0 & 0 \\ I & -I & 0 \\ 0 & 0 & -I \end{bmatrix}$, then $T^{-1} \begin{bmatrix} A & BH_1 \\ GC_1 & F \end{bmatrix} T$ is equal to the system matrix above. This proves (AS).

For the regulation requirement (R) it can be verified that using $B\Gamma + E = 0$ and $u$ given in (4.19b) we have $\dot{e} = (A + BK)e - BK\hat{\xi}_1 - B\Gamma\hat{\xi}_2$. Since $\sigma(A + BK) \subset \mathbb{C}_\alpha^-$ and $\hat{\xi}_1$ and $\hat{\xi}_2$ decay according to poles in $\mathbb{C}_\alpha^-$, we obtain (R).

### 4.2.2 Method 2

The second method exploits the fact that the reach control problem regards achieving a desired phase portrait. We modify the regulation requirement to allow the tracking of a phase portrait instead of tracking an individual signal.

Let $\mathcal{P} \in \mathbb{R}^n$ be a full dimensional polyhedron and let $\mathcal{T} = \{S_1, \ldots, S_l\}$ be a triangulation of $\mathcal{P}$.
Consider the system

\[
\dot{x} = Ax + Bu + Ew + a \quad (4.22a)
\]
\[
\dot{w} = Sw + s \quad (4.22b)
\]
\[
y = C_1x + C_2w, \quad (4.22c)
\]

and reach controllers \(u^i_{rcp} = K^i x + g^i\). The main difference between this model and that for Method 1 is that the exosystem only generates the disturbance. We require the following assumption.

**Assumption 4.2.2.** The system (4.22) satisfies the following:

\[
\begin{pmatrix}
C_1 \\
C_2
\end{pmatrix}

\begin{bmatrix}
A & E \\
0 & S
\end{bmatrix}

\text{is detectable.}

**Problem 4.2.2.** Given a polytope \(P\), a triangulation of \(\mathcal{P} = \{S_1, \ldots, S_l\}\), the system (4.22), and RCP controllers \(u^1_{rcp}, \ldots, u^l_{rcp}\). Find dynamic feedbacks on each simplex \(S_i, i = 1, \ldots, l\), of the form

\[
\dot{\xi} = F\xi + Gy + f \quad (4.23)
\]
\[
u^{\kappa(\xi)} = H^{\kappa(\xi)}\xi + h^{\kappa(\xi)}, \quad (4.24)
\]

where \(\kappa(\xi)\) is a state-dependent switching signal, and such that

\[
(R) \quad \text{For all } (x(0), \xi(0), w(0)) \text{ the closed-loop system satisfies } \lim_{t \to \infty} \| (Ax(t) + Bu(t) + Ew(t) + a) - ((A + BK^{\kappa(\xi)})x(t) + (a + Bg^{\kappa(\xi)})) \| = 0.
\]

Notice that the (AS) requirement has been removed. This is to show that the problem is no longer a tracking problem; it is simply a disturbance rejection problem. Then we can achieve the desired phase portrait without the need to track any individual signal. We also do not require the use of the \(\alpha\)-shifted complex plane since we are not building a desired reference behaviour in an exosystem that may have stable poles.

**Theorem 4.2.2.** Consider the system (4.22) and suppose Assumption 4.2.2 holds. Then Problem 4.2.2 is solvable if \(\exists \Gamma\) such that \(B\Gamma + E = 0\). Moreover a suitable dynamic feedback solving Problem 4.2.2 is
given by

\[
\begin{bmatrix}
\dot{\xi}_1 \\
\dot{\xi}_2
\end{bmatrix} = \begin{bmatrix} A & E \\ 0 & S \end{bmatrix} \begin{bmatrix} \xi_1 \\
\xi_2
\end{bmatrix} + \begin{bmatrix} B \\
0
\end{bmatrix} u + \begin{bmatrix} G_1 \\
G_2
\end{bmatrix} (y - \hat{y}) + \begin{bmatrix} a \\
s
\end{bmatrix}
\]

\[
\dot{y} = C_1 \xi_1 + C_2 \xi_2
\]

\[
u = K(x) \xi_1 + g(x) \xi_2 + \Gamma \xi_2
\]

where \( G = \begin{bmatrix} G_1^T \\
G_2^T \end{bmatrix} \) is chosen such that \( \sigma \left( \begin{bmatrix} A - G_1 C_1 & E - G_1 C_2 \\
-G_2 C_1 & S - G_2 C_2 \end{bmatrix} \right) \subset C^- \).

**Proof.** We construct an observer for \( x \) and \( w \) as follows.

\[
\begin{bmatrix}
\dot{\hat{\xi}}_1 \\
\dot{\hat{\xi}}_2
\end{bmatrix} = \begin{bmatrix} A & E \\ 0 & S \end{bmatrix} \begin{bmatrix} \hat{\xi}_1 \\
\hat{\xi}_2
\end{bmatrix} + \begin{bmatrix} B \\
0
\end{bmatrix} u + \begin{bmatrix} G_1 \\
G_2
\end{bmatrix} (y - \hat{y}) + \begin{bmatrix} a \\
s
\end{bmatrix}
\]

Define the estimator error states \( \hat{\xi}_1 = x - \xi_1 \) and \( \hat{\xi}_2 = w - \xi_2 \). Then

\[
\begin{bmatrix}
\dot{\hat{\xi}}_1 \\
\dot{\hat{\xi}}_2
\end{bmatrix} = \left( \begin{bmatrix} A & E \\ 0 & S \end{bmatrix} - \begin{bmatrix} G_1 \\
G_2
\end{bmatrix} \begin{bmatrix} C_1 \\
C_2
\end{bmatrix} \right) \begin{bmatrix} \hat{\xi}_1 \\
\hat{\xi}_2
\end{bmatrix}.
\]

By (A1) we can choose \( G_1 \) and \( G_2 \) such that the estimator error dynamics are asymptotically stable. Therefore \( \xi_1(t) \to x(t) \), and \( \xi_2(t) \to w(t) \) as \( t \to \infty \). Let \( u(t) \) be given by (4.25b). Then since \( BT + E = 0 \),

\[
\lim_{t \to \infty} || (Ax(t) + Bu(t) + Ew(t) + a) - ((A + BK^x) x(t) + (a + Bg^{x})) ||
\]

\[
= \lim_{t \to \infty} || BK^x (\xi_1(t) - x(t)) + BT \xi_2(t) + Ew(t) ||
\]

\[
= 0
\]

Therefore our regulation requirement has been achieved.

**Example 4.2.1.** We return to the example of the Canadarm considered above. This time the exosystem only models the disturbance, as the reach controllers will provide the tracking part of the design. The
dynamics of the end effector of the manipulator after feedback linearisation are given by

\[
\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} w \\
y = x_1 \\
\dot{w} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} w.
\]

The control objective is to find a state feedback such that the closed-loop system achieves the following specifications.

(S1) **Temporal Sequence**
- The end effector begins at \( x_1(0) \in [0, 10] \). Eventually it reaches \( x_1(0) = 15 \), corresponding with the Canadarm being in contact with the environment.

(S2) **Safety**
- \( 0 \leq x_1 \leq 15 \)
- \( |x_2| \leq 5 \)
- If \( x_1 \in [10, 15] \) and \( x_2 \geq 0 \), then \( x_1 + x_2 \leq 15 \)

(S3) **Liveness**
- If \( x_1 \in [0, 10] \), then \( x_2 \geq 0 \)
- If \( x_1 \in [10, 15] \) and \( x_2 \leq 0 \), then \( x_1 + x_2 \geq 10 \).

The Safety and Liveness specifications are encoded as linear inequalities. These linear inequalities define half-planes in the state space, and combining these half-planes defines a polytope. For example the leftmost facet of the polytope is defined by the Safety specification that \( x_1 >= 0 \), and the topmost facet of the polytope is defined by the Safety specification that \( |x_2| <= 0 \).

The reach controllers designed to solve these specifications are shown in the Simplex Data table. The close-loop behaviour of the system without disturbances is shown by the blue trajectories in Figure 4.2. Although there are no formal guarantees that the system with a disturbance satisfies the specifications, it is shown in the red trajectory of Figure 4.2 that there exists an initial condition and disturbance such that the specifications are achieved.

Combining the affine inequalities for safety and liveness, one obtains a state space which is a non-convex polytope \( \mathcal{P} \), as depicted in Figure 4.2. The temporal sequence determines movement of closed-loop trajectories through this polytope. Following the design procedure based on RCP, \( \mathcal{P} \) must be triangulated.
The first step is to compute reach controllers based on this triangulation and the desired closed-loop dynamics. The reach controllers $v_i = K_i x + g_i$ can be calculated following the method of [6]. Table 4.2.2 below shows the simplex data along with the computed reach controllers for each simplex. We now apply Theorem 4.2.1 to achieve the disturbance rejection requirement of the problem. It can be seen that Assumption 4.2.1 holds for this problem, and also that $\Gamma = \begin{bmatrix} 0 & -1 \end{bmatrix}$ solves the equation.
Chapter 4. Output Regulation for Affine Systems

$BT + E = 0$. By placing $\sigma(A_c - GC_c) = \{-1, -2, -3\}$ we obtain the following regulator

\[
\begin{bmatrix}
  \dot{\xi}_1 \\
  \dot{\xi}_2
\end{bmatrix} =
\begin{bmatrix}
  0 & 1 & 0 & 0 \\
  0 & 0 & 0 & 1 \\
  0 & 0 & 0 & 1 \\
  0 & 0 & -1 & 0
\end{bmatrix}
\begin{bmatrix}
  \xi_1 \\
  \xi_2
\end{bmatrix} + 
\begin{bmatrix}
  0 \\
  1 \\
  0 \\
  0
\end{bmatrix} v_i +
\begin{bmatrix}
  6.0140 & -0.9482 \\
  8.5544 & -3.7495 \\
  0.4915 & -3.9860 \\
  1.2545 & -1.9398
\end{bmatrix} (y - \hat{y})
\]

\[
\dot{y} =
\begin{bmatrix}
  1 \\
  0
\end{bmatrix} \xi_1 +
\begin{bmatrix}
  0 & 0 \\
  0 & -1 \\
\end{bmatrix} \xi_2
\]

\[
v_i = K_i \xi_1 + g_i - 
\begin{bmatrix}
  0 & 1
\end{bmatrix} \xi_2.
\]

This regulator solves our disturbance rejection problem. Figure 4.3 shows the closed-loop phase portrait. It can be seen that the control specifications have been met.
4.3 Well-Posedness and Robust Regulation

In this section we extend the results on output regulation for affine systems to include robustness to uncertainties in the plant parameters. We focus only on the general formulation given in Problem 4.1.2, and not on the RCP specific subproblems which were derived from it. Our development here follows closely the development of by Saberi, Stoorvogel, and Sannuti [14].

4.3.1 Well-Posedness

We first discuss the notion of well-posedness of the output regulation problem. A problem is called well-posed if it is solvable, and if it remains solvable on some neighborhood of the problem data.

Suppose that the matrices in a given system $\Sigma = (A, B, E, a, C_1, C_2, c, D_1, D_2, d, S, s)$ are not known exactly. Define the space of parameters

$$\mathcal{P} = \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^{n \times q} \times \mathbb{R}^n \times \mathbb{R}^{p \times n} \times \mathbb{R}^{r \times n} \times \mathbb{R}^r \times \mathbb{R}^{q \times q} \times \mathbb{R}^q.$$

We can express uncertainty in the system parameters by allowing the set $(A, B, E, a, C_1, C_2, c, D_1, D_2, d, S, s)$ to vary in some neighborhood $\mathcal{P}_0$ of the nominal values $(A_0, B_0, E_0, a_0, C_{1,0}, C_{2,0}, c_0, D_{1,0}, D_{2,0}, d_0, S_0, s_0) \in \mathcal{P}$. To show clearly that the parameters $(A, B, E, a, C_1, C_2, c, D_1, D_2, d, S, s)$ are known only with certain tolerances around the nominal values $(A_0, B_0, E_0, a_0, C_{1,0}, C_{2,0}, c_0, D_{1,0}, D_{2,0}, d_0, S_0, s_0)$, we
can write the equation
\[(A, B, E, a, C_1, C_2, c, D_1, D_2, d, S, s) = (A_0 + \delta A, B_0 + \delta B, E_0 + \delta E, a_0 + \delta a, C_{1_0} + \delta C_1, C_{2_0} + \delta C_2, c_0 + \delta c, D_{1_0} + \delta D_1, D_{2_0} + \delta D_2, d_0 + \delta d, S_0 + \delta S, s_0 + \delta s).\]

In order to properly understand the problem of well-posedness of the output regulation problem, we first study well-posedness of a set of linear equations. Let \(X\) and \(Y\) be finite-dimensional linear spaces and let \(A : X \to Y\) be a linear map and \(b \in Y\). Consider the linear equation
\[Ax = b \quad (4.26)\]
in the variable \(x\). It is said that the equation (4.26) is well-posed at the nominal parameter values \((A_0, b_0)\) if there exists a neighborhood of \((A_0, b_0)\) in the parameter space such that (4.26) is solvable in this neighborhood.

**Lemma 4.3.1** ([14]). Equation (4.26) in the variable \(x\) is well-posed at the nominal parameter values \((A_0, b_0)\) if and only if \(A_0\) is surjective.

We now formally state the problem of well-posedness for output regulation problem.

**Definition 4.3.1** (Well-Posedness). The system (4.1) is said to be well posed for Problem 4.1.2 at the nominal parameters \((A_0, B_0, E_0, a_0, C_{1_0}, C_{2_0}, c_0, D_{1_0}, D_{2_0}, d_0, S_0, s_0)\) if there exists a neighborhood of the nominal values such that Problem 4.1.2 is solvable for each element in that neighborhood.

The first thing to notice in the problem of well-posedness is that Assumption 4.1.1(A1) and Assumption 4.1.2(A3) are invariant to small perturbation, while Assumption 4.1.1(A2) may not be. On the other hand, Assumption 4.1.1(A2) is used to show that the solvability of the affine FBI equations is a sufficient condition for the solvability of the output regulation problem. So the problem of well-posedness of the output regulation problem can be restated as the problem of well-posedness of the affine FBI equations (4.2). Therefore, Problem 4.1.2 is well posed if and only if the equations (4.2) are solvable for both the nominal parameters, and also for all the parameters sufficiently close to the nominal parameters. Thus, we require that with the unknowns \(\Pi, \Gamma, p, \gamma,\)
\[
\begin{align*}
\Pi S - A\Pi - B\Gamma & = E \\
\Pi s - Ap - B\gamma & = a \\
D_1\Pi & = -D_2 \\
D_1p & = -d 
\end{align*}
\]
is well-posed at the nominal parameters \((A_0, B_0, E_0, a_0, C_{1_0}, C_{2_0}, c_0, D_{1_0}, D_{2_0}, d_0, S_0, s_0)\). By Lemma 4.3.1, the above linear equations are well-posed if and only if the linear map

\[
L : (\Pi, \Gamma, p, \gamma) \rightarrow (\Pi S - A\Pi - B\Gamma, \Pi s - Ap - B\gamma, D_1\Pi, D_1p) \tag{4.28}
\]
is surjective at \((A_0, B_0, E_0, a_0, C_{1_0}, C_{2_0}, c_0, D_{1_0}, D_{2_0}, d_0, S_0, s_0)\). We can achieve this condition by following the method of universal solvability as proposed by Hautus [14]. An equation of the form

\[
\sum_{i=1}^{k} A_i X S_i = R \tag{4.29}
\]
is said to be universally solvable if it has a solution for every \(R\).

**Theorem 4.3.1** ([14]). Let \(\bar{A}_i \in \mathbb{R}^{\bar{n} \times \bar{s}_i}\), \(\bar{S} \in \mathbb{R}^{\bar{q} \\bar{q}}\), and \(\bar{R} \in \mathbb{R}^{\bar{n} \times \bar{q}}\). Also, let \(q_i(\lambda)\) be polynomials for \(i = 1, \ldots, k\). Consider a matrix polynomial in the variable \(\lambda\),

\[
\bar{A}(\lambda) := \sum_{i=1}^{k} \bar{A}_i q_i(\lambda). \tag{4.30}
\]

Then the equation

\[
\sum_{i=1}^{k} \bar{A}_i X q_i(\bar{S}) = \bar{R} \tag{4.31}
\]
is universally solvable if and only if the matrix \(\bar{A}(\lambda)\) has full row-rank for each \(\lambda\) which is an eigenvalue of \(\bar{S}\). If this is the case and \(\bar{A}(\lambda)\) is square, then the solution \(X\) is unique.
This results helps us prove the main result for well-posedness of the output regulation problem.

**Theorem 4.3.2.** Suppose Assumption 4.1.1 and Assumption 4.1.2 hold. Then the system (4.1) is well-posed for Problem 4.1.2 at the nominal parameters \((A_0, B_0, E_0, a_0, C_{10}, C_{20}, c_0, D_{10}, D_{20}, d_0, S_0, s_0)\) if and only if the matrix

\[
\begin{bmatrix}
  A_0 - \lambda I & B_0 \\
  D_{10} & 0
\end{bmatrix}
\]  

(4.32)

has full row rank for \(\lambda = 0\), and for each \(\lambda\) which is an eigenvalue of \(S_0\).

**Proof.** We begin with the sufficiency part of the proof. Note that by the assumption of the theorem and by continuity, there exists a neighborhood \(P_0\) of the nominal values such that for each \((A, B, E, a, C_1, C_2, c, D_1, D_2, d, S, s)\) \(\in P_0\), \((A, B)\) is \(\alpha\)-stabilizable, and

\[
\left( \begin{bmatrix} C_1 & C_2 \end{bmatrix}, \begin{bmatrix} A & E \\ 0 & S \end{bmatrix} \right)
\]

is \(\alpha\)-detectable. Therefore Assumption 4.1.1(A1) and Assumption 4.1.2(A3) hold. Also by (4.32) and by continuity, for each \((A, B, E, a, C_1, C_2, c, D_1, D_2, d, S, s)\) \(\in P_0\) the matrix

\[
\begin{bmatrix}
  A - \lambda I & B \\
  D_1 & 0
\end{bmatrix}
\]

has full row-rank for \(\lambda = 0\), and for each \(\lambda\) that is an eigenvalue of \(S\).

We now invoke Theorem 4.3.1 in two separate instances to show that first (4.2a) and (4.2c) hold, and again to show that (4.2b) and (4.2d) hold. By the notation given in Theorem 4.3.1, \(\bar{A}(\lambda), \bar{R},\) and \(X\) for (4.2a) and (4.2c) are given by

\[
\bar{A}(\lambda) = \bar{A}_1 q_1(\lambda) + \bar{A}_2 q_2(\lambda) = \begin{bmatrix} A & B \\ D_1 & 0 \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \cdot (-\lambda), \quad \bar{R} = \begin{bmatrix} E \\ -D_2 \end{bmatrix}, \quad X = \begin{bmatrix} \Pi \\ \Gamma \end{bmatrix}
\]

and therefore equations (4.2a) and (4.2c) are solvable for \((\Pi, \Gamma)\). Also by the notation given in Theorem
4.3.1, $\bar{A}(\lambda)$, $\bar{R}$ and $X$ for (4.2b) and (4.2d) are given by

$$\bar{A}(\lambda) = \bar{A}_1 q_1(\lambda) = \begin{bmatrix} A & B \\ D & 0 \end{bmatrix} \cdot 1, \quad \bar{R} = \begin{bmatrix} \Pi s - a \\ -d \end{bmatrix}, \quad X = \begin{bmatrix} p \\ \gamma \end{bmatrix}$$

and therefore equations (4.2b) and (4.2d) are solvable for $(p, \gamma)$. Therefore by Theorem 4.1.4, Problem 4.1.2 is solvable for each $(A, B, E, a, C_1, C_2, c, D_1, D_2, d, S, s) \in P_0$.

For the necessity portion of the proof, we note again by Theorem 4.3.1, if the matrix

$$\begin{bmatrix} A - \lambda I & B \\ D & 0 \end{bmatrix}$$

does not have full row-rank for $\lambda = 0$, or for some $\lambda$ that is an eigenvalue of $S$, then either the set of all $(E, D_2)$ for which the linear equations,

$$\Pi S = A_0 \Pi + B_0 \Gamma + E \\
0 = D_{10} \Pi + D_2$$

are solvable spans only a proper subspace of $\mathbb{R}^{n \times q} \times \mathbb{R}^{r \times q}$, or the set of all $(\Pi s - a, d)$ for which the linear equations

$$\Pi s = A_0 p + B_0 \gamma + a \\
0 = D_{10} p + d$$

are solvable only span a proper subspace of $\mathbb{R}^n \times \mathbb{R}^r$. Therefore, the existence conditions in Theorem 4.1.4 cannot be satisfied at each $(A, B, E, a, C_1, C_2, c, D_1, D_2, d, S, s) \in P_0$, a contradiction.

4.3.2 Structural Stability

The previous section was only concerned with the existence of a controller that solves the output regulation problem for a given set of perturbed parameters. This controller may need to be changed for each new set of perturbed parameters. In this section on structural stability, we search for a deeper result. We are looking for one fixed controller that solves the output regulation problem for all perturbed parameters in a neighborhood of the nominal values.

Just as in the literature for the purely linear structurally stable output regulation problem we will
make two assumptions. The first assumption is that we only perturb the plant, and not the exosystem. The reason for this is that the system must have a zero from \( w \) to \( e \) at all frequencies of the exosystem. A perturbation of the exosystem will in general move that zero, and the closed-loop system will no longer achieve output regulation.

The second assumption is that we have a specific structure for our measurement. In particular, we partition the output \( y = (e, y_2) \) to include the error. This leads to a partition on \( C_1, C_2, \) and \( c \) as follows:

\[
C_1 = \begin{bmatrix} D_1 \\ C_{1,2} \end{bmatrix}, \quad C_2 = \begin{bmatrix} D_2 \\ C_{2,2} \end{bmatrix}, \quad c = \begin{bmatrix} d \\ c_2 \end{bmatrix}
\]  

(4.33)

and that by perturbing \( D_1, C_{1,2}, D_2, C_{2,2}, d, c_2 \) this structure is preserved. The reason for this specific structure becomes more apparent when we try perturbing \( D_1, D_2, d \) while the other parameters remain fixed. This is because a controller is designed to achieve

\[
e^{\alpha t}(D_1x + D_2w + d) \to 0 \text{ as } t \to \infty.
\]  

(4.34)

We notice that if in addition to the above regulation, we also have \( e^{\alpha t}(D_1x + D_2w + (d + \epsilon)) \to 0 \text{ as } t \to \infty \), then it must be that \( e^{\alpha t} \epsilon \to 0 \). This would imply that \( \alpha < 0 \), contradicting our assumption. Also \( e^{\alpha t}(D_1x + (D_2 + \epsilon I)w + d) \to 0 \text{ as } t \to \infty \), would require either \( \alpha < 0 \) or \( w \to 0 \), which breaks Assumption (A2). Therefore we know that we require a specific structure to accommodate for these cases. The above assumption implies that the error signal must be part of the measurements. Now that we have stated our assumptions, we are ready to define our structurally stable output regulation problem.

**Definition 4.3.2.** The system (4.1) with the additional structure given in (4.33) is said to solve the *structurally stable output regulation problem* at the nominal parameters \((A_0, B_0, E_0, a_0, C_{1_0}, C_{2_0}, c_0, D_{1_0}, D_{2_0}, d_0)\) if there exists a fixed controller of the form (4.12) which satisfies the following two properties:

1. The controller solves Problem 4.1.2 with parameters \((A_0, B_0, E_0, a_0, C_{1_0}, C_{2_0}, c_0, D_{1_0}, D_{2_0}, d_0)\).

2. There exists a neighborhood \(P_0\) of \((A_0, B_0, E_0, a_0, C_{1_0}, C_{2_0}, c_0, D_{1_0}, D_{2_0}, d_0)\) such that the controller solves Problem 4.1.2 for each set of perturbed plant parameters in \(P_0\).

A first note in the above definition is that there are no perturbations in the exosystem parameters \((S, s)\). This is due to the fact that the system must have a zero from \( w \) to \( e \) at all frequencies of the exosystem. Therefore, general perturbations of \((S, s)\) will result in a closed-loop system which no longer
achieves output regulation. Because of this, it is standard in the literature to only study perturbations of the plant parameters.

The result below shows that the necessary and sufficient conditions for well-posedness are also necessary and sufficient conditions for the existence of a regulator that solves the structurally stable output regulation problem.

**Theorem 4.3.3.** Suppose Assumption 4.1.2 hold. Then system (4.1) with structural constraint (4.33) solves the structurally stable output regulation problem at \((A_0, B_0, E_0, a_0, C_1, C_2, c_0, D_1, D_2, d_0)\) if and only if the matrix

\[
\begin{bmatrix}
A_0 - \lambda I & B_0 \\
D_{1_0} & 0
\end{bmatrix}
\]

has full row rank for \(\lambda = 0\), and for each \(\lambda\) which is an eigenvalue of \(S\).

The proof of necessity is clear from Theorem 4.3.2. The proof of sufficiency is a consequence of Theorem 4.3.4. Before we present Theorem 4.3.4, we need the following results.

**Lemma 4.3.1 ([14]).** Assume

\[
\begin{bmatrix}
A_0 - \lambda I & B_0 \\
D_{1_0} & 0
\end{bmatrix}
\]

has full row rank for all eigenvalues \(\lambda\) of \(S\). Then there exists a static preliminary feedback \(u = Ne + \tilde{u}\) such that

\[\bar{A} = A_0 + B_0ND_{1_0}\]

has no eigenvalues in common with \(S\).

**Lemma 4.3.2 ([14]).** Suppose the matrix

\[
\sigma \left( \begin{bmatrix}
\bar{A}_0 & B_0H \\
GC_{1_0} & F
\end{bmatrix} \right) \subset \mathbb{C}^-
\]

and \(\bar{A}_0\) and \(S\) have no eigenvalues in common. Then the linear map

\[
\mathcal{A} : \mathbb{R}^{(n_x + p - r) \times q} \rightarrow \mathbb{R}^{n_x \times q}
\]

\[
: \Sigma \rightarrow \mathcal{A}(\Sigma, V) = F\Sigma - \Sigma S + G_2V
\]
has at least $pq$ independent solutions $\Sigma$ of the equation $A(\Sigma, V) = 0$.

**Lemma 4.3.3.** Suppose the matrix

$$\sigma \left( \begin{bmatrix} \bar{A}_0 & \bar{B}_0H \\ GC_{10} & F \end{bmatrix} \right) \subset \mathbb{C}^-$$

and $\bar{A}_0$ has no eigenvalue $\lambda = 0$. Then the map

$$C : \mathbb{R}^{(n_c+p-r)} \to \mathbb{R}^{n_c}$$

$$: \sigma \to C(\sigma, W) = F\sigma - \sigma \cdot 0 + G_2W + G_1d_0 + G_2c_2 + f - \Sigma s$$

has at least $p$ independent solutions $\sigma$ for the equation $C(\sigma, W) = 0$.

**Proof.** Since the matrix

$$\sigma \left( \begin{bmatrix} \bar{A}_0 & \bar{B}_0H \\ GC_{10} & F \end{bmatrix} \right) \subset \mathbb{C}^-$$

there exists a solution $\bar{p}, \bar{\sigma}$, to the equations

$$\begin{bmatrix} \bar{p} \\ \bar{\sigma} \end{bmatrix} \cdot 0 = \begin{bmatrix} \bar{A}_0 & \bar{B}_0H \\ GC_{10} & F \end{bmatrix} \begin{bmatrix} \bar{p} \\ \bar{\sigma} \end{bmatrix} + \begin{bmatrix} \bar{B}_0h + \bar{a}_0 - \Pi s \\ G_1d_0 + G_2c_2 + f - \Sigma s \end{bmatrix}$$

(4.37)

Recall that $y$ is partitioned as $y = (e, y_2)$ and therefore

$$C_{10} = \begin{bmatrix} D_{10} \\ C_{1,2} \end{bmatrix}, \quad C_{20} = \begin{bmatrix} D_{20} \\ C_{2,2} \end{bmatrix}, \quad c_0 = \begin{bmatrix} d_0 \\ c_2 \end{bmatrix}.$$ 

Then (4.37) gives $C(\bar{\sigma}, C_{1,2}p + c_2) = 0$, and hence $\bar{\sigma}$ is a solution to the affine map $C(\sigma, W) = 0$. Define the linear map

$$\bar{C} : \mathbb{R}^{(n_c+p-r)} \to \mathbb{R}^{n_c}$$

$$: \sigma \to \bar{C}(\sigma, W) = F\sigma - \sigma \cdot 0 + G_2W$$

By applying Lemma 4.3.2 with $S = 0$, and hence $q = 1$, there exists $p$ independent solutions $\bar{\sigma}_1, \ldots, \bar{\sigma}_p$ for the equation $\bar{C}(\sigma, W) = 0$. Since $\bar{C}$ is the linearisation of the affine map $C$, for $i = 1, \ldots, p$, $C(\bar{\sigma} + \bar{\sigma}_i, W) = C(\bar{\sigma}, W) + \bar{C}(\bar{\sigma}_i, W) = 0$. Therefore $\bar{\sigma} + \bar{\sigma}_1, \ldots, \bar{\sigma} + \bar{\sigma}_p$ are independent solutions for the equation
\[ C(\sigma, W) = 0. \]

We now present the main result for structural stability.

**Theorem 4.3.4.** Assume the matrix (4.35) has full row rank for \( \lambda = 0 \), and for each \( \lambda \) which is an eigenvalue of \( S \). Then there exists a regulator of the form (4.12) which solves the output regulation problem for the nominal plant parameters \((A_0, B_0, E_0, a_0, C_1, C_2, c_0, D_1, D_2, d_0)\). Moreover, this controller achieves output regulation for each set of perturbed plant parameters \((A, B, E, a, C_1, C_2, c, D_1, D_2, d)\) for which the closed-loop system is \( \alpha \)-internally stable, i.e. when the matrix

\[ \sigma \left( \begin{bmatrix} A & BH \\ GC_1 & F \end{bmatrix} \right) \subset C^-_\alpha. \]  

(4.38)

**Proof.** The proof is done by construction of an appropriate controller. The construction of such a controller is divided into two parts. We first apply a preliminary static output feedback to the plant which guarantees that our system has no poles in common with the exosystem. We then formulate an auxiliary output regulation problem with measurement feedback based on this new system.

It will be shown that a controller which solves output regulation for this auxiliary system, achieves structurally stable output regulation for the original system. By the assumption that matrix (4.35) has full row rank for \( \lambda = 0 \), and for each \( \lambda \) which is an eigenvalue of \( S \), along with applying Lemma 4.3.1 with exosystem matrix

\[
\begin{bmatrix}
S & 0 \\
0 & 0
\end{bmatrix}
\]

there exists a static preliminary feedback \( u = Ne + \tilde{u} \) such that

\[ \tilde{A}_0 = A_0 + B_0ND_{1a} \]

which has no eigenvalue \( \lambda = 0 \), and no eigenvalues in common with \( S \). After applying this preliminary static feedback to the nominal system we obtain the following extended system.

\[
\begin{align*}
\dot{x} &= \tilde{A}_0 x + \tilde{B}_0 \tilde{u} + \tilde{E}_0 w + \tilde{a}_0 \\
y &= C_{1a} x + C_{2a} w + c_0 \\
e &= D_{1a} x + D_{2a} w + d_0.
\end{align*}
\]

(4.39a, 4.39b, 4.39c)

Where \( \tilde{A}_0 = A_0 + B_0ND_{1a} \), \( \tilde{B}_0 = B_0 \), \( \tilde{E}_0 = E_0 + B_0ND_{2a} \), and \( \tilde{a}_0 = a_0 + B_0Nd_0 \). Note that since
$C_{10}, C_{20}, c_0$ remain unchanged, the structure in (4.33) is preserved. Also note that perturbations of the above parameters are basically the same as perturbations of the original system (the shape of the neighborhood changes but not the fact that the neighborhood is open). Therefore, achieving structural stability for the extended system is intrinsically the same problem as achieving structural stability for the original system. We therefore focus the proof on showing structural stability for this extended system.

The next step of our construction involves extending the exosystem. Without loss of generality, assume $S$ and $s$ have been transformed into the form

$$S = \begin{bmatrix} S^* & 0 \\ 0 & S_{\text{min}} \end{bmatrix}, \quad s = \begin{bmatrix} s^* \\ s_{\text{min}} \end{bmatrix}$$

in which $S_{\text{min}}$ is a matrix whose characteristic polynomial coincides with the minimal polynomial of $S$, and $s_{\text{min}}$ is the remaining affine term after the coordinate transformation. Define $\tilde{q}$ such that $S_{\text{min}} \in \mathbb{R}^{\tilde{q} \times \tilde{q}}$. Consider the auxiliary exosystem

$$\dot{\tilde{w}} = \tilde{S}_p \tilde{w} + \tilde{s}_p$$

where $\tilde{w} \in \mathbb{R}^{pq}$ and $\tilde{S}_p$ and $\tilde{s}_p$ are given by

$$\tilde{S}_p = \begin{bmatrix} S_{\text{min}} & 0 & \cdots & 0 \\ 0 & S_{\text{min}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & S_{\text{min}} \end{bmatrix}, \quad \tilde{s}_p = \begin{bmatrix} s_{\text{min}} \\ \vdots \\ s_{\text{min}} \end{bmatrix}.$$

Here the auxiliary exosystem system contains $p$ copies of the minimal representation of $S$, where $p$ is the dimension of $y$. We decompose $\tilde{S}_p$ and $\tilde{s}_p$ as follows:

$$\tilde{S}_p = \begin{bmatrix} \tilde{S}_r & 0 \\ 0 & \tilde{S}_{p-r} \end{bmatrix}, \quad \tilde{s}_p = \begin{bmatrix} \tilde{s}_r \\ \tilde{s}_{p-r} \end{bmatrix},$$

where $r$ is the dimension of the of $e$. Consider the auxiliary system which is made up of the extended
plant and the auxiliary exosystem,

\[
\begin{align*}
\dot{\tilde{x}} &= \tilde{A}_0 \tilde{x} + \tilde{B}_0 \tilde{u} + \tilde{a} \\
\dot{\tilde{w}} &= \tilde{S}_p \tilde{w} + \tilde{s}_p \\
y &= C_1 \tilde{x} + \tilde{C}_2 \tilde{w} + c_0 \\
e &= D_1 \tilde{x} + \tilde{D}_2 \tilde{w} + d_0,
\end{align*}
\]

where the matrix \( \tilde{C}_2 \) and \( \tilde{D}_2 \) are further partitioned as

\[
\tilde{C}_2 = \begin{bmatrix} \tilde{D}_2 \\ \tilde{C}_{2,2} \end{bmatrix} = \begin{bmatrix} \tilde{D}_{2,1} & 0 \\ 0 & \tilde{C}_{2,2,2} \end{bmatrix}
\]

where

\[
\begin{align*}
\tilde{D}_{2,1} &= \begin{bmatrix} \tilde{D}_{2,1,1} & \tilde{D}_{2,1,2} & \cdots & \tilde{D}_{2,1,r} \\ \vdots & \ddots & \ddots & \ddots \\ \tilde{D}_{2,1,r} & \cdots & \tilde{D}_{2,1,1} \\ \end{bmatrix} \\
\tilde{C}_{2,2,2} &= \begin{bmatrix} \tilde{C}_{2,2,2,1} & \tilde{C}_{2,2,2,2} & \cdots & \tilde{C}_{2,2,2,p-r} \\ \end{bmatrix}.
\end{align*}
\]

Here the matrices \( \tilde{D}_{2,1} \) and \( \tilde{C}_{2,2,2} \) are selected so that the pairs of matrices \( (\tilde{D}_{2,1}, \tilde{S}_r) \) and \( (\tilde{C}_{2,2,2}, \tilde{S}_{p-r}) \) are observable. This construction can be done since \( \tilde{S}_r \) and \( \tilde{S}_{p-r} \) have exactly \( r \) and \( p - r \) Jordan blocks for each different eigenvalue respectively. Since \( \tilde{D}_{2,1} \) and \( \tilde{C}_{2,2,2} \) have \( r \) and \( p - r \) rows respectively, we can construct them in such a way to make sure the pairs of matrices \( (\tilde{D}_{2,1}, \tilde{S}_r) \) and \( (\tilde{C}_{2,2,2}, \tilde{S}_{p-r}) \) are observable, and hence \( \alpha \)-detectable.

We now show that Assumption 4.1.1 and Assumption 4.1.2 hold for this auxiliary system, and that the regulator equations for this auxiliary are solvable given that the matrix

\[
\begin{bmatrix} A_0 - \lambda I & B_0 \\ D_1 & 0 \end{bmatrix}
\]

has full row rank for \( \lambda = 0 \), and for each \( \lambda \) which is an eigenvalue of \( S \).

By Lemma 4.1.3, the pair \((A_0, B_0)\) satisfies Assumption 4.1.1(A1). Since \( \tilde{S}_p \) consists of \( p \) copies of the minimal block of \( S \), Assumption 4.1.1(A2) holds. Finally the pair

\[
\begin{bmatrix} C_1 & \tilde{C}_2 \end{bmatrix}, \begin{bmatrix} A_0 & 0 \\ 0 & \tilde{S}_p \end{bmatrix}
\]

(4.44)
satisfy Assumption 4.1.2(A3), since \((C_1, \bar{A}_0)\) and \((\tilde{C}_2, \tilde{S}_p)\) each satisfy Assumption 4.1.2(A3) and \(\bar{A}_0\) and \(S\) have no eigenvalues in common.

Now if we partition \(\Pi\) and \(\Gamma\) as

\[
\Pi = \begin{bmatrix} \Pi_1 & \cdots & \Pi_p \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \Gamma_1 & \cdots & \Gamma_p \end{bmatrix}
\]

we can rewrite our affine FBI equations into the form, for \(i = 1, \ldots, p\)

\[
\Pi_i S_{\min} = \bar{A}_0 \Pi_i + \bar{B}_0 \Gamma_i \tag{4.45}
\]

\[
\Pi_i s_{\min} = \bar{A}_0 p + \bar{B}_0 \gamma + \bar{a}_0 \tag{4.46}
\]

\[
0 = D_{1i} \Pi_i + \bar{D}_{2,1,i} \tag{4.47}
\]

\[
0 = D_{1i} p + \bar{a}_0 \tag{4.48}
\]

Since the matrix in (4.43) has full row-rank for \(\lambda = 0\) and each \(\lambda\) which is an eigenvalue of \(S\), and thus \(S_{\min}\), by Theorem 4.3.2, Problem 4.1.2 is solvable. Therefore, a controller constructed as in Theorem 4.1.4 will achieve output regulation for the auxiliary plant.

We now show that any controller of the form (4.12) which solves the output regulation problem for the auxiliary plant (4.42) solves the structurally stable output regulation problem for the system (4.39). As in the the proof of Theorem 4.1.4, we can conclude that the proposed controller indeed solves the structurally stable output regulation problem for the system (4.39) if the equations

\[
\Pi S = \bar{A} \Pi + \bar{B} H \Sigma + \bar{E} \tag{4.49a}
\]

\[
\Sigma S = GC_1 \Pi + F \Sigma + GC_2 \tag{4.49b}
\]

\[
p \cdot 0 = \bar{A} p + \bar{B} H \sigma + \bar{B} h + \bar{a} - \Pi s \tag{4.49c}
\]

\[
\sigma \cdot 0 = GC_1 p + F \sigma + Gc + f - \Sigma s \tag{4.49d}
\]

\[
0 = D_1 \Pi + D_2 \tag{4.49e}
\]

\[
0 = D_1 p + d \tag{4.49f}
\]

has a solution for each set of perturbed parameters \((\bar{A}, \bar{B}, \bar{E}, \bar{a}, C_1, C_2, c, D_1, D_2, d)\), for which the closed-
loop system is $\alpha$-internally stable, i.e. when the matrix
\[
\sigma \begin{bmatrix}
\bar{A} & BH \\
GC_1 & F
\end{bmatrix} \subset \mathbb{C}_\alpha^{-}.
\] (4.50)

Just as in the proof of Theorem 4.1.4, we have that equations (4.49a)-(4.49d) are solvable for any of the perturbed parameters. If we prove that the solution $(\Pi, \Sigma, p, \sigma)$ also satisfy (4.49e)-(4.49f) then we have satisfied our structurally stable output regulation problem. We define the following linear mappings:

\begin{align*}
\mathcal{A} & : \mathbb{R}^{(n_c+p-r)\times q} \rightarrow \mathbb{R}^{n_c\times q} \\
& : \Sigma \rightarrow \mathcal{A}(\Sigma, V) = F\Sigma - \Sigma S + G_2 V \\
\mathcal{B} & : \mathbb{R}^{r\times q} \rightarrow \mathbb{R}^{n_c\times q} \\
& : Z \rightarrow \mathcal{B}(Z) = G_1 Z \\
\mathcal{C} & : \mathbb{R}^{(n_c+p-r)} \rightarrow \mathbb{R}^{n_c} \\
& : \sigma \rightarrow \mathcal{C}(\sigma, W) = F\sigma - \sigma \cdot 0 + G_2 W + f - \Sigma s \\
\mathcal{D} & : \mathbb{R}^{r} \rightarrow \mathbb{R}^{n_c} \\
& : T \rightarrow \mathcal{D}(T) = G_1 T
\end{align*}

where $n_c$ is the order of the controller and $G = \begin{bmatrix} G_1 & G_2 \end{bmatrix}$ is partitioned just as $y$ is partitioned into $e$ and $y_2$. With this notation, equation (4.49b) and (4.49d) can be rewritten as

\begin{align*}
\mathcal{A}(\Sigma, C_1, \Sigma + C_2) & = -\mathcal{B}(D_1 \Pi + D_2) \\
\mathcal{C}(\sigma, C_1, p + c_2) & = -\mathcal{D}(D_1 p + d)
\end{align*}

(4.51)  (4.52)

respectively. If we can prove that the images of $\mathcal{A}$ and $\mathcal{B}$ intersect at $\{0\}$, and also that the images of $\mathcal{C}$ and $\mathcal{D}$ intersect at $\{0\}$. Then we would have that

\begin{align*}
\mathcal{A}(\Sigma, C_1, \Sigma + C_2) & = 0 \\
\mathcal{B}(D_1 \Pi + D_2) & = 0 \\
\mathcal{C}(\sigma, C_1, p + c_2) & = 0 \\
\mathcal{D}(D_1 p + d) & = 0.
\end{align*}

(4.53)  (4.54)  (4.55)  (4.56)

If we also show that $\text{Ker} \mathcal{B} = \{0\}$ and $\text{Ker} \mathcal{D} = \{0\}$, then equations (4.49e) and (4.49f) hold. Therefore
if we can show
\[
\text{Im } A \cap \text{Im } B = \{0\}, \tag{4.57}
\]
\[
\text{Ker } B = \{0\}, \tag{4.58}
\]
\[
\text{Im } C \cap \text{Im } D = \{0\}, \tag{4.59}
\]
\[
\text{Ker } D = \{0\}, \tag{4.60}
\]
we have proven that the proposed regulator does indeed solve the structurally stable output regulation problem. Recall that
\[
\sigma \left( \begin{bmatrix} \bar{A}_0 & \bar{B}_0 H \\ GC_{10} & F \end{bmatrix} \right) \subset \mathbb{C}^{-} \subset \mathbb{C}^{-}. \tag{4.61}
\]
Also, by construction of \( \bar{A}_0 \), \( \bar{A}_0 \) has no eigenvalue \( \lambda = 0 \) and no eigenvalues in common with \( S \). By Lemma 4.3.2 there exists at least \( pq \) independent solutions \( \Sigma \) of the equation \( A(\Sigma, V) = 0 \). By Lemma 4.3.3 there exists at least \( p \) independent solutions \( \sigma \) of the equation \( C(\sigma, W) = 0 \). This implies that the dimension of \( \text{Ker } A \) is at least \( pq \), and the dimension of \( \text{Ker } C \) is at least \( p \). Therefore by the rank-nullity theorem
\[
\dim \text{Im } A \leq (n_c - r)q
\]
\[
\dim \text{Im } C \leq n_c - r.
\]
Also, since the dimension of the image of a linear map cannot exceed its domain, we have that
\[
\dim \text{Im } B \leq rq
\]
\[
\dim \text{Im } D \leq r.
\]
Consider the Sylvester equations
\[
\begin{bmatrix} \bar{A} & \bar{B} H \\ GC_1 & F \end{bmatrix} \begin{bmatrix} \Pi \\ \Sigma \end{bmatrix} - \begin{bmatrix} \Pi \\ \Sigma \end{bmatrix} S = \begin{bmatrix} X \\ Y \end{bmatrix}
\]
\[
\begin{bmatrix} \bar{A} & \bar{B} H \\ GC_1 & F \end{bmatrix} \begin{bmatrix} p \\ \sigma \end{bmatrix} - \begin{bmatrix} p \\ \sigma \end{bmatrix} \cdot 0 = \begin{bmatrix} V \\ W \end{bmatrix}
\]
and use any arbitrary $X,Y,V,W$, and denote the corresponding solution by $\Pi, \Sigma, p, \sigma$, and note that by construction

$$GC_1\Pi + F\Sigma - \Sigma S = Y$$
$$GC_1p + F\sigma - \sigma \cdot 0 + f - \Sigma s = W$$

That is,

$$\mathcal{A}(\Sigma, C_{1,2}\Pi) + B(D_1\Pi) = Y \quad (4.61)$$
$$\mathcal{C}(\sigma, C_{1,2}p) + D(D_1p) = W \quad (4.62)$$

The arbitrariness of $Y$ and $W$ shows that

$$\text{Im } \mathcal{A} + \text{Im } \mathcal{B} = \mathbb{R}^{n_c \times q}$$
$$\text{Im } \mathcal{C} + \text{Im } \mathcal{D} = \mathbb{R}^{n_c \times 1}$$

Therefore we have that

$$\dim \text{Im } \mathcal{A} = (n_c - r)q$$
$$\dim \text{Im } \mathcal{B} = rq$$
$$\dim \text{Im } \mathcal{C} = n_c - r$$
$$\dim \text{Im } \mathcal{D} = r$$

These relations prove that

$$\text{Im } \mathcal{A} \cap \text{Im } \mathcal{B} = \{0\}, \quad (4.63)$$
$$\text{Ker } \mathcal{B} = \{0\}, \quad (4.64)$$
$$\text{Im } \mathcal{C} \cap \text{Im } \mathcal{D} = \{0\}, \quad (4.65)$$
$$\text{Ker } \mathcal{D} = \{0\}, \quad (4.66)$$

hold and therefore that the controller achieves structurally stable output regulation for the system auxiliary system, and hence also achieves structurally stable output regulation for (4.39). □
Chapter 5

Output Reach Control Problem

In this chapter we formulate and solve the ORCP by using methods inspired by viability theory [15, 21]. The main idea is to construct a maximal (or approximately maximal) polytope \( P \) in the state space and an associated piecewise affine controller defined on \( P \), such that the closed-loop system solves the ORCP (when solutions are projected to the output space). A procedure to construct piecewise affine controllers in the full state space on a polytope is available in [8]. Therefore, we do not focus on the control synthesis aspect of the problem. Rather, we focus our attention on the necessary conditions, called invariance conditions, that make the procedure of [8] feasible.

Our methodology to find a (close to) maximal polytope in the state space on which the invariance conditions are solvable is to adapt the algorithm in [23] for finding maximal positively invariant polytopes under linear dynamics. The main idea of our adaptation is that, instead of beginning with an initial positively invariant polytope, we begin with an exit facet in the state space that is determined by the exit facet in the output space of the ORCP that we want to solve. Then we iteratively “grow” a polytope such that, at each iteration, the invariance conditions are solvable for the current iterate.

5.1 Problem Statement

Consider a \( p \)-dimensional output simplex \( S := \text{co} \{v_0, \ldots, v_p\} \subset \mathbb{R}^p \), the convex hull of \( p + 1 \) affinely independent points \( v_i \in \mathbb{R}^p, i = 0, \ldots, p \). Let its vertex set be \( V := \{v_0, \ldots, v_p\} \) and its facets \( \mathcal{F}_0, \ldots, \mathcal{F}_p \). The facet is indexed by the vertex it does not contain. Let \( h_j \in \mathbb{R}^p, j \in \{0, \ldots, p\} \), be the unit normal vector to each facet \( \mathcal{F}_j \) pointing outside of the simplex. Facet \( \mathcal{F}_0 \) is called the exit facet. Let \( I := \{1, \ldots, p\} \) and define \( I(x) \) to be the minimal index set among \( \{0, \ldots, p\} \) such that \( x \in \text{co} \{v_i \mid i \in I(x)\} \).
We consider the affine control system

\[ \dot{x} = Ax + Bu + a \]
\[ y = Cx, \]

where \( A \in \mathbb{R}^{n \times n}, \ a \in \mathbb{R}^n, \ B \in \mathbb{R}^{n \times m}, \ C \in \mathbb{R}^{p \times n}, \ \text{rank}(B) = m, \ \text{and rank}(C) = p. \) Let \( \mathcal{B} = \text{Im}(B) \), the image of \( B \). Note that using an output which is a linear function of the state rather than an affine function \( y = Cx + c, \ c \in \mathbb{R}^p \), is no loss of generality because one can always translate the origin of the output space to convert an affine function to a linear function. Let \( \phi(t, x_0) \) denote the trajectory of (5.1) starting at \( x_0 \) under some control law \( u \). Let \( y(t, x_0) := C\phi(t, x_0) \) be the output trajectory corresponding to \( \phi(t, x_0) \). We are interested in studying reachability by the system output of the exit facet \( \mathcal{F}_0 \) of \( \mathcal{S} \). 

**Problem 5.1.1 (Output Reach Control Problem (ORCP)).** Consider system (5.1)-(5.2) and the output simplex \( \mathcal{S} \subset \mathbb{R}^p \). Find a state feedback \( u = f(x) \) such that for each initial condition \( x_0 \in \mathbb{R}^n \) with \( Cx_0 \in \mathcal{S} \), there exist \( T \geq 0 \) and \( \gamma > 0 \) such that

(i) \( y(t, x_0) \in \mathcal{S} \) for all \( t \in [0, T] \);

(ii) \( y(T, x_0) \in \mathcal{F}_0 \); and

(iii) \( y(t, x_0) \notin \mathcal{S} \) for all \( t \in (T, T + \gamma) \).

The problem formulation of the ORCP differs from the standard RCP in that conditions (i)-(iii) are normally imposed on the state trajectory \( \phi(\cdot, x_0) \), whereas here they are stated in terms of the output trajectory \( y(\cdot, x_0) \).

While the ORCP stated is ultimately the problem we would like to solve, in order to use the existing RCP literature, we must bound the state space to a polytope. Since, in general, the states \( x_0 \in \mathbb{R}^n \) such that \( Cx_0 \in \mathcal{S} \) form a polyhedron, we further restrict these state to form a (bounded) polytope. Therefore, we pose a related, but modified version of the ORCP.

**Problem 5.1.2.** Consider system (5.1)-(5.2) and the output simplex \( \mathcal{S} \subset \mathbb{R}^p \). Find a state feedback \( u = f(x) \) and a polytope \( \mathcal{P} \subset \mathbb{R}^n \) such that for each initial condition \( x_0 \in \mathcal{P} \), there exist \( T \geq 0 \) and \( \gamma > 0 \) such that

(i) \( y(t, x_0) \in \mathcal{S} \) for all \( t \in [0, T] \);

(ii) \( y(T, x_0) \in \mathcal{F}_0 \); and

(iii) \( y(t, x_0) \notin \mathcal{S} \) for all \( t \in (T, T + \gamma) \).
5.2 From ORCP to RCP

In this section we develop our method to solve Problem 5.1.2 with attention on how to find the polytope \( \mathcal{P} \). The main challenge in solving Problem 5.1.2 is that it is formulated in the output space so there are no explicit constraints on the full state vector, unlike the standard RCP. We seek to impose extra constraints on the states in order to guarantee that the evolution of the output meets the requirements of Problem 5.1.2. Additionally, we hope to leverage the existing theoretical tools for solving the standard RCP, since there is now a substantial literature available [1], [16], [2], [17], [7], [8], [9], [10]. In essence, Problem 5.1.2 will be lifted to the state space, additional constraints will be imposed on the states, a feasible state set will be constructed, and finally, we invoke the standard RCP on that feasible state set. If our procedure works correctly, then the solution of the standard RCP will result in a solution of Problem 5.1.2 on \( \mathcal{S} \).

The main ideas of our methodology are as follows. We begin by constructing a polytope \( \mathcal{P} \subset \mathbb{R}^n \) with the property that if the initial state \( x_0 \) satisfies \( x_0 \in \mathcal{P} \), then \( y_0 := Cx_0 \in \mathcal{S} \). The polytope \( \mathcal{P} \) is constructed by first “lifting” \( \mathcal{S} \) into the state space to create an (unbounded) \( n \)-dimensional polyhedron, and second imposing additional state constraints to ensure that \( \mathcal{P} \) is bounded, i.e. it is an \( n \)-dimensional polytope. Existing theory of polyhedra tells us that the lift of the exit facet of \( \mathcal{S} \) to \( \mathcal{P} \) is again a facet of \( \mathcal{P} \), so we show this lifted exit facet can serve as the exit facet for \( \mathcal{P} \).

Once the initial condition set \( \mathcal{P} \) has been formed and an exit facet identified, one would like to solve a standard RCP on \( \mathcal{P} \). It is well-known that a necessary condition for solvability of the RCP is that the so-called invariance conditions are solvable [8]. Unfortunately, these conditions are not guaranteed to be solvable on \( \mathcal{P} \). Thus, we invoke viability theory [15], [21] to construct (a polytopic estimate) of the largest subset of \( \mathcal{P} \) on which the invariance conditions are solvable. An algorithm inspired by [23] is proposed with the crucial property that any intermediate solution of the algorithm \( \mathcal{P}^k \) includes the exit facet of \( \mathcal{P} \), and moreover, the invariance conditions are solvable on \( \mathcal{P}^k \). This implies that the algorithm can be terminated after any number of iterations to obtain an estimate \( \mathcal{P}^j \) of \( \mathcal{P}^* \), the largest subset of \( \mathcal{P} \) containing the exit facet and such that the invariance conditions are solvable. The differences between the usual application of viability theory and our approach for reach control will be highlighted in the sequel. Finally, it is worth pointing out that our algorithm can be applied to any instance of the standard RCP on a polytope when the invariance conditions do not hold a priori, not only problems originating from Problem 5.1.2.
5.2.1 Computing $\mathcal{P}$

In this section we develop our method to construct the initial polytope $\mathcal{P}$. It consists of two steps: first, lift the output simplex $\mathcal{S} \subset \mathbb{R}^p$ into the state space $\mathbb{R}^n$; second, impose additional constraints on the states so that the resulting set is a bounded polytope. Define the output map

$$y(x) := Cx.$$ 

Also, for $\mathcal{V} \subset \mathbb{R}^n$, let $y(\mathcal{V}) := \{Cx \mid x \in \mathcal{V}\}$, and for $\mathcal{W} \subset \mathbb{R}^p$, let $y^{-1}(\mathcal{V}) := \{x \in \mathbb{R}^n \mid Cx \in \mathcal{V}\}$. The lift of $\mathcal{S}$ to $\mathbb{R}^n$ is defined to be $y^{-1}(\mathcal{S})$. It is easily shown that $y^{-1}(\mathcal{S})$ is an $n$-dimensional polyhedron. To convert it to a (bounded) $n$-dimensional polytope we must impose additional constraints on the states, particularly the states in Ker $C$. To that end, let $\mathbb{R}^n = \text{Im} C^T \oplus \text{Ker} C$. Also let $C' \in \mathbb{R}^{n \times p}$ be a maximal rank solution of $CC' = 0$. Define the coordinate transformation $x = T\hat{x}$, where $T = \begin{bmatrix} C^T & C' \end{bmatrix}$.

Let $\hat{x} := (\hat{x}_1, \hat{x}_2)$ where $\hat{x}_1 \in \mathbb{R}^p$ and $\hat{x}_2 \in \mathbb{R}^{n-p}$. Let $a, b \in \mathbb{R}^{n-p}$ with $a < b$. Define

$$\mathcal{P}_{\text{box}} := \{x = T\hat{x} \mid a \preceq \hat{x}_2 \preceq b\}.$$ 

Then we define

$$\mathcal{P} := y^{-1}(\mathcal{S}) \cap \mathcal{P}_{\text{box}}. \quad (5.3)$$

**Lemma 5.2.1.** $\mathcal{P}$ is an $n$-dimensional polytope.

**Proof.** First, since $\text{rank}(C) = \text{rank}(C^T) = p$, $\text{rank}(CC^T) = p$ and $(CC^T)^{-1}$ exists. Define the sets

$$\tilde{\mathcal{W}}_1 := \{\hat{x}_1 \in \mathbb{R}^p \mid \hat{x}_1 = (CC^T)^{-1}y, y \in \mathcal{S}\}$$

$$\tilde{\mathcal{W}}_2 := \{\hat{x}_2 \in \mathbb{R}^{n-p} \mid a \preceq \hat{x}_2 \preceq b\}.$$ 

The set $\tilde{\mathcal{W}}_1$ is compact since $\mathcal{S}$ is compact, and $\tilde{\mathcal{W}}_2$ is compact by construction. Hence, $\tilde{\mathcal{W}}_1 \times \tilde{\mathcal{W}}_2$ is compact. We claim that

$$\mathcal{P} = \{x = T\hat{x} \mid \hat{x} = (\hat{x}_1, \hat{x}_2) \in \tilde{\mathcal{W}}_1 \times \tilde{\mathcal{W}}_2\}. \quad (5.4)$$

To prove the claim, let $x \in \mathcal{P}$. By definition of $\mathcal{P}$, $y = Cx \in \mathcal{S}$. Then $y = CT\hat{x} = CC^T\hat{x}_1 + CC'\hat{x}_2$. But $CC' = 0$, so $y = CC^T\hat{x}_1$ and $\hat{x}_1 = (CC^T)^{-1}y$. Since $y \in \mathcal{S}$, we conclude $\hat{x}_1 \in \tilde{\mathcal{W}}_1$. Also since $x \in \mathcal{P}_{\text{box}}, a \preceq \hat{x}_2 \preceq b$, so $\hat{x}_2 \in \tilde{\mathcal{W}}_2$. We conclude $\hat{x} \in \tilde{\mathcal{W}}_1 \times \tilde{\mathcal{W}}_2$, as desired. Conversely, suppose $x = T\hat{x}$ with $\hat{x} \in \tilde{\mathcal{W}}_1 \times \tilde{\mathcal{W}}_2$. Then $a \preceq \hat{x}_2 \preceq b$, so $x \in \mathcal{P}_{\text{box}}$. Also since $\hat{x}_1 \in \tilde{\mathcal{W}}_1, y = Cx = CC^T\hat{x}_1 \in \mathcal{S}$, so $x \in y^{-1}(\mathcal{S})$. We conclude $x \in \mathcal{P} = y^{-1}(\mathcal{S}) \cap \mathcal{P}_{\text{box}}$. 

Chapter 5. Output Reach Control Problem

Lemma 5.2.3. Let $\mathcal{P}$ be a projection of polytopes $f : \mathcal{P} \to \mathcal{P}'$ be a projection of polytopes. Then for every face $\mathcal{F}'$ of $\mathcal{P}'$, the preimage $f^{-1}(\mathcal{F}') = \{ x \in \mathcal{P} | f(x) \in \mathcal{F}' \}$ is a face of $\mathcal{P}$.

The next result shows that the linear map $y : \mathbb{R}^n \to \mathbb{R}^p$ is a projection of polytopes.

**Definition 5.2.1.** A *projection of polytopes* $f : \mathcal{P} \to \mathcal{P}'$ is an affine map $f : \mathbb{R}^n \to \mathbb{R}^p$, where $\mathcal{P} \subseteq \mathbb{R}^n$ is an $n$-dimensional polytope, $\mathcal{P}' \subseteq \mathbb{R}^p$ is a $p$-dimensional polytope, and $f(\mathcal{P}) = \mathcal{P}'$.

**Lemma 5.2.2.** [31] Let $f : \mathcal{P} \to \mathcal{P}'$ be a projection of polytopes. Then for every face $\mathcal{F}'$ of $\mathcal{P}'$, the preimage $f^{-1}(\mathcal{F}') = \{ x \in \mathcal{P} | f(x) \in \mathcal{F}' \}$ is a face of $\mathcal{P}$.

Lemma 5.2.3. Let $\mathcal{S} \subseteq \mathbb{R}^p$ and $\mathcal{P} = y^{-1}(\mathcal{S}) \cap \mathcal{P}_{box}$. Then $y(\mathcal{P}) = \mathcal{S}$.
Proof. Let $\bar{y} \in y(P)$. By the definition of $P$, $\bar{y} \in y(y^{-1}(S))$. Since $y : \mathbb{R}^n \to \mathbb{R}^p$ is surjective, $y(y^{-1}(S)) = S$, and therefore $\bar{y} \in S$. Conversely, let $\bar{y} \in S$. Select any $\bar{x}_2 \in \mathbb{R}^{n-p}$ such that $a \leq \bar{x}_2 \leq b$ and define $\bar{x} := ((CC^T)^{-1}\bar{y}, \bar{x}_2)$. By the proof of Lemma 5.2.1, $\bar{x} \in \tilde{W}_1 \times \tilde{W}_2$, and therefore $x := T\bar{x} \in P$. This leads to $Cx = C(C^Tx_1 + C'\bar{x}_2) = (CC^T)(CC^T)^{-1}\bar{y} = \bar{y} \in y(P)$, as desired. \hfill $\square$

Using the previous two lemmas we can now define a feasible exit facet of $P$ by lifting $F_0$, the exit facet of $S$:

$$F_0^P := y^{-1}(F_0) \cap P. \tag{5.6}$$

Lemma 5.2.4. $F_0^P$ is a facet of $P$ with outward normal vector given by $h_0^P = \frac{CTh_0}{||CTh_0||}$.

Proof. The exit facet of $S$ is $F_0 = \text{co}\{v_1, \ldots, v_p\}$. Define $\tilde{v}_i := (CC^T)^{-1}v_i$, $i = 1, \ldots, p$. By Lemma 5.2.2, $F_0^P$ is a face of $P$. We show it is a facet of $P$ by showing $F_0^P$ is $(n-1)$-dimensional. To that end, let $w_i := Tw_i$, $i = 1, \ldots, n$, with $w_i$ given in the proof of Lemma 5.2.1. We will show that $w_i \in F_0^P$, $i = 1, \ldots, n$. Then since $\{\tilde{w}_1, \ldots, \tilde{w}_n\}$ are affinely independent, so are $\{w_1, \ldots, w_n\}$, so we conclude $F_0^P$ is $(n-1)$-dimensional. To show $w_i \in F_0^P$, first recall from the proof of Lemma 5.2.1 that $\tilde{w}_i \in \tilde{W}_1 \times \tilde{W}_2$. Then by (5.4), $w_i = Tw_i \in P$, $i = 1, \ldots, n$. Second, we have $Cw_i = CT\tilde{w}_i = CCT\tilde{v}_i = v_i \in F_0$, for $i = 1, \ldots, n$. Thus, $w_i \in y^{-1}(F_0)$. We conclude $w_i \in P \cap y^{-1}(F_0) = F_0^P$, as desired.

Next, we assume without loss of generality (w.l.o.g.) that $0 \in F_0$, so $0 \in y^{-1}(F_0)$. We observe that if $x \in y^{-1}(F_0)$, then $y = Cx \in F_0$, so $h_0 \cdot Cx = 0$. Equivalently, $(CTh_0) \cdot x = 0$. Now $\text{Ker}(CT) \neq \{0\}$ since $\text{rank}(CT) = p$, so $h_0 \neq 0$ implies $h_0 \notin \text{Ker} CT$. Hence $CTh_0 \neq 0$ so the unit vector $h_0^P = \frac{CTh_0}{||CTh_0||}$ is well-defined. Finally, it is easy to show since $h_0$ is the outward normal vector of $F_0$, then $h_0^P$ is also the outward normal vector of $F_0^P$. \hfill $\square$

We have now constructed an $n$-dimensional polytope $P$ and an appropriate exit facet $F_0^P$ which consistently lift the requirements of Problem 5.1.2 on the output simplex $S$ up to the full state space. The next step is to solve the standard RCP on this polytope and show that the solution of the RCP on $P$ results in solving Problem 5.1.2 on $S$. Unfortunately, our work is not complete because to solve the standard RCP, it is necessary that so-called invariance conditions are solvable on $P$ [7, 9]. In the next section we address this gap by proposing a viability algorithm introduced in [23] but adapted to the RCP to help ensure the invariance conditions are met on a possibly smaller polytope in $P$. 


5.3 Viable Polytope for Reach Control

In this section we present an algorithm that provides a polytopic under-approximation of the original state space polytope $\mathcal{P}$ given in (5.3) such that the under-approximation satisfies the invariance conditions associated with the RCP. There are two additional considerations to address, beyond satisfaction of the invariance conditions. First, the exit facet for each iterate $\mathcal{P}^k$ must be well-defined to guarantee that solutions of the affine system under a suitable feedback do indeed exit through the given exit facet $\mathcal{F}_0^\mathcal{P}$. Second, it is necessary to introduce a bound on the control inputs to make the algorithm computationally tractable. We begin this section by stating the invariance conditions. Then we define the notion of an exit set in $\mathcal{F}_0^\mathcal{P}$. This exit set will become the first iterate $\mathcal{P}^0$ of the algorithm. Finally, we present the algorithm and give comparisons to a standard viability algorithm.

We consider an $n$-dimensional polytope

$$\mathcal{P} := \text{co}\{p_0, \ldots, p_r\}$$

with vertex set $V^\mathcal{P} := \{p_0, \ldots, p_r\}$ and facets $\mathcal{F}_0^\mathcal{P}, \ldots, \mathcal{F}_q^\mathcal{P}$, where $\mathcal{F}_0^\mathcal{P}$ is the exit facet. Let $h_j^\mathcal{P}$ be the unit normal to each facet $\mathcal{F}_j^\mathcal{P}$ pointing outside the polytope. Define the index set $J = \{1, \ldots, q\}$. For each $x \in \mathcal{P}$ define the closed, convex cone

$$\mathcal{C}(x) := \{y \in \mathbb{R}^n | h_j^\mathcal{P} \cdot y \leq 0, j \in J \text{ s.t. } x \in \mathcal{F}_j^\mathcal{P}\}.$$  \hfill (5.7)

Next we introduce a bound on the control inputs. To that end, we define $U = \text{co}\{u_1, \ldots, u_M\}$ to be a polytope that bounds the inputs. The invariance conditions will be stated in terms of this bound.

**Definition 5.3.1.** We say the *invariance conditions are solvable* on a polytope $\mathcal{P}$ if for each $v \in V^\mathcal{P}$ there exists $u \in U$ such that

$$Av + Bu + a \in \mathcal{C}(v).$$  \hfill (5.8)

Next we define the exit set of an exit facet. The exit set can be thought of as the set of points on the exit facet such that there exists a velocity vector of the affine system to force trajectories to immediately leave the polytope.

**Definition 5.3.2** (Exit Set). Consider the affine system (5.1) on a polytope $\mathcal{P}$. Let $\mathcal{F}_0^\mathcal{P}$ be a facet of
The exit set of $\mathcal{F}_0^P$ is the set

$$\mathcal{F}_{exit}^P := \text{cl}\{ x \in \mathcal{F}_0^P \mid (\exists u \in U) \ h_0^P \cdot (Ax + Bu + a) > 0 \}.$$ 

There are three pathologies which can arise with $\mathcal{F}_{exit}^P$. First it may be empty. In that case, the RCP is not solvable, and there is no point to proceed with the algorithm. Second, $\mathcal{F}_{exit}^P$ may not be a full dimensional facet of $\mathcal{P}$. But this is impossible as $\mathcal{F}_{exit}^P$ has been constructed as the closure of an open subset in $\mathcal{F}_0^P$. Finally, $\mathcal{F}_{exit}^P$ may not be a polytopic set, a requirement for the algorithm. This is resolved by finding any polytopic under-approximation of $\mathcal{F}_{exit}^P$. We omit this step and we assume that $\mathcal{F}_{exit}^P$ is already presented as an $(n-1)$-dimensional polytope

$$\mathcal{P}^0 := \mathcal{F}_{exit}^P = \text{co} \{ v \in V^0 \},$$

where $V^0$ is the vertex set of $\mathcal{P}^0$. By construction $\mathcal{P}^0$ is an $n-1$ dimensional polytope in the exit set of $\mathcal{F}_0^P$.

Now we introduce the notation of the algorithm. We use $\mathcal{P}^0$ to designate the initial polytope, as defined above. The polytope at iteration $i$ is $\mathcal{P}^i$ and its vertex set is $V^i$. The polytope $\mathcal{P}$ given by (5.3) is the largest possible polytope that the algorithm could construct. The vertex set $D$ bookkeeps the vertices in $V^i$ not yet used in the $i$th iteration.

The algorithm attempts to find the largest polytope $\mathcal{P}^i \subset \mathcal{P}$ such that the invariance conditions hold on $\mathcal{P}^i$. At each iteration, an optimization problem is solved. The objective of the optimization problem is to take a vertex $v$ of $\mathcal{P}^i$, and extend the polytope $\mathcal{P}^i$ by adding new vertices along the rays between $v$ and each of the vertices of $\mathcal{P}$. This creates a new candidate polytope which consists of the convex hull of the new vertices and the old vertices of $\mathcal{P}^i$, except for $v$. The constraint of the optimization problem is that the new polytope must satisfy the invariance conditions. This process continues until the current polytope can no longer be enlarged, while satisfying the invariance conditions.

Algorithm 5.3.1 (Viable Polytope for Reach Control).

1. Initialization:
   $$\mathcal{P} = \text{co} \{ p_0, \ldots, p_r \}; \quad i = 0;$$
   $$V^0 = \{ p_0^0, \ldots \}; \quad \mathcal{P}^0 = \text{co} \{ v \in V^0 \}; \quad D := V^0.$$ 

2. If $D = \emptyset$, end with $\mathcal{P}^i$.

3. If $D \neq \emptyset$, select $v \in D$. 
4. Solve the optimization problem:
\[
\arg\min_{w_j \in co\{v, p_j\}} \sum_{j=0}^{r} ||w_j - p_j||_2 \\
\text{subject to: (5.8) hold for the polytope } P \text{ with vertex set } V := V^i \cup \{w_0, \ldots, w_r\} \setminus \{v\}.
\]

5. If \( \overline{P} \neq P\) and \(V^{i+1} = V\); \(D = V^{i+1}\):
   \(i = i + 1\). Return to step 2.

6. If \( \overline{P} = P\) and \(D = D \setminus \{v\}\). Return to step 2.

Remark 5.3.2. The procedures in Sections 5.2 and 5.3 are not complete. It is possible that no polytope in the \(n\)-dimensional space can be found. Furthermore, there is no guarantee that, even if a full dimensional polytope is discovered, that the RCP will be solvable on that polytope.

The Algorithm 5.3.1 is an adaptation of the algorithm originally given in Gao and Lygeros [23]. The purpose of the algorithm in [23] was to find a polytopic under-approximation \(P'\) of an initial polytope \(P\) such that \(P'\) is positively invariant under given linear dynamics \(\dot{x} = Ax + Bu\). Instead, our objective is to obtain a polytopic under-approximation \(P'\) of a given polytope \(P\) such that the invariance conditions (5.8) hold on \(P'\) for the system (5.1).

A second difference between the two algorithms is that the algorithm in [23] starts with an initial positively invariant polytopic set, and the algorithm iteratively “grows” this set, inside the maximal state space \(P\). In contrast, our algorithm starts with an exit facet of a polytope \(P\), and iteratively grows the polytopic set, with each iterate containing the exit facet, as guaranteed by Proposition 5.3.1.

Mathematically, the difference in the two algorithms is that Lygeros works with the Bouligand tangent cone at the vertices of the polytopes, while we use the cones (5.7) at the vertices. The difference between tangent cones and our cones arises only at vertices on the exit facet.

Turning to the algorithm, the distinction between our version and that of Lygeros is seen in Step 4, where we impose that (5.8) must hold on the polytope \(\overline{P}\).

Note that while the problem has a linear objective function, the constraints are bilinear. Although existing algorithms can be applied to this optimization problem [42], the problem involves converting the polytope \(P^i\) from the \(V\)-representation to the \(H\)-representation (for solving invariance conditions), which can be computationally demanding. What gives this algorithm promise is that not only do we have a viable polytope at each iteration, we also have that the polytopes are non-decreasing in size with each iteration. The following proposition captures the salient properties of the algorithm that we inherit directly from [23].

Proposition 5.3.1 (Prop. 2 of [23]). Each \(P^i\) generated by the algorithm satisfies its invariance conditions. Moreover, \(P^i \subseteq P^{i+1} \subseteq P\).
We also require some further properties which are specific to our problem. These are summarized in the next lemma. The first property guarantees that the ORCP will still be solvable using $P^i$ rather than the original $P$. The second property guarantees that each $P^i$ has a well-defined exit facet.

**Lemma 5.3.1.** Suppose that each $P^i$ generated by the algorithm is full dimensional. Then for each such $P^i$,

(i) $P^i \subset y^{-1}(S)$.

(ii) $F_{P^i}^0 := y^{-1}(F_0) \cap P^i$ is the exit facet of $P^i$.

**Proof.**

(i) By construction, $P \subset y^{-1}(S)$ and by Proposition 5.3.1, $P^i \subset P$. Hence, $P^i \subset y^{-1}(S)$.

(ii) We need only show that $F_{P^i}^0$ is a facet of $P^i$. The fact that it supplies a consistent exit facet for the ORCP is treated in the next section. Recall from Lemma 5.2.4 that $F_{P^i}^0 = y^{-1}(F_0) \cap P$ is a facet of $P$. Let $H$ be the hyperplane in $\mathbb{R}^n$ that contains $F_{P^i}^0$. Since $P^i \subset P$, $P^i$ lies in the closed half-space bounded by $H$ and containing $P$. By construction $P^0 \subset y^{-1}(F_0)$ and by Proposition 5.3.1, $P^0 \subset P$ and $P^0 \subset P^i$. Thus, $P^0 \subset y^{-1}(F_0) \cap P = F_{P^i}^0 \subset H$. Thus $H \cap P^i \neq \emptyset$. We conclude that $H$ is a supporting hyperplane of $P^i$ so $F_{P^i}^0 = y^{-1}(F_0) \cap P^i$ is a face of $P^i$ [24]. Since $P^0 \subset F_{P^i}^0$ and $P^0$ is, by construction, $(n-1)$-dimensional, we obtain, moreover, that $F_{P^i}^0$ is a facet of $P^i$.

\[\Box\]

### 5.4 Main Results

We consider again an $n$-dimensional polytope

$$P := \text{co}\{p_0, \ldots, p_r\}$$

with vertex set $V_P := \{p_0, \ldots, p_r\}$ and facets $F_0^P, \ldots, F_q^P$, where $F_0^P$ is the exit facet. Let $h_j^P$ be the unit normal to each facet $F_j^P$ pointing outside the polytope. We assume that the standard RCP is solvable on this polytope $P$. The polytope may have been obtained as the output of Algorithm 10 to guarantee solvability of the invariance conditions. We abuse notation and rename the output of the algorithm as $P$. It is assumed that the exit facet of $P$ is $F_0^P = y^{-1}(F_0) \cap P$, with outward normal $h_0^P$. Also $P \subset y^{-1}(S)$. 
Solvability of the RCP on $\mathcal{P}$ means that there exists a state feedback $u(x)$ such that for all $x_0 \in \mathcal{P}$, there exist $T \geq 0$ and $\gamma > 0$ such that

(i) $\phi(t, x_0) \in \mathcal{P}$ for all $t \in [0, T]$,

(ii) $\phi(T, x_0) \in \mathcal{F}_0^\mathcal{P}$, and

(iii) $\phi(t, x_0) \notin \mathcal{P}$ for all $t \in (T, T + \gamma)$.

The following lemma will be used in the main theorem. It examines the case when $\phi(T, x_0)$ is on the intersection of several facets. The lemma shows that trajectories cannot exit through a restricted facet without exiting the exit facet. The proof will be published elsewhere. A similar result can be found in [6], their proof does not address the issue of chattering; namely, trajectories may exit through two facets with infinitely high frequency.

**Lemma 5.4.1** ([19]). Consider an $n$-dimensional convex polytope $\mathcal{P}$ with facets $\mathcal{F}_0, \ldots, \mathcal{F}_r$, such that $0 \in \mathcal{F}_0, \ldots, \mathcal{F}_k$ for some $k < r$. Consider the affine system (5.1) and let $u(x)$ be a continuous piecewise affine feedback such that

$$h_i^\mathcal{P} \cdot (Ax + Bu(x) + a) \leq 0, \quad x \in \text{aff}(\mathcal{F}_i), \quad i = 1, \ldots, r. \quad (5.9)$$

Suppose $\phi(\cdot, 0)$ is the unique closed-loop trajectory of (5.1) with $\phi(0, 0) = 0$. If $\phi(\cdot, 0)$ exits $\mathcal{P}$ at time $T = 0$, then it does so by crossing $\mathcal{F}_0$.

We now present our main theorem.

**Theorem 5.4.1.** Suppose the RCP is solved by continuous piecewise affine feedback on $\mathcal{P}$. Then the ORCP given in Problem 5.1.2 is solved on $\mathcal{S}$.

**Proof.** We must show (i)-(iii) of Problem 5.1.2 are satisfied.

(i) Since $\mathcal{P} \subset y^{-1}(\mathcal{S})$, $\phi(t, x_0) \in \mathcal{P}$ for all $t \in [0, T]$. This implies that $\phi(t, x_0) \in y^{-1}(\mathcal{S})$ for all $t \in [0, T]$, and thus $y(t, x_0) \in \mathcal{S}$ for all $t \in [0, T]$.

(ii) Since $\phi(T, x_0) \in \mathcal{F}_0^\mathcal{P} = y^{-1}(\mathcal{F}_0) \cap \mathcal{P}$, we have that $y(T, x_0) \in y(y^{-1}(\mathcal{F}_0)) = \mathcal{F}_0$.

(iii) We know that $\phi(t, x_0) \notin \mathcal{P}$ for all $t \in (T, T + \gamma)$. First, suppose that $h_i^\mathcal{P} \cdot \phi(t, x_0) \leq 0$ for all $t \in (T, T + \gamma)$ and for all $i \in \{1, \ldots, q\}$. Then it must be that $h_0^\mathcal{P} \cdot \phi(t, x_0) > 0$ for all $t \in (T, T + \gamma)$. By Lemma 5.2.4, the outward normal vector of $\mathcal{F}_0^\mathcal{P}$ is $h_0^\mathcal{P} = \frac{C^T h_0}{\|C^T h_0\|}$. Thus $h_0^\mathcal{P} \cdot \phi(t, x_0) = \|C^T h_0\| h_0 \cdot C \phi(t, x_0) > 0$ for all $t \in (T, T + \gamma)$. This implies $h_0 \cdot y(t, x_0) > 0$ for all $t \in (T, T + \gamma)$, which proves (iii).

Suppose w.l.o.g. $\phi(T, x_0) = 0 \in \mathcal{F}_0^\mathcal{P} \cap \cdots \cap \mathcal{F}_k^\mathcal{P}$, where $0 \leq k < q$ is the largest such integer. In the first case we assumed $\phi(t, x_0)$ did not cross a restricted facet $\mathcal{F}_i^\mathcal{P}$, $i = 1, \ldots, q$, on the interval $(T, T + \gamma)$.
the first case we assumed \( \phi(t, x_0) \) did not exit through a restricted facet \( \mathcal{F}_i^P, i = 1, \ldots, q \), on the interval \((T, T + \gamma)\). Second, suppose w.l.o.g. that the first \( l \) restricted facets are exited through at certain times in the interval \((T, T + \gamma)\). By assumption, \( \phi(T, x_0) \not\in \mathcal{F}_{k+1} \cup \cdots \cup \mathcal{F}_q \). Therefore, there exists \( \bar{\gamma} > 0 \) such that \( \phi(t, x_0) \) does not exit through the restricted facets \( \mathcal{F}_i^P, i = k + 1, \ldots, q \), on the interval \((T, T + \bar{\gamma})\).

Let \( \gamma' := \min\{\bar{\gamma}, \gamma_0, \ldots, \gamma_l\} \). Then by (iv), \( \phi(t, x_0) \in \{x \in \mathbb{R}^n \mid h_0^P \cdot x > 0, j = 1, \ldots, l\} \) for all \( t \in (T, T + \gamma') \). By Lemma 5.4.1, there exists \( \gamma'(T, x_0) \in \{x \in \mathbb{R}^n \mid h_0^P \cdot x > 0\} \). Then again by (iv), \( \phi(t, x_0) \in \{x \in \mathbb{R}^n \mid h_0^P \cdot x > 0\} \) for all \( t \in (T, T + \gamma') \). This implies \( h_0 \cdot y(t, x_0) > 0 \) for all \( t \in (T, T + \gamma) \), which proves (iii).

\[ \square \]

### 5.5 Example

Consider the system

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u, \quad u \in [-1, 1] \\
y &= x_1 \in [0, 1]
\end{align*}
\]

where \( S = [0, 1] \) is the output simplex, and \( \mathcal{F}_0 = 1 \) is the exit facet. The control objective is for the output trajectories to leave through the facet \( \mathcal{F}_0 \) in finite time, without exiting through \( y = 0 \) first. We begin by constructing the polytope \( \mathcal{P} \). Since \( T = [C^T C]' = I \) for this problem, we have that \( x = \tilde{x} \).

Let \( \mathcal{P}_{\text{box}} = \{x \mid -1 \leq x_2 \leq 1\} \). We have that \( \mathcal{P} = y^{-1}(S) \cap \mathcal{P}_{\text{box}} = \{x \mid x \in [0, 1] \times [-1, 1]\} \) shown in Figure 5.1a below with exit facet \( \mathcal{F}_0^P = y^{-1}(\mathcal{F}_0) \cap \mathcal{P} \).

Let \( \mathcal{P} := \text{co}\{p_0, p_1, p_2, p_3\} \) as seen in Figure 5.1a. The invariance conditions fail for vertex \( p_2 \), and therefore the standard RCP is not solvable for \( \mathcal{P} \). We use Algorithm 7 to create a polytope which satisfies the invariance conditions. We have that \( \mathcal{P}^0 = \text{cl}\{x \in \mathcal{F}_0^P \mid h_0^P (Ax + a) > 0\} = \{x \in \mathcal{F}_0^P \mid x_2 \geq 0\} = \text{co}\{(1, 1), (0, 0)\} := \text{co}\{p_0^0, p_1^0\} \).

In step 3) we choose \( v = p_0^0 \). Proceeding to step 4), since \( D \neq \emptyset \), we solve the optimization problem which yields \( \{w_0, w_1, w_2, w_3\} = \{(0, 1), (1, 1), (0.36, -0.28), (1, -0.95)\} \). Since \( \bar{\mathcal{P}} \neq \mathcal{P}^0 \), we have a viable polytope \( \mathcal{P}^1 = \bar{\mathcal{P}} = \text{co}\{w_0, w_1, w_2, w_3\} \) as shown in Figure 5.1b below.

The dashed lines are to show that each \( w_j \in \text{co}\{v, p_j\} \). Continuing the algorithm selecting \( v \) for step 3) as shown in each of the figures, we arrive at \( \mathcal{P}^3 = \text{co}\{(0, 1), (1, 1), (0, 0), (0.25, -0.5), (0.75, -1), (1, -1)\} \) where the algorithm terminates.
With a viable polytope, we can try and use standard RCP techniques to ensure trajectories exit the polytope. In this example we choose to triangulate the polytope $\mathcal{P}^3$ and solve the RCP on each simplex. The proposed triangulation is shown in Figure 5.1f. Also shown in the figure are the chosen closed-loop velocity vectors. Using Algorithm 5.2 of [7] we solve the desired RCP on the polytope $\mathcal{P}^3$.

It is clear from the closed-loop velocity vectors that any initial condition in $\mathcal{P}^3$ has a solution which ensures that $x_1 \geq 0$, and that $x_1$ leaves the output simplex through $\mathcal{F}_0$ as desired.

**Remark 5.5.1.** We observe that this example cannot be solved by the standard reach control methods. The specifications imposed in the example form a polyhedron in the state space, not a polytope. As such, standard reach control methodologies cannot be applied. Chapter 5 is to provide a theoretical
framework to address such examples where the specifications do not form a polytope in the state space.

All of the development in Chapter 4 can be applied to Chapter 5 since the solution involves generating reach controllers in the full dimensional state space. That being said, there are no formal guarantees that the ORCP will be solved under modeling uncertainty or disturbances.
Chapter 6

Motion Planning Framework for Integrator Systems

Chapters 6 and 7 take a completely different approach from Chapters 4 and 5, yet the theme remains the same. Rather than considering general affine control systems, we focus on integrator systems. We specify atomic behaviors in the output space, called motion primitives. Because we are dealing with a specific class of systems and specific closed-loop behaviors, one can readily apply the viability theory algorithm in Chapter 7 to obtain polytopes on which piecewise affine feedbacks can be defined to achieve the goal of each motion primitive.

In essence, the idea of motion primitives is to devise closed-loop behavior using reach controllers with the state space triangulation already “built in” to the control design. The difficulty of constructing a triangulation is completely bypassed by this approach. On the other hand, the price to be paid is that we only work with integrator systems. Such systems have received considerable attention in the multi-agent systems community. Many highly complex robotic models, such as the quadrocopter model, can be reduced to a collection of double integrator models by using feedback linearization. This fact has inspired our choice of the integrator model on which to design motion primitives.

We presents a modular, hierarchical framework for motion planning and control of robotic systems. While motion planning has received a great deal of attention by many researchers, because the problem is highly complex especially when there are several robotic agents working together in a cluttered environment, significant challenges remain. Hierarchy, in which the control design has several layers, is an architectural strategy to overcome this complexity. Almost all hierarchical frameworks for motion planning aim to balance flexibility in the control specification at the high level, guarantees on correct-
Figure 6.1: A two output ($p = 2$) example of a reach-avoid task. Shown on the left is the feasible space $\mathcal{P}$ consisting of 15 non-obstacle boxes (not red) and the goal region $\mathcal{O}_\mathcal{P}$ (green). The output transition system (OTS), which abstracts the box regions and their neighbour connectivity, is shown on the right. Shown below, the possible offsets towards a neighbouring box are labelled using $\Sigma = \{-1, 0, 1\}^2$, which includes the box edges, vertices, and interior.

We propose a modular hierarchical framework so that one can plug and play both low level controllers and high level planning algorithms in order to realize a balance between flexibility at the high level, safety at the low level, and computational feasibility overall. This chapter was developed in collaboration with my colleague Mario Vukosavljev, and I would like to thank him for his attention to detail.

We introduce two assumptions. First, we address reach-avoid specifications, in which the system must reach a desired configuration in a safe manner [34, 38, 35, 33]. Reach-avoid offers a sufficiently rich behavior set so that, for instance, a fragment of linear temporal logic (LTL) can be encoded as a sequence of
reach-avoid problems [50]. Second, we assume that the underlying dynamics have certain symmetries, namely position invariance, a property satisfied by many robot models. Under these two assumptions, our overarching contribution can be summarized as follows:

The general motion planning problem for robotic systems with LTL specifications is highly computationally complex and generally infeasible if a full account of low level dynamics is required. We propose a hierarchical control framework that manages the complexity of the planning problem, in essence, by introducing a set of reasonable design assumptions that at no point sacrifice on the resilience and accuracy of low level behavior. We invoke reach control theory for built-in safety guarantees at the low level. We invoke standard discrete planning algorithms at the high level. And we provide a formal guarantee that the two levels will operate consistently.

Now we given an overview of the features and techniques we employ, and we highlight other frameworks that share those features. As already mentioned, we focus on a reach-avoid specification [33, 34, 35], in which robotic agents must navigate an environment cluttered with obstacles to reach a final goal configuration. We abstract the physical workspace into a finite number of rectangular regions [39] rather than more general polytopic regions [38, 37, 51, 33]. We assume the nonlinear control system has a translational symmetry in the output, thus simplifying the low level control design [45]. To further simplify the low level control design, we employ motion primitives [47, 39, 35], while feasible sequences of motion primitives are encoded by a maneuver automaton [45]. The low level control design of motion primitives is based on reach control theory because this theory provides guarantees on safety [9, 7, 51]. Finally, planning at the high level is based on standard shortest path algorithms [52] applied to the graph arising from the synchronous product of the discrete part of the maneuver automaton and the graph arising from the workspace abstraction.

Going more deeply into our contributions, we highlight three aspects: robustness due to feedback control at two levels; correct by design state space partitioning; and computational feasibility. First, a benefit of our approach is the robustness it affords to the motion planning problem by way of low level controllers based on reach control theory [7, 9, 16, 17, 1]. Reach control theory provides a highly flexible and intuitive set of design tools that have two notable advantages over tracking: first, it is not necessary to find feasible open-loop trajectories; second, safety constraints on the system states are guaranteed by design. Therefore, in addition to the robustness derived from using high level planning algorithms that account for all possible initial conditions of the system, we also inherit robustness from these low level feedback controllers that offer strict safety guarantees.

Reach control has been used by other researchers in the context of low level control design for motion planning [7, 51, 34]. Typically these methods only apply in the robot workspace with no guarantees
on the behavior in the full state space. Or otherwise a generic partition of the state space, such as a
triangulation, is employed, with no guarantees that reach controllers can be devised to work correctly on
that specific partition. In contrast, we construct a partition of the state space such that the requirements
of reach control are guaranteed to hold on that partition. A set of closed-loop vector fields are defined for
integrator systems; these so-called motion primitives are bundled with the state space partition so that
no further partition refinement is need to obtain a feasible transition system for the high level planning
stage of design.

Along with correct by design state space partitioning, we also propose general conditions on the
maneuver automaton (MA) in order that the high level (discrete) plan is consistent with low level
(continuous time) behaviors. The design of a well-posed MA for higher order systems can be challenging,
in general. Fortunately, higher-order system models often have a decoupled structure; for example, when
there are multiple vehicles or the dynamics are decoupled in individual degrees of freedom.

Finally, one of our main modeling assumptions is that the system dynamics have a symmetry in
the outputs \[45\]. This assumption implies that motion primitives can be designed over a single box
in a gridded output space. The design can then be reapplied to other boxes in the output space since
they are merely translations of the original box. This feature significantly reduces the computational
task of generating feedback controllers over partitions of the output space. Since the complete motion
capabilities of the system, represented as a graph, explodes in complexity with the number of outputs,
this decoupled and compact representation of the system’s motion capabilities enables computational
feasibility for some modestly sized and realistic motion planning scenarios.

## 6.1 Related Literature

Several papers have been inspirational for our work. First, \[51\] provides a general framework for solving
control problems for affine systems with linear temporal logic (LTL) specifications. Their approach
involves constructing a transition system over a polyhedral partition of the state space that arises from
linear inequality constraints that constitute the atomic propositions of the LTL specification. Transitions
between states of the transition system can occur if there exists an affine or piecewise affine feedback
steering all continuous time trajectories from one polyhedral region to a contiguous one. The elegance of
their approach derives from its generality - any LTL specification can be dealt with using this framework,
and their construction of the transition system gives a faithful account of the low level system capabilities.

Our framework has many similarities to the one in \[51\]. For example, both in \[51\] and in our work,
low level controllers are constructed based on reach control theory for affine systems. Both frameworks
build a transition system to abstract the low level dynamics, and both work with a polyhedral partition of the state space. There are also some differences. First, to make the problem tractable, we only deal with a reach-avoid specification, a subset of LTL. In principle, our framework is extendable to LTL specifications; in practice, as with [51], we would need to contend with the computational complexity of synthesis for LTL, and we would need to deal with the possibility of Zeno behavior. Second, our transition system is constructed rather differently from the one in [51]. We start with a rectangular, rather than polyhedral, partition of the output space, not the state space. We model constraints on successive motion primitives using a maneuver automaton. Then our transition system is the synchronous product of the transition system abstracting the gridded output space and the discrete part of the maneuver automaton.

This construction offers two advantages. First, the resulting partition of the full state space does not come directly from the control specification, but rather it comes through the motion primitives, which embody what the system can do. This means our partition is consonant with system capabilities. A second advantage of our transition system construction is that we are able to restrict the sequence of transitions such that there is no chattering or Zeno behavior, and so that large discontinuities in the inputs or state derivatives do not occur upon taking a transition.

Finally, we mention that a general limitation of some synthesis methods for LTL specifications is that they do not yield an actual feedback at the high level, despite the fact that low level controllers may be feedbacks. This is because an explicit accepting run of the transition system is specified for each initial condition. Under large disturbances, the system may not follow this run at all, and there is no corrective measure in realtime to be taken. For reach-avoid specifications, it is known that a feedback solution can be constructed [50], as we do.

A second inspirational paper is [50] which, like [51], deals with LTL specifications. They identify the difficulty to solve general LTL control problems, so they focus on a fragment of LTL that reduces to solving a sequence of reach-avoid problems. They provide high level planning strategies that potentially can have wide practical applicability. On the other hand, they assume a transition system is already known. We have built our framework in order that it can be extended to address their fragment of LTL; see Remark 6.3.5(iii). Our contribution is to bridge the gap on how to systematically and efficiently construct transition systems that faithfully represent the low level dynamics.

The design presented here was informed by our overriding objective to devise a framework for motion control of a collection of quadrocopters. This chapter provides the theoretical foundations to complement our experimental work on quadrocopters [32, 53]. Current experimental results can be viewed at http://tiny.cc/quad5scenes.
6.2 Problem Statement

Consider the general nonlinear control system

\[
\dot{x} = f(x, u), \quad y = h(x),
\]  

(6.1)

where \(x \in \mathbb{R}^n\) is the state, \(u \in \mathbb{R}^\mu\) is the input, and \(y \in \mathbb{R}^p\) is the output. Let \(\phi(\cdot, x_0)\) and \(y(\cdot, x_0)\) denote the state and output trajectories of (6.1) starting at initial condition \(x_0 \in \mathbb{R}^n\) and under some open-loop or feedback control.

Let \(P \subset \mathbb{R}^p\) be a feasible set in the output space and let \(O_P \subset P\) be a goal set. In multi-vehicle motion planning contexts, \(P\) represents the feasible joint output configurations of the system, which can arise from specifications involving obstacle avoidance, collision avoidance, communication constraints, and others. We consider the following problem.

**Problem 6.2.1 (Reach-Avoid).** We are given the system (6.1), a non-empty feasible set \(P \subset \mathbb{R}^p\) and a non-empty goal set \(O_P \subset P\). Find a feedback control \(u(x)\) and a set of initial conditions \(X_0 \subset \mathbb{R}^n\) such that for each \(x_0 \in X_0\) we have

(i) **Avoid:** \(y(t, x_0) \notin \mathbb{R}^p \setminus P\) for all \(t \geq 0\),

(ii) **Reach:** there exists \(T \geq 0\) such that for all \(t \geq T\), \(y(t, x_0) \in O_P\).

We make an assumption regarding the outputs of the system (6.1) in order to exploit symmetry; see [45] for an exposition on nonlinear control systems with symmetries.

**Assumption 6.2.1.** First, we assume that there is an injective map \(o : \{1, \ldots, p\} \to \{1, \ldots, n\}\) associating each output to a distinct state, so that \(h(x) = (x_{o(1)}, \ldots, x_{o(p)})\). We define an (injective) insertion map \(h^{-1}_o : \mathbb{R}^p \to \mathbb{R}^n\) as \(h^{-1}_o(y) = x\), which satisfies \(h(x) = y\) and \(x_i = 0\) for all \(i \in \{1, \ldots, n\} \setminus \{o(1), \ldots, o(p)\}\).

Second, we assume that the system has a translational invariance with respect to its outputs. That is, for all \(x \in \mathbb{R}^n\), \(u \in \mathbb{R}^\mu\) and \(y \in \mathbb{R}^p\), we have \(f(x, u) = f(x + h^{-1}_o(y), u)\).

The assumption that the outputs of the system are a subset of the states is used in our framework to be able to design feedback controllers in the full state space that achieve desirable behavior in the output space. The second statement says that the vector field is invariant to the value of the output. In the literature this condition is called a symmetry of the system or translational invariance. This assumption is satisfied for many robotic systems, for example, when the outputs are positions. Also, we will see in Section 6.5 that it significantly simplifies our control design.
6.3 Modular Framework

In this section we present our methodology to solve the motion planning problem in the form of an architecture that breaks down Problem 6.2.1. This architecture consists of five main modules, as depicted in Figure 6.2.

- The problem data include the system (6.1) with $p$ outputs satisfying Assumption 6.2.1 and a reach-avoid task to be executed in the output space.

- The Output Transition System (OTS) is a graph that captures an abstraction of the output space based on a partition of the output space. Specifically, we partition the output space into $p$-dimensional boxes with lengths $d = (d_1, \ldots, d_p)$, where $d_i > 0$ is the length of $i$th edge. We use a finite number of boxes to under-approximate the feasible set $\mathcal{P}$. Enumerating the boxes as $\{Y_1, \ldots, Y_{n_L}\}$, the $j$-th box is given by

$$Y_j := \prod_{i=1}^{p} [\eta_{ji}d_i, (\eta_{ji} + 1)d_i],$$

where $\eta_{ji} \in \mathbb{Z}$, $i = 1, \ldots, p$ are constants. The OTS is a directed graph whose nodes (called locations) represent boxes and whose edges describe which boxes in the output space are contiguous.

In essence, rather than bookkeeping where a particular robot is positioned in its output space, we will only record which box it is in and which boxes it can reach next. The reach-avoid task will be
re-expressed in terms of desired behaviors on the OTS.

- The **Maneuver Automaton** (MA) is a hybrid system whose modes correspond to so-called motion primitives. Each motion primitive is associated with a closed-loop vector field by applying a feedback law to (6.1). The edges of the MA represent feasible successive motion primitives. Each motion primitive generates some desired behavior of the output trajectories of the closed-loop system over a box in the output space. Because of the uniform gridding of the output space into boxes and because of the symmetry in the outputs described in Assumption 6.2.1, motion primitives can be designed over only one canonical box $Y^*$.

- The **Product Automaton** (PA) is a graph which is the synchronous product of the OTS and the discrete part of the MA. It represents the combined constraints on feasible motions in the output space and feasible successive motion primitives.

- The **hybrid control strategy** is a combination of low level controllers obtained from the design of motion primitives, and a high level plan obtained by applying a shortest path algorithm adapted to non-deterministic graphs on the PA [48].

Next we describe in greater detail the OTS, MA, and PA.

### 6.3.1 Output Transition System

The OTS provides an abstract description of the **workspace or output space** associated with the system (6.1). It serves to capture the feasible motions of output trajectories of the system (6.1) in a gridded output space, as in Figure 6.1. Let $\{Y_j\}_{j=1}^{n_L}$ be a collection of $p$-dimensional boxes that under-approximate the safe output space $\mathcal{P}$, with $Y_j$ as defined in (6.2). That is,

$$\bigcup_{j=1}^{n_L} Y_j \subset \mathcal{P}.$$  

Among these boxes, we assume there is a non-empty set of indices $I_g \subset \{1, \ldots, n_L\}$, so that we may similarly under-approximate the goal region as

$$\bigcup_{j \in I_g} Y_j \subset \mathcal{O} \subset \mathcal{P}.$$
We define a canonical $p$-dimensional box with edge lengths $d_i > 0$ given by

$$Y^* = \prod_{i=1}^{p} [0, d_i].$$

Referring to (6.2), we can see that each box $Y_j$, $j = 1, \ldots, n_L$ is a translation of $Y^*$ by an amount $\eta_{ji}d_i$ along the $i$-th axis.

**Definition 6.3.1.** Given the lengths $d$ and a non-empty goal index set $I_g$, an output transition system (OTS) is a tuple $\mathcal{A}_{OTS} = (L_{OTS}, \Sigma, E_{OTS})$ with the following components:

- **Locations** $L_{OTS} := \{l_1, \ldots, l_{n_L}\} \subset \mathbb{Z}^p$ is a finite set of nodes called locations. Each location $l_j \in L_{OTS}$ is associated with a safe box $Y_j \subset \mathcal{P}$ in the output space and hence we write $l_j = (\eta_{j1}, \ldots, \eta_{jp})$. The set $L_{OTS}^g = \{l_j \in L_{OTS} | j \in I_g\}$ denotes the set of locations associated with goal boxes.

- **Labels** $\Sigma := \{-1, 0, 1\}^p \subset \mathbb{Z}^p$ is a finite set of labels. A label $\sigma \in \Sigma$ is used to identify the offset between neighbouring boxes.

- **Edges** $E_{OTS} \subset L_{OTS} \times \Sigma \times L_{OTS}$ is a set of directed edges where $(l_j, \sigma, l_j') \in E_{OTS}$ if $j \neq j'$ and $Y_j \cap Y_{j'} \neq \emptyset$. We define $\sigma = l_j' - l_j \in \Sigma$. Thus, for each $i = 1, \ldots, p$, the neighbouring box $l_j'$ is either one box to the left ($\sigma_i = -1$), the same box ($\sigma_i = 0$), or one box to the right ($\sigma_i = 1$). In this manner $\sigma$ records the offset between contiguous boxes.

**Remark 6.3.1.** We observe that the OTS is deterministic. That is, for a given $l \in L_{OTS}$ and $\sigma \in \Sigma$, there is at most one $l' \in L_{OTS}$ such that $(l, \sigma, l') \in E_{OTS}$. This follows immediately from the fact that $\sigma = l' - l$ records the offset between the neighbouring boxes.

Figure 6.1 shows a sample OTS for a simple scenario. The OTS locations are associated with 15 feasible boxes, including a goal box for the reach-avoid task. The OTS edges are shown as bidirectional arrows; for example, interpreting $l_1 = (0, 0)$ and $l_6 = (1, 1)$ on the grid, then $e = (l_6, (-1, -1), l_1) \in E_{OTS}$.

### 6.3.2 Maneuver Automaton

The maneuver automaton (MA) is a hybrid system consisting of a finite automaton and continuous time dynamics in each discrete state. The discrete states of the finite automaton correspond to motion primitives, while transitions between discrete states correspond to the allowable transitions between motion primitives. The continuous time dynamics are given by closed-loop vector fields (6.1) with a control law designed based on reach control theory (any other feedback control design method can be used).
Before presenting the MA, we first explain how this module is used in the overall framework. To solve Problem 6.2.1, we assign motion primitives to the boxes $Y_j$ of the partitioned output space such that obstacle regions are avoided and the goal region is eventually reached. The discrete part of the MA encodes the constraints on successive motion primitives. Such constraints might arise from a non-chattering requirement, continuity requirement, or requirement on correct switching between regions of the state space. A dynamic programming algorithm for assignment of motion primitives on boxes is addressed in Section 6.3.4; the salient point about this algorithm at this stage is that it only uses the discrete part of the MA.

In contrast, the continuous time part of the MA is used both for simulation of the closed-loop dynamics to verify that the motion primitives are well designed, as well as for the implementation of the low level feedback in real-time. The motion primitives are defined only on the canonical box $Y^*$ to simplify the design. This simplification is possible because of the translational symmetry provided by Assumption 6.2.1 and the fact that each box $Y_j$ is a translation of $Y^*$. In simulation, a given motion primitive can cause output trajectories to reach certain faces of $Y^*$. If a face is reached, the output trajectory is interpreted as being reset to the opposite face and another motion primitive is selected to be implemented over $Y^*$ (of course, the real experimental output trajectories do not undergo resets but move continuously from box to box in the output space). The selection of the next motion primitive is constrained by a combination of the previous motion primitive and the face of $Y^*$ that is reached. The discrete transitions in the MA encode these constraints.

**Definition 6.3.2.** Consider the system (6.1) satisfying Assumption 6.2.1 and the box $Y^*$ with lengths $d$. The maneuver automaton (MA) is a tuple $H_{MA} = (Q_{MA}, \Sigma, E_{MA}, X_{MA}, I_{MA}, G_{MA}, R_{MA}, Q_{0MA})$, where

**State Space** $Q_{MA} = M \times \mathbb{R}^n$ is the hybrid state space, where $M = \{m_1, \ldots, m_{nM}\}$ is a finite set of nodes, each corresponding to a motion primitive.

**Labels** $\Sigma$, the same labels used in the OTS.

**Edges** $E_{MA} \subseteq M \times \Sigma \times M$ is a finite set of edges.

**Vector Fields** $X_{MA} : M \to \mathcal{X}(\mathbb{R}^n)$ is a function assigning a globally Lipschitz closed-loop vector field to each motion primitive $m \in M$. That is, for each $m \in M$, we have $X_{MA}(m) = f(\cdot, u_m(\cdot))$ where $u_m(\cdot)$ is a feedback controller associated with $m \in M$.

**Invariants** $I_{MA} : M \to 2^{\mathbb{R}^n}$ assigns a bounded invariant set $I_{MA}(m)$ to each $m \in M$. We impose that $I_{MA}(m) \subset h^{-1}(Y^*)$. The set $I_{MA}(m)$ defines the region on which the vector field $X_{MA}(m)$ is defined. Note that there is no requirement that the invariant is a closed set.
Enabling Conditions $G_{\mathrm{MA}} : E_{\mathrm{MA}} \to \{ g_e \}_{e \in E_{\mathrm{MA}}} \text{ assigns to each edge } e = (m, \sigma, m') \in E_{\mathrm{MA}} \text{ a non-empty enabling or guard condition } g_e \subset \mathbb{R}^n$. We require that $g_e \subset I_{\mathrm{MA}}(m)$. We make an additional requirement that $g_e \text{ lies on a certain face of } Y^* \text{ determined by the label } \sigma = (\sigma_1, \ldots, \sigma_p) \in \Sigma$. Defining the face associated with $\sigma$ as

$$F_\sigma = \left\{ y \in Y^* \left| \begin{array}{l} y_i = 0, \quad \text{if } \sigma_i = -1 \\ y_i = d_i, \quad \text{if } \sigma_i = 1 \end{array} \right. \right\},$$

we require that also $g_e \subset h^{-1}(F_\sigma)$.

Reset Conditions $R_{\mathrm{MA}} : E_{\mathrm{MA}} \to \{ r_e \}_{e \in E_{\mathrm{MA}}} \text{ assigns to each edge } e = (m, \sigma, m') \in E_{\mathrm{MA}} \text{ a reset map } r_e : \mathbb{R}^n \to \mathbb{R}^n$. We require that $r_e(x) = x - h_0^{-1}(d \circ \sigma)$. This definition says that the $i$-th output component is reset to the right face of $Y^*$, $x_{o(i)} = d_i$, if $\sigma_i = -1$, reset to the left face $x_{o(i)} = 0$ if $\sigma_i = 1$, and unchanged otherwise. Overall, resets of states are determined by the event $\sigma \in \Sigma$ and only affect the output coordinates in order to maintain output trajectories inside the canonical box $Y^*$.

Initial Conditions $Q^0_{\mathrm{MA}} \subset Q_{\mathrm{MA}}$ is the set of initial conditions given by $Q^0_{\mathrm{MA}} = \{(m, x) \in Q_{\mathrm{MA}} \mid x \in I_{\mathrm{MA}}(m)\}$.

We now formulate assumptions on the motion primitives so that correct continuous time behavior is ensured at the low level for consistency with the high level. For each $m \in M$, define the set of possible events as

$$\Sigma_{\mathrm{MA}}(m) := \{ \sigma \in \Sigma \mid (\exists m' \in M)(m, \sigma, m') \in E_{\mathrm{MA}} \}.$$  \hspace{1cm} (6.3)

Assumption 6.3.1.

(i) For all $m \in M$, $\varepsilon := (0, \ldots, 0) \not\in \Sigma_{\mathrm{MA}}(m)$.

(ii) For all $e_1, e_2 \in E_{\mathrm{MA}}$ such that $e_1 = (m_1, \sigma, m_2)$ and $e_2 = (m_1, \sigma, m_3)$, $g_{e_1} = g_{e_2}$.

(iii) For all $e_1, e_2 \in E_{\mathrm{MA}}$ such that $e = (m_1, \sigma_1, m_2)$ and $e_2 = (m_1, \sigma_2, m_3)$, if $\sigma_1 \neq \sigma_2$, then $g_{e_1} \cap g_{e_2} = \emptyset$.

(iv) For all $e_1, e_2 \in E_{\mathrm{MA}}$ such that $e_1 = (m_1, \sigma_1, m_2)$ and $e_2 = (m_2, \sigma_2, m_3)$, $r_{e_1}(g_{e_1}) \cap g_{e_2} = \emptyset$.

(v) For all $e = (m_1, \sigma, m_2) \in E_{\mathrm{MA}}$, $r_e(g_e) \subset I_{\mathrm{MA}}(m_2)$.

(vi) For all $m \in M$, if $\Sigma_{\mathrm{MA}}(m) = \emptyset$ then for all $x_0 \in I_{\mathrm{MA}}(m)$ and $t \geq 0$, $\phi_{\mathrm{MA}}(t, x_0) \in I_{\mathrm{MA}}(m)$.
(vii) For all \( m \in M \), if \( \Sigma_{\text{MA}}(m) \neq \emptyset \), then for all \( x_0 \in I_{\text{MA}}(m) \) there exist (a unique) \( \sigma \in \Sigma_{\text{MA}}(m) \) and (a unique) \( T \geq 0 \) such that for all \( e = (m, \sigma, m') \in E_{\text{MA}} \) and for all \( t \in [0, T] \), \( \phi_{\text{MA}}(t, x_0) \in I_{\text{MA}}(m) \) and \( \phi_{\text{MA}}(T, x_0) \in g_e \).

\[ \square \]

Condition (i) disallows tautological chattering behavior that arises by erroneously interpreting continuous evolution of trajectories in the interior of \( Y^* \) as “discrete transitions” of the MA (see Section 6.4 for definitions). Condition (ii) imposes that guard sets are independent of the next motion primitive. Since guard sets arise as the set of exit points of closed-loop trajectories from \( Y^* \) under a given motion primitive, it is reasonable that these exit points should depend only on the current motion primitive \( m \in M \), and not on the choice of next motion primitive. Condition (iii) imposes that all guard sets corresponding to different labels are non-overlapping. This ensures that when the continuous trajectory reaches a guard \( g_e \), then it is unambiguous which edge of the MA is taken next; namely \( e \in E_{\text{MA}} \). Conditions (v), (vi), and (vii) are placed to guarantee that the MA is non-blocking. These conditions are based on known results in the literature [60]; see Lemma 6.4.2. In order for condition (vii) to make sense, there must exist a unique label \( \sigma \in \Sigma \) and a unique time \( T \geq 0 \) for an MA trajectory to reach a guard set. First, we have uniqueness of solutions since the vector fields are globally Lipschitz. Second, the unique MA trajectory can only reach one guard set by condition (iii); this in turn means there is a unique \( \sigma \). Obviously there exists a unique time to reach the guard set. Conditions (vi) and (vii) work together to state that either all trajectories do not leave, or all trajectories do eventually leave. Finally, condition (iv) eliminates potential chattering Zeno behavior.

Remark 6.3.2. We make several further observations about the MA.

(i) Motion primitives are defined only on \( Y^* \) so trajectories of the MA undergo resets when MA output trajectories reach a guard lying on a face of \( Y^* \) so that they will continue to evolve on \( Y^* \). In contrast, the trajectories of (6.1) and of the real physical system do not undergo resets.

(ii) The MA is non-deterministic in the sense that given \( m \in M \) and \( \sigma \in \Sigma \), there may be multiple \( m' \in M \) such that \( (m, \sigma, m') \in E_{\text{MA}} \). The discrete part of the MA is non-deterministic in a second sense: for each \( m \in M \), the cardinality of the set \( \Sigma_{\text{MA}}(m) \) may be greater than one. The latter situation corresponds to the fact that for different initial conditions \( x_1, x_2 \in I_{\text{MA}}(m) \) of the continuous part, the associated output trajectories can reach different guard sets. In essence, which guard is enabled is interpreted, at the high level, as an uncontrollable event [59]. Remark 6.3.3 further illustrates these two types of non-determinism in the case of the PA.
(iii) The set of events $\Sigma$ in the MA correspond to the same events $\Sigma$ in the OTS. This correspondence is used in the product automaton PA, described in the next section, to synchronize transitions in the MA with transitions in the OTS. The interpretation is that when a continuous trajectory of the MA (over the box $Y^{\ast}$) undergoes a reset with the label $\omega \in \Sigma$, the associated continuous trajectory of (6.1) in the box $Y_j$ enters a neighboring box $Y_{j'}$ with the offset $\omega = l_{j'} - l_j$. Obviously, this interpretation assumes that the vector of box lengths $d > 0$ is the same in both OTS and MA.

### 6.3.3 Product Automaton

In this section we introduce the product automaton (PA). It is constructed as the synchronous product of the OTS and the discrete part of the MA, namely $(M, \Sigma, E_{\text{MA}})$. The purpose of the PA is to merge the constraints on successive motion primitives with the constraints on transitions in the OTS. As such, it captures the overall feasible motions of the system – any high level plan must adhere to these feasible motions.

**Definition 6.3.3.** We are given an OTS $\mathcal{A}_{\text{OTS}}$ and an MA $\mathcal{H}_{\text{MA}}$ satisfying Assumption 6.3.1. We define the product automaton (PA) to be the tuple $\mathcal{A}_{\text{PA}} = (Q_{\text{PA}}, \Sigma, E_{\text{PA}}, Q_{f_{\text{PA}}}, D_{\text{PA}}, H_{\text{PA}})$, where

**State Space** $Q_{\text{PA}} \subset L_{\text{OTS}} \times M$ is a finite set of PA states. A PA state $q = (l, m) \in Q_{\text{PA}}$ satisfies the following: if there exists $\omega \in \Sigma$ and $m' \in M$ such that $(m, \omega, m') \in E_{\text{MA}}$, then there exists $l' \in L_{\text{OTS}}$ such that $(l, \omega, l') \in E_{\text{OTS}}$. That is, $(l, m) \in Q_{\text{PA}}$ if all faces that can be reached by motion primitive $m \in M$ lead to a neighboring box of the box associated with location $l \in L$ of the OTS.

**Labels** $\Sigma$ is the same set of labels used by the OTS and the MA.

**Edges** $E_{\text{PA}} \subset Q_{\text{PA}} \times \Sigma \times Q_{\text{PA}}$ is a set of directed edges defined according to the following rule. Let $q = (l, m) \in Q_{\text{PA}}$, $q' = (l', m') \in Q_{\text{PA}}$, and $\omega \in \Sigma$. If $(l, \omega, l') \in E_{\text{OTS}}$ and $(m, \omega, m') \in E_{\text{MA}}$, then $(q, \omega, q') \in E_{\text{PA}}$.

**Final Condition** $Q_{f_{\text{PA}}} \subset L_{\text{OTS}}^g \times M$ is the set of final PA states.

**Discrete Cost** $D_{\text{PA}} : E_{\text{PA}} \to \mathbb{R}$ is the instantaneous cost. We assume that $D_{\text{PA}}(e) > 0$ for all $e \in E_{\text{PA}}$.

**Terminal Cost** $H_{\text{PA}} : Q_{\text{PA}} \to \mathbb{R}$ is the terminal cost.

**Remark 6.3.3.** Formally an automaton is said to be non-deterministic if there exists a state with more than one outgoing edge with the same label. The PA is non-deterministic. First, consider a PA state $q = (l, m) \in Q_{\text{PA}}$. Because the MA allows for more than one feasible next motion primitive $m'$ such that
(m, σ, m') ∈ E_{MA}, the PA will also have multiple next PA states q' = (l', m') such that (q, σ, q') ∈ E_{vA}.

Second, there can be multiple possible labels σ ∈ Σ such that e = (q, σ, q') ∈ E_{vA} for some q' ∈ Q_{vA}. Thus, the PA inherits the two types of non-determinism of the MA that we discussed in Remark 6.3.2.

For example, consider the PA fragment in Figure 6.3. For the first type of non-determinism, observe that there are two PA edges (q_1, σ_1, q_2) ∈ E_{vA} and (q_1, σ_1, q_3) ∈ E_{vA} with the same label. For the second type, observe that there are two possible events σ_1, σ_2 ∈ Σ from q_1, each with its own set of PA edges. Note also some additional structure: since the OTS is deterministic, the box state is l_2 in both q_2 and q_3, corresponding to the OTS edge (l_1, σ_1, l_2) ∈ E_{OTS}.

Remark 6.3.4. We note that the introduction of a cost function on the product automaton is an artificial device in order to be able to apply dynamic programming; see [7] for a similar formulation.

### 6.3.4 High-Level Plan

In this section we formulate a high level plan on the PA. Informally, the objective of the high level plan is to find a set of initial PA states and to develop a rule for selecting subsequent PA states (effectively by choosing the next motion primitive) such that a goal PA state is eventually reached. To this end, in this section we develop a Dynamic Programming Principle (DPP) suitable for use on the PA. Because of the two types of non-determinism of the PA, existing algorithms cannot be applied [48, 50]. By adapting the algorithm in [48], we obtain two formulations of the DPP, one of which is more computationally efficient as it exploits certain structure in the PA; further details are provided in Remark 6.3.6.

First some notation will be useful. Recall from (6.3), given m ∈ M, Σ_{MA}(m) is the set of all labels σ ∈ Σ on outgoing edges e ∈ E_{MA} starting at m. Similarly, Σ_{vA}(q) is the set of all labels σ ∈ Σ on
outgoing edges \( e \in E_{\pi} \) starting at \( q \). That is,

\[
\Sigma_{\pi}(q) := \{ \sigma \in \Sigma \mid (\exists q' \in Q_{\pi})(q, \sigma, q') \in E_{\pi} \}.
\]

Now we formalize the semantics of the PA. A state of the PA is a pair \( q = (l, m) \in Q_{\pi} \) where \( l \in L_{\text{ots}} \) is a location in the OTS and \( m \in M \) is a motion primitive. A run \( \pi \) of \( A_{\pi} \) is a finite or infinite sequence of states \( \pi = q^0q^1q^2 \ldots \), with \( q^i = (l^i, m^i) \in Q_{\pi} \) and for each \( i \), there exists \( \sigma^i \in \Sigma_{\pi}(q^i) \) such that \( (q^i, \sigma^i, q^{i+1}) \in E_{\pi} \). We define the length of a run to be \( n_{\pi} \); for infinite runs \( n_{\pi} \) is defined to be \( \infty \).

We consider a subset of runs \( \Pi_{\pi}(q) \) starting at \( q \in Q_{\pi} \) that satisfy one further property. If the run \( \pi \) is infinite, then \( \pi \in \Pi_{\pi}(q) \) if \( q^0 = q \). Instead if the run \( \pi \) is finite, then \( \pi \in \Pi_{\pi}(q) \) if \( q^0 = q \) and additionally, \( \Sigma_{\pi}(q^{n_{\pi}}) = \emptyset \). It is the latter requirement – that the last PA state of a finite run may not have outgoing edges in the PA – which is of interest. The interpretation is that we regard the event labels between PA states are uncontrollable, so if any event is possible, then it must occur eventually, meaning that the run cannot be over.

Given \( q \in Q_{\pi} \) and \( \sigma \in \Sigma_{\pi}(q) \), the set of admissible motion primitives is

\[
\mathcal{M}(q, \sigma) := \{ m' \in M \mid (\exists q' = (l', m')) (q, \sigma, q') \in E_{\pi} \}.
\]

More generally, given \( q \in Q_{\pi} \) and \( \Sigma_{\pi}(q) = \{ \sigma_1, \ldots, \sigma_k \} \), the set of admissible motion primitives at \( q \) is

\[
\mathcal{M}(q) := \{ (m_1, \ldots, m_k) \mid m_i \in \mathcal{M}(q, \sigma_i), i = 1, \ldots, k \}.
\]

Next we introduce the notion of a control policy. Given \( q \in Q_{\pi} \) and \( \Sigma_{\pi}(q) = \{ \sigma_1, \ldots, \sigma_k \} \), an admissible control assignment at \( q \) is a vector

\[
c(q) = (c(q, \sigma_1), \ldots, c(q, \sigma_k)),
\]

where \( c(q, \sigma_i) \in \mathcal{M}(q, \sigma_i) \), or equivalently \( c(q) \in \mathcal{M}(q) \). Notice that \( c(q) \) is a vector whose dimension varies as a function of the cardinality of the set \( \Sigma_{\pi}(q) \). An admissible control policy \( c : Q_{\pi} \times \Sigma \rightarrow M \) is a map that assigns an admissible control assignment at each \( q \in Q_{\pi} \). Thus, for each \( q \in Q_{\pi} \) and \( \sigma \in \Sigma_{\pi} \), \( c(q, \sigma) \in \mathcal{M}(q, \sigma) \). The set of all admissible control policies is denoted by \( \mathcal{C} \).

Consider an admission control policy \( c \in \mathcal{C} \) and a state \( q \in Q_{\pi} \). We denote the set of runs in \( \Pi_{\pi}(q) \) induced by \( c \) as \( \Pi_c(q) \). Formally, \( \pi = q^0q^1 \cdots \in \Pi_c(q) \) if \( q^0 = q \), and for all \( i \geq 0 \) and \( i < n_{\pi} \), \( m^{i+1} = c(q^i, \sigma^i) \). Similarly, we denote the subset of runs in \( \Pi_c(q) \) that eventually reach a state in \( Q_{\pi} \) as...
Π^f_c(q). Formally, \( \pi \in \Pi^f_c(q) \) if there exists an integer \( i \in \{0, \ldots, n_\pi \} \) such that \( q^i \in Q^f_{PA} \). For \( \pi \in \Pi^f_c(q) \), we define
\[
r_\pi := \min\{i \in \{0, \ldots, n_\pi \} \mid q^i \in Q^f_{PA}\}.
\]

Now consider the run \( \pi = q^0q^1 \ldots q^{n_\pi} \in \Pi^f_c(q) \) with \( q^0 = q, c(q^i, \sigma^i) = m^{i+1} \), and \( e^i := (q^i, \sigma^i, q^{i+1}) \in E_{PA} \). We define a cost-to-go \( J : Q_{PA} \times C \to \mathbb{R} \) by
\[
J(q,c) = \begin{cases} 
\max_{\pi \in \Pi_c(q)} \left\{ \sum_{j=0}^{r_\pi-1} D_{PA}(e^j) + H_{PA}(q^{r_\pi}) \right\}, & \Pi_c(q) = \Pi^f_c(q) \\
\infty, & \text{otherwise}.
\end{cases}
\]

**Remark 6.3.5.** There are several notable features of our formula for the cost-to-go.

(i) For a given \( \pi \in \Pi_{PA} \), there may be multiple sequences \( \sigma^0\sigma^1 \ldots \) that can be associated to \( \pi \) due to the (second, non-standard type of) non-determinism of the PA. As such, we take the maximum over \( \Pi_c(q) \) in the cost-to-go because of this non-determinacy in \( A_{PA} \): it is uncertain which, among the possibly multiple trajectories allowed by \( c \), that will be taken so we assume the worst-case situation. Moreover, we require \( \Pi_c(q) = \Pi^f_c(q) \) for a finite cost-to-go, otherwise there may exist a run starting at \( q \) and applying control policy \( c \) that does not reach \( Q^f_{PA} \). Also, when \( \Pi_c(q) = \Pi^f_c(q) \), \( r_\pi \) is well-defined.

(ii) We have assumed that finite runs must terminate on PA states that have no outgoing edges. The motivation for this choice becomes clear in light of the formulation of the cost-to-go. Suppose we included in \( \Pi_c(q) \) finite prefixes of (finite or infinite) runs. These necessarily would be finite runs with final PA states that have outgoing edges. Then if we take a finite or infinite run that eventually reaches a goal PA state, certain finite prefixes of that run may not yet have reached a goal PA state, and we would get \( \Pi_c(q) \neq \Pi^f_c(q) \) and an infinite cost-to-go. This anomaly arises from creating an artificial situation in which not all runs starting at an initial PA state reach a goal PA state because we included (unsuccessful) finite prefixes of successful runs.

(iii) The cost-to-go function also accounts for infinite runs by using the variable \( r_\pi \) to record the first time a goal PA state is reached and by taking the cost only over the associated prefix of the infinite run. Allowing infinite runs seems to contradict a reach-avoid specification in which we only want finite runs that terminate on goal PA states with no outgoing edges. The reason we also allow infinite runs in the formulation of the high level plan is that it allows us to extend our framework to a fragment of LTL where, for example, a goal PA state is reached always eventually; see [50] for further details.
Figure 6.4: At the top, a PA is depicted for a single output system having three motion primitives \( M = \{ \mathcal{H}, \mathcal{F}, \mathcal{B} \} \) over three boxes \( L = \{ l_j \mid j = 1, 2, 3 \} \). The numbers \( 1, -1 \in \Sigma \) on the edges (shown as arrows) are the corresponding labels. The bottom pictures show the reduced set of transitions induced by control policies \( c_1, c_2 \in \mathcal{C} \).

**Example 6.3.1.** Consider the PA shown at the top of Figure 6.4 corresponding to a single output system with three motion primitives \( M = \{ \mathcal{H}, \mathcal{F}, \mathcal{B} \} \) over three boxes \( L = \{ l_j \mid j = 1, 2, 3 \} \). The motion primitive \( \mathcal{F} \) causes the output to increase or “move forward”, \( \mathcal{B} \) causes the output to “move backward”, and \( \mathcal{H} \) acts as a hold. Suppose that \( D_{\text{fa}}(e) = 1 \) for all \( e \in E_{\text{fa}} \) and that \( H_{\text{fa}} = 0 \) for all \( q \in Q_{\text{fa}} \).

First consider the feasible control policy \( c_1 \in \mathcal{C} \) with the control assignments: \( c_1(q_1, 1) = \mathcal{F} \), \( c_1(q_3, 1) = \mathcal{B} \), \( c_1(q_5, -1) = \mathcal{F} \), and \( c_1(q_7, -1) = \mathcal{B} \). The bottom left of Figure 6.4 shows how the control policy trims away possible edges in the PA. Now suppose that \( Q_{\text{fa}}^f = \{ q_7 \} \). Choosing the initial condition \( q_1 \in Q_{\text{fa}} \) and under the assumption that we do not include finite runs that terminate at PA states with outgoing edges, we can see that \( \Pi_{c_1}(q_1) \) consists of only the single infinite run \( \pi = q_1 q_3 q_7 q_1 \ldots \). Even though this run is infinite, \( \pi \in \Pi_{c_1}^f(q_1) \), \( r_\pi = 2 \), and \( J(q_1, c_1) = 2 \). Similarly, we compute \( J(q_5, c_1) = 3 \), \( J(q_3, c_1) = 1 \), \( J(q_7, c_1) = 0 \), and \( J(q_2, c_1) = J(q_4, c_1) = J(q_6, c_1) = \infty \).

Next, consider the feasible control policy \( c_2 \in \mathcal{C} \) with the control assignments: \( c_2(q_1, 1) = \mathcal{F} \), \( c_2(q_3, 1) = \mathcal{H} \), \( c_2(q_5, -1) = \mathcal{F} \), and \( c_2(q_7, -1) = \mathcal{B} \). This control policy is shown on the bottom right of Figure 6.4. Suppose that \( Q_{\text{fa}}^f = \{ q_6 \} \). Then we find \( J(q_7, c_2) = 4 \), \( J(q_5, c_2) = 3 \), \( J(q_1, c_2) = 2 \), \( J(q_3, c_2) = 1 \), \( J(q_6, c_2) = 0 \), and \( J(q_2, c_2) = J(q_4, c_2) = \infty \). The difference between \( c_1 \) and \( c_2 \) is that runs are infinite under \( c_1 \) but finite under \( c_2 \).
Finally, suppose we had omitted the extra condition that finite runs must terminate on PA states with no outgoing edges. Considering $c_1 \in \mathcal{C}$ at $q_1 \in Q_{PA}$, then $\Pi_{c_1}(q_1)$ would contain an infinite number of finite runs $q_1, q_1 q_3, q_1 q_3 q_7, \ldots$ as well as the infinite run already noted. In particular, the two runs $q_1, q_1 q_3 \notin \Pi_{c_1}(q_1)$, so by definition of the cost-to-go, we would obtain the undesired result $J(q_1, c_1) = \infty$. A similar problem would arise with the control policy $c_2$.

Next we define the value function $V : Q_{PA} \to \mathbb{R}$ to be

$$V(q) := \min_{c \in \mathcal{C}} J(q, c).$$

The value function satisfies a dynamic programming principle (DPP) that takes into account the non-determinacy of $A_{PA}$; see [48] where a slightly different result is proved. The proof is found in the appendix.

**Theorem 6.3.1.** Consider $q \in Q_{PA} \setminus Q^f_{PA}$ and suppose $|\Sigma_{PA}(q)| > 0$. Then $V$ satisfies

$$V(q) = \min_{c(q) \in \mathcal{M}(q)} \left\{ \max_{\sigma \in \Sigma_{PA}(q)} \left\{ D_{PA}(e) + V(q') \right\} \right\}$$

(6.4)

$$= \max_{\sigma \in \Sigma_{PA}(q)} \left\{ \min_{\tilde{m} \in \mathcal{M}(q, \sigma)} \left\{ D_{PA}(\tilde{e}) + V(\tilde{q}) \right\} \right\},$$

(6.5)

where $q' = (l', c(q, \sigma)) \in Q_{PA}$, $e = (q, \sigma, q') \in E_{PA}$, $\tilde{q} = (\tilde{l}, \tilde{m}) \in Q_{PA}$, and $\tilde{e} = (q, \sigma, \tilde{q}) \in E_{PA}$.

**Proof.** First we prove (6.4). Consider $q = (l, m) \in Q_{PA} \setminus Q^f_{PA}$ and suppose $|\Sigma_{PA}(q)| > 0$. By the definition of $J$, for any $c \in \mathcal{C}$,

$$J(q, c) = \max_{e = (q, \sigma, q') \in E_{PA}} \left\{ D_{PA}(e) + J(q', c) \right\},$$

(6.6)

where $q' = (l', c(q, \sigma)) \in Q_{PA}$. Observe that given $q = (l, m) \in Q_{PA}$ and $\sigma \in \Sigma_{PA}(q)$, there exists a unique $l' \in L_{OTS}$ such that $(l, \sigma, l') \in E_{OTS}$ (since the OTS is a deterministic automaton). Therefore, when we take the maximum over $e = (q, \sigma, q') \in E_{PA}$ in (6.6) with $q' = (l', c(q, \sigma))$, the only free variable to maximize over is $\sigma \in \Sigma_{PA}(q)$. Therefore, (6.6) is equivalent to

$$J(q, c) = \max_{\sigma \in \Sigma_{PA}(q)} \left\{ D_{PA}(e) + J(q', c) \right\},$$

(6.7)

where, as before, $q' = (l', c(q, \sigma)) \in Q_{PA}$ and $e = (q, \sigma, q') \in E_{PA}$. By definition of $V$

$$J(q, c) \geq \max_{\sigma \in \Sigma_{PA}(q)} \left\{ D_{PA}(e) + V(q') \right\}.$$
Again by definition of $V$

$$V(q) = \min_{c \in C} J(q, c) \geq \min_{c \in C} \left\{ \max_{\sigma \in \Sigma_{PA}(q)} \{ D_{PA}(e) + V(q') \} \right\}.$$ 

Thus,

$$V(q) \geq \min_{\tilde{c}(q) \in M(q)} \left\{ \max_{\sigma \in \Sigma_{PA}(q)} \{ D_{PA}(e) + V(q') \} \right\}.$$ 

To prove the reverse inequality, suppose by the way of contradiction that there exists an admissible control assignment at $q \in Q_{PA}$, $\hat{c}(q) \in M(q)$, such that

$$V(q) > \max_{\sigma \in \Sigma_{PA}(q)} \{ D_{PA}(\hat{e}) + V(\hat{q}) \} \geq \min_{\hat{c}(q) \in M(q)} \left\{ \max_{\sigma \in \Sigma_{PA}(q)} \{ D_{PA}(e) + V(q') \} \right\},$$

where $\hat{q} = (\hat{l}, \hat{c}(q, \sigma)) \in Q_{PA}$, $\hat{e} = (q, \sigma, \hat{q}) \in E_{PA}$, $q' = (l', c(q, \sigma)) \in Q_{PA}$, and $e = (q, \sigma, q') \in E_{PA}$. Suppose the maximum for $\hat{c}(q)$ is achieved with $\sigma^* \in \Sigma_{PA}(q)$. We define $q^* = (l^*, \hat{c}(q, \sigma^*))$, and $e^* = (q, \sigma^*, q^*) \in E_{PA}$. Then

$$V(q) > D_{PA}(e^*) + V(q^*).$$

Suppose an admissible optimal control policy for $q^*$ to achieve $V(q^*)$ is $c^* \in C$. Define a new policy $c = c^*$ on $Q_{PA} \setminus \{q\}$ and $c(q) = \hat{c}(q)$. Then

$$J(q, c) = \max_{\sigma \in \Sigma_{PA}(q)} \{ D_{PA}(e) + V(q') \} = D_{PA}(e^*) + V(q^*) < V(q),$$

a contradiction. Hence, it must be that

$$V(q) \leq \min_{\tilde{c}(q) \in M(q)} \left\{ \max_{\sigma \in \Sigma_{PA}(q)} \{ D_{PA}(e) + V(q') \} \right\},$$

as desired. This proves (6.4).

Second we prove (6.5). Consider $q = (l, m) \in Q_{PA} \setminus Q_{PA}^f$ and suppose $|\Sigma_{PA}(q)| > 0$. Let $\tilde{c}(q) \in M(q)$ be an admissible control assignment at $q$ such that for all $\sigma \in \Sigma_{PA}(q)$,

$$\tilde{c}(q, \sigma) \in \arg\min_{\tilde{m} \in M(q, \sigma)} \{ D_{PA}(\tilde{e}) + V(\tilde{q}) \},$$

(6.8)
where \( \bar{q} = (\bar{l}, \bar{m}) \in Q_{PA} \), and \( \bar{e} = (q, \sigma, \bar{q}) \in E_{PA} \). We will show that

\[
\max_{\sigma \in \Sigma_{PA}(q)} \{ D_{PA}(\bar{e}) + V(\bar{q}) \} \leq \min_{c(q) \in M(q)} \left\{ \max_{\sigma \in \Sigma_{PA}(q)} \{ D_{PA}(e) + V(q') \} \right\}, \tag{6.9}
\]

where \( \bar{q} = (\bar{l}, \bar{c}(q, \bar{\sigma}^*) \in Q_{PA}, \bar{e} = (q, \sigma, \bar{q}) \in E_{PA}, q' = (l', c(q, \sigma)) \in Q_{PA}, \) and \( e = (q, \sigma, q') \in E_{PA} \). Suppose the minimum and maximum on the r.h.s. are achieved with \( c^*(q) \in M(q) \) and \( \sigma^* \in \Sigma_{PA}(q) \). Also, suppose the maximum on the l.h.s. is achieved with \( \bar{\sigma}^* \in \Sigma_{PA}(q) \). Then (6.9) becomes

\[
D_{PA}(\bar{e}^*) + V(\bar{q}^*) \leq D_{PA}(e^*) + V(q^*), \tag{6.10}
\]

where \( \bar{q}^* = (\bar{l}^*, \bar{c}(q, \bar{\sigma}^*)) \in Q_{PA}, \bar{e}^* = (q, \bar{\sigma}^*, \bar{q}^*) \in E_{PA}, q^* = (l^*, c^*(q, \sigma^*)) \in Q_{PA}, \) and \( e^* = (q, \sigma^*, q^*) \in E_{PA} \). Suppose by way of contradiction that (6.9) does not hold. Then (6.10) does not hold. That is,

\[
D_{PA}(\bar{e}^*) + V(\bar{q}^*) > D_{PA}(e^*) + V(q^*).
\]

By the maximality of \( \sigma^* \in \Sigma_{PA}(q) \) we have

\[
D_{PA}(e^*) + V(q^*) = D_{PA}((q, \sigma^*, (l^*, c^*(q, \sigma^*)))) + V((l^*, c^*(q, \sigma^*))) \geq D_{PA}((q, \bar{\sigma}^*, (\bar{l}^*, c^*(q, \bar{\sigma}^*)))) + V((\bar{l}^*, c^*(q, \bar{\sigma}^*))).
\]

Therefore

\[
D_{PA}(\bar{e}^*) + V(\bar{q}^*) = D_{PA}((q, \bar{\sigma}^*, (\bar{l}^*, c(q, \bar{\sigma}^*))))) + V((\bar{l}^*, c(q, \bar{\sigma}^*)))) > D_{PA}((q, \bar{\sigma}^*, (\bar{l}^*, c^*(q, \bar{\sigma}^*))))) + V((\bar{l}^*, c^*(q, \bar{\sigma}^*)))).
\]

This contradicts the definition of \( \bar{c}(q, \bar{\sigma}^*) \) in (6.8). We conclude that (6.9) must hold.

Now consider \( \bar{e}^* \in \mathcal{C} \) such that \( \bar{c}^*(q) = \bar{c}(q) \) and \( \bar{e}^* \) is any admissible optimal control policy for \( q' \neq q \).
Using (6.7) we have

\[
J(q, \bar{c}^*) = \max_{\sigma \in \Sigma_{\mathcal{P}\mathcal{A}(q)}} \{D_{\mathcal{P}\mathcal{A}}(\bar{e}) + J(\bar{q}, \bar{c}^*)\} \\
= \max_{\sigma \in \Sigma_{\mathcal{P}\mathcal{A}(q)}} \{D_{\mathcal{P}\mathcal{A}}(\bar{e}) + V(\bar{q})\},
\]

where \(\bar{q} = (\bar{l}, \bar{c}^*(q, \sigma)) = (\bar{l}, \bar{c}(q, \sigma)) \in Q_{\mathcal{P}\mathcal{A}}\) and \(\bar{e} = (q, \sigma, \bar{q}) \in E_{\mathcal{P}\mathcal{A}}.\) Now by (6.4) and (6.9) we have

\[
J(q, \bar{c}^*) = \max_{\sigma \in \Sigma_{\mathcal{P}\mathcal{A}(q)}} \{D_{\mathcal{P}\mathcal{A}}(\bar{e}) + V(\bar{q})\} \leq \min_{c(q) \in \mathcal{M}(q)} \left\{ \max_{\sigma \in \Sigma_{\mathcal{P}\mathcal{A}(q)}} \{D_{\mathcal{P}\mathcal{A}}(e) + V(q')\} \right\} = V(q).
\]

By definition of \(V,\) we obtain that \(J(q, \bar{c}^*) = V(q).\) That is,

\[
J(q, \bar{c}^*) = V(q) = \max_{\sigma \in \Sigma_{\mathcal{P}\mathcal{A}(q)}} \{D_{\mathcal{P}\mathcal{A}}(\bar{e}) + V(\bar{q})\} \tag{6.11}
\]

with \(\bar{q} = (\bar{l}, \bar{c}(q, \sigma)) \in Q_{\mathcal{P}\mathcal{A}}\) and \(\bar{e} = (q, \sigma, \bar{q}) \in E_{\mathcal{P}\mathcal{A}}.\) However, we know \(\bar{c}^*(q) = \bar{c}(q),\) and \(\bar{c}(q)\) satisfies (6.8). Therefore,

\[
V(q) = \max_{\sigma \in \Sigma_{\mathcal{P}\mathcal{A}(q)}} \left\{ \min_{\bar{m} \in \mathcal{M}(q, \sigma)} \{D_{\mathcal{P}\mathcal{A}}(\bar{e}) + V(\bar{q})\} \right\},
\]

where now \(\bar{q} = (\bar{l}, \bar{m}) \in Q_{\mathcal{P}\mathcal{A}},\) and this proves (6.5).

Notice that for all \(q \in Q_{\mathcal{P}\mathcal{A}} \setminus Q^f_{\mathcal{P}\mathcal{A}}\) such that \(|\Sigma_{\mathcal{P}\mathcal{A}}(q)| = 0,\) \(V(q) = \infty\) (since there can be no paths to the goal). Also, for all \(q \in Q^f_{\mathcal{P}\mathcal{A}},\) \(V(q) = H_{\mathcal{P}\mathcal{A}}(q).\)

Remark 6.3.6. In (6.4) of Theorem 6.3.1, it is shown that \(V(q)\) can be computed using the local information of \(\mathcal{M}(q)\) instead of using all of \(\mathcal{C}.)\) In (6.5), the result is taken one step further by showing that \(V(q)\) can be calculated using only \(\mathcal{M}(q, \sigma)\) for each \(\sigma \in \Sigma_{\mathcal{P}\mathcal{A}}(q).\) The benefit of (6.5) becomes clear when we compare the cardinality of the sets over which the minimizations occur. Given \(q \in Q_{\mathcal{P}\mathcal{A}},\) let \(\Sigma_{\mathcal{P}\mathcal{A}}(q) = \{\sigma_1, \ldots, \sigma_k\}.\) In (6.4) the minimization is over \(\mathcal{M}(q),\) and therefore the cardinality of the minimization set is \(\prod_{i=1}^k |\mathcal{M}(q, \sigma_k)|.\) In (6.5) the minimization is over \(\mathcal{M}(q, \sigma)\) for each \(\sigma \in \Sigma_{\mathcal{P}\mathcal{A}}(q),\) and therefore the cardinality of the set is \(|\mathcal{M}(q, \sigma)|.\) While both (6.4) and (6.5) can be used to compute \(V(q),\) in general (6.5) will be more computationally efficient.
Figure 6.5: This figure shows a discrete control strategy for the scenario shown in Figure 6.1.

**Corollary 6.3.2.** Consider the control policy \( c^\ast \) such that for all \( q \in Q_{PA} \), and \( \sigma \in \Sigma_{PA}(q) \)

\[
c^\ast(q,\sigma) \in \operatorname{argmin}_{m' \in M(q,\sigma)} \{ D_{PA}(e) + V(q') \},
\]

where \( q' = (l',m') \), and \( e = (q,\sigma,q') \). Then \( c^\ast \) is an optimal control policy such that for all \( q \in Q_{PA} \),

\[ V(q) = J(q,c^\ast). \]

Figure 6.5 shows a possible control policy for the scenario in Figure 6.1. Notice that different routes may be taken from the same product state depending on the face reached, but ultimately the control policy ensures that all paths lead to the goal.

### 6.4 Main Results

In this section we present our main results on a solution to Problem 6.2.1. Our final result combines the notion of a control policy at the high level with feedback controllers executing correct continuous time behavior at the low level. First, in accordance with the reach-avoid objective (see Remark 6.3.5(iii)), we restrict the final PA states to be goal OTS states equipped with motion primitives that have no guard sets

\[
Q_{PA}^f = \{(l,m) \in L_{OTS}^0 \times M \mid \Sigma_{MA}(m) = \emptyset\}. \tag{6.12}
\]

Now suppose we have an admissible control policy \( c \in C \) derived using Theorem 6.3.1 or otherwise with \( Q_{PA}^f \) as above. We present a complete solution to Problem 6.2.1 including an initial condition set \( X^0 \subset \mathbb{R}^n \), a feedback control \( u(x) \), and conditions on the motion primitives so that the reach-avoid specifications of Problem 6.2.1 are met.
First we specify the initial condition set $A^0$. The set of feasible initial PA states is

$$Q^0_{\text{PA}} := \{ q \in Q_{\text{PA}} \mid \Pi_c(q) = \Pi^f_c(q) \}.$$ 

That is, a feasible initial PA state satisfies that every run (induced by the control policy) starting at the PA state eventually reaches a goal PA state. The associated set of feasible initial locations of the OTS is

$$L^0 := \{ l \in L_{\text{OTS}} \mid (\exists m \in M) (l, m) \in Q^0_{\text{PA}} \}.$$ 

Now consider a state $x_0 \in \mathbb{R}^n$. It can be used as an initial state of the system (6.1) to solve Problem 6.2.1 if the following are met:

- The output $h(x_0)$ lies in some box $Y_j$ of the output space whose associated OTS location $l_j$ belongs to $L^0$. This condition guarantees that there exists a motion primitive $m \in M$ such that all runs starting from $(l_j, m)$ reach a goal PA state. Moreover, if $x_0$ is shifted so that its corresponding output lies in $Y^*$, then the invariance condition of $m$ is satisfied.

- Because of the ambiguity arising from the output lying in more than one box, we place an extra condition that $h(x_0) \in \text{int} (Y_j)$. This condition can be easily relaxed by adding a box selection criterion; we omit the details.

Recall that for all $y \in \mathbb{R}^p$ and $j \in \{1, \ldots, n_L\}$, $y \in Y^*$ if and only if $y + d \circ l_j \in Y_j$. With this in mind, we define the set of initial states to be:

$$A^0 = \bigcup_{l_j \in L^0} \left\{ (x + h_\alpha^{-1}(d \circ l_j)) \in \mathbb{R}^n \mid h(x) \in \text{int} (Y^*), \right. $$

$$\left. (\exists m \in M)(l_j, m) \in Q^0_{\text{PA}}, x \in I_{\text{stat}}(m) \right\}. \quad (6.13)$$

Notice this initial condition set takes into account the requirements listed above.

Next we specify the feedback controllers to solve Problem 6.2.1. Consider any $q = (l_j, m) \in Q^0_{\text{PA}}$. Then for all $x \in \mathbb{R}^n$ such that $x - h_\alpha^{-1}(d \circ l_j) \in I_{\text{stat}}(m)$, we define the feedback

$$u(x, q) := u_m(x - h_\alpha^{-1}(d \circ l_j)). \quad (6.14)$$

This defines a family of feedback controllers parametrized by $x$, the state of (6.1) and by the PA state $q = (l_j, m)$. These feedbacks work in tandem with the control policy $c \in \mathcal{C}$, which effectively determines the next feasible PA state $q' \in Q^0_{\text{PA}}$. For example, suppose $q = (l_j, m) \in Q^0_{\text{PA}}$ and suppose the label
σ ∈ Σ is measured. This event corresponds to x ∈ g_e for e = (m, σ, c(q, σ)) ∈ E_{MA}. Let m' := c(q, σ) and let l' ∈ L_{OTS} be the unique location of the OTS such that (l, σ, l') ∈ E_{OTS}. Then the next PA state is q' = (l', m') ∈ Q_{PA}^0 and the controller that is applied in the next location l' ∈ L_{OTS} is u_{m'}(·).

The main result of the chapter is the following.

Theorem 6.4.1. Consider the system (6.1) satisfying Assumption 6.2.1, the non-empty feasible set \( P \subset \mathbb{R}^p \), and the goal set \( O \subset P \). Let \( d \) be the vector of box lengths such that the goal indices \( I^0 \) is non-empty. Consider an associated OTS \( A_{OTS} \), an MA \( A_{MA} \) satisfying Assumption 6.3.1, a PA \( A_{PA} \) with \( Q_{PA}^f \) as in (6.12), and an admissible control policy \( c \in C \). Then the initial condition set \( X_0 \) given in (6.13) and the feedback controllers (6.14) solve Problem 6.2.1.

In the remainder of this section we prove Theorem 6.4.1. First we give a roadmap for these results. The verification of correctness at the low level is broken down into several theoretical steps that we now describe. First, we show that the MA is non-blocking in Lemma 6.4.2. The key requirements are summarized in Assumption 6.3.1. The non-blocking condition ensures that MA trajectories continually evolve in time and stay within the invariant regions. We also put conditions to avoid chattering in which two discrete transitions can occur in immediate succession. While physical systems never undergo infinite switching in finite time, if our model predictions diverge from reality, then we have no grounds to claim that Problem 6.2.1 is indeed solved. In Lemma 6.4.3 we can show that to each closed-loop trajectory of (6.1) under the feedback controllers (6.14) and a control policy \( c \in C \), we can associate a unique execution of the MA (defined below) and run of the PA.

We begin by describing the semantics of the MA. These definitions are standard; see [60]. A state of the MA is a pair \((m, x)\), where \( m \in M \) and \( x \in \mathbb{R}^n \). Trajectories of the MA are called executions and are defined over hybrid time domains that identify the time intervals when the trajectory of a hybrid system is in a fixed motion primitive \( m \in M \). Precisely, a hybrid time domain of the MA is a finite or infinite sequence of intervals \( \tau = \{I_0, \ldots, I_n \} \), such that

(i) \( I_i = [\tau_i, \tau'_i] \), for all \( 0 \leq i < n_\tau \),

(ii) if \( n_\tau < \infty \), then either \( I_n_\tau = [\tau_n_\tau, \tau'_n_\tau] \) or \( I_n_\tau = [\tau_n_\tau, \tau'_n_\tau] \),

(iii) \( \tau_i \leq \tau'_i = \tau_{i+1} \), for all \( 0 \leq i < n_\tau \).

Definition 6.4.1. An execution of the MA is a collection \( \chi = (\tau, (m(\cdot), \phi_{MA}(\cdot, x_0)) \) such that

(i) the initial condition of the execution satisfies: \((m(0), x_0) \in Q_{MA}^0 \).
(ii) the continuous evolution of the execution satisfies: for all \( i \in \{0, \ldots, n_\tau \} \) with \( \tau_i < \tau_i' \), then for all \( t \in [\tau_i, \tau_i'] \), \( m(\cdot) \) is constant and \( \frac{d}{dt} \phi_{MA}(t, x_0) = f(\phi_{MA}(t, x_0), u_{m(\cdot)}(\phi_{MA}(t, x_0))) \), while for all \( t \in [\tau_i, \tau_i'] \), \( \phi_{MA}(t, x_0) \in I_{MA}(m(t)) \).

(iii) a discrete transition of the execution satisfies: for all \( i \in \{0, \ldots, n_\tau - 1 \} \), there exists \( \sigma_i \in \Sigma_{MA}(m(\tau_i')) \) such that \( (m(\tau_i'), \sigma_i, m(\tau_{i+1})) =: e_i \in E_{MA}, \phi_{MA}(\tau_i', x_0) \in g_{e_i}, \) and \( \phi_{MA}(\tau_{i+1}, x_0) = r_{e_i}(\phi_{MA}(\tau_i', x_0)) \).

Given an execution \( \chi = (\tau, m(\cdot), \phi_{MA}(\cdot, x_0)) \), we associate to it the \textit{output trajectory of the MA} given by \( y_{MA}(\cdot, x_0) := h(\phi_{MA}(\cdot, x_0)) \) (the subscript MA is included to avoid confusion with output trajectories \( y(\cdot, x_0) \) of the physical system (6.1) which do not undergo resets). The \textit{execution time} of an execution \( \chi \) is defined as \( \mathcal{T}(\chi) := \sum_{i=0}^{n_\tau} (\tau_i' - \tau_i) = \lim_{n_{\tau} \to \infty} n_{\tau} \tau_i' \). An execution is called \textit{finite} if \( \tau \) is a finite sequence ending with a compact time interval. An execution is called \textit{infinite} if either \( \tau \) is an infinite sequence or if \( \mathcal{T}(\chi) = \infty \). Finally, an execution is called \textit{Zeno} if it is infinite but \( \mathcal{T}(\chi) < \infty \).

\textbf{Remark 6.4.1.} There are two types of Zeno behavior. In one type that we call chattering, transitions are instantaneous. The second more subtle type is when the times between discrete transitions of the MA converge to zero, but the transitions are not instantaneous. Assumptions 6.3.1 (i) and (iv) ensure that we cannot have chattering. True Zeno behavior with convergent transition times is more difficult to identify in the setting when the MA is formed as a parallel composition. Fortunately, for our reach-avoid objective, the induced MA executions cannot be Zeno since there are a finite number of transitions by construction, see Lemma 6.4.3.

\textbf{Definition 6.4.2.} The MA is \textit{non-blocking} if for all \( (m(0), x_0) \in Q_{MA}^0 \), the set of all infinite executions of the MA with initial condition \( (m(0), x_0) \) is non-empty.

\textbf{Lemma 6.4.2.} Under Assumption 6.3.1, the MA is non-blocking.

\textit{Proof.} Let \( (m, x) \in Q_{MA}^0 \). If \( \Sigma_{MA}(m) = \emptyset \), then by Assumption 6.3.1(vi), \( I_{MA}(m) \) is invariant, so the trajectory \( \phi_{MA}(t, x) \) starting at \( (m, x) \) remains in \( I_{MA}(m) \) for all future time. Therefore, trivially, the MA is non-blocking for this initial condition. If \( \Sigma_{MA}(m) \neq \emptyset \), then by Assumption 6.3.1(vii), \( \phi_{MA}(t, x) \) remains in \( I_{MA}(m) \) until it reaches a guard set. Additionally, by Assumption 6.3.1 (v), the trajectory is mapped under the reset into the next invariant. By Lemma 1 of [60], the MA is again non-blocking for this initial condition. Overall, the MA is non-blocking. \( \square \)

The purpose of the Assumptions 6.3.1 is to guarantee consistency between low level continuous time behavior and the high level discrete plan. This consistency is formalized by way of a one-to-one
correspondence between infinite MA executions and finite PA runs, both starting from the same initial condition. The proof is found in the appendix.

**Lemma 6.4.3.** Suppose we have an admission control policy \( c \in C \), and we have an MA satisfying Assumption 6.3.1. For each \((l^0, m^0) \in Q^0_{\text{pa}}\) and \(x_0 \in I_{\text{ma}}(m^0)\) there exist a unique infinite MA execution \( \chi = (r, m(\cdot), \phi_{\text{ma}}(\cdot, x_0)) \) and a unique finite PA run \( \pi = q^0 q^1 \ldots q^N \).

**Proof.** Let \((l^0, m^0) \in Q^0_{\text{pa}}\) and \(x_0 \in I_{\text{ma}}(m^0)\). The initial MA state of the MA execution is \((m(0), x_0) = (m^0, x_0) \in Q^0_{\text{ma}}\), and the initial PA state of the PA run is \( q^0 = (l^0, m^0) \). The hybrid time domain of \( \chi \) will be denoted as \( \tau = \{ I_k \}_{k=0}^{n} \).

Suppose \( \Sigma_{\text{ma}}(m^0) = \emptyset \). Then by Assumption 6.3.1(vi), \( \phi_{\text{ma}}(t, x_0) \in I_{\text{ma}}(m^0) \) for all \( t \in I_0 := [0, \infty) \). The complete PA run is \( \pi = q^0 \). Suppose instead that \( \Sigma_{\text{ma}}(m^0) \neq \emptyset \). Assumption 6.3.1(vii) tells us that there exist unique \( \sigma^0 \in \Sigma_{\text{ma}}(m^0) \) and \( T^0 \geq 0 \) such that for all \( t \in I_0 := [0, T^0] \), \( \phi_{\text{ma}}(t, x_0) \in I_{\text{ma}}(m^0) \). Also, for each \( e = (m^0, \sigma^0, m^1) \in E_{\text{ma}} \), there exists a guard set \( g_e \) such that \( \phi_{\text{ma}}(T, x_0) \in g_e \). Assumptions 6.3.1(ii) tells us that for all such \( m^1 \), the guard set is the same. Then with Assumption 6.3.1(iii), we deduce there is a unique guard set reached by \( \phi_{\text{ma}}(\cdot, x_0) \). Now we invoke the control policy to select a specific \( m^1 \); namely, let \( m^1 := c(q^0, \sigma^0) \) and \( e^0 := (m_0, \sigma^0, m^1) \in E_{\text{ma}} \). Then as above, \( \phi_{\text{ma}}(T^0, x_0) \in g_e \). The execution time so far is \( T^0 \). Define \( x_0^1 := r_{e^0}(\phi_{\text{ma}}(T^1, x_0)) \). By Assumption 6.3.1(iv), \( x_0^1 \notin g_e \) for any \( e = (m^0, \sigma^0, m^1) \in E_{\text{ma}} \). The next PA state is \( q^1 = (l^1, m^1) \), where \( l^1 \in L_{\text{ots}} \) is uniquely determined through \((l^0, \sigma^0, l^1) \in E_{\text{ots}} \), by the determinism of the OTS. The PA run so far is \( \pi = q^0 q^1 \).

We now construct the remainder of the MA execution and the PA run by induction. The base case has been established above. Suppose that the run so far is \( \pi = q^0 \ldots q^k \), where \( q^i = (l^i, m^i) \) for \( i = 0, \ldots, k \). Suppose \( \Sigma_{\text{ma}}(m^k) = \emptyset \). Then by Assumption 6.3.1(vi), \( \phi_{\text{ma}}(t, x_0^k) \in I_{\text{ma}}(m^k) \) for all \( t \in I_k := [T^k, \infty) \). The complete PA run is \( \pi = q^0 \ldots q^k \). Suppose instead \( \Sigma_{\text{ma}}(m^k) \neq \emptyset \). Then by similar arguments as above, there exist unique \( \sigma^k \in \Sigma_{\text{ma}}(m^k) \) and \( T^k \geq 0 \) such that for all \( t \in I_k := [T^k, \infty) \), \( \phi_{\text{ma}}(t, x_0^k) \in I_{\text{ma}}(m^k) \). Moreover, as above we assign \( m^{k+1} := c(q^k, \sigma^k) \) so that \( e^k := (m^k, \sigma^k, m^{k+1}) \in E_{\text{ma}} \) and \( \phi_{\text{ma}}(T^k, x_0^k) \in g_{e^k} \). The execution time so far is \( T^k + \sum_{j=0}^{k} T^j \). Define \( x_0^{k+1} := r_{e^k}(\phi_{\text{ma}}(T^k, x_0)) \). The next PA state is \( q^{k+1} = (l^{k+1}, m^{k+1}) \), where \( l^{k+1} \in L_{\text{ots}} \) is uniquely determined through \((l^k, \sigma^k, l^{k+1}) \in E_{\text{ots}} \), by the determinism of the OTS. The PA run so far is \( \pi = q^0 \ldots q^{k+1} \).

The above inductive process is guaranteed to terminate with a finite PA run by definition of \( Q^0_{\text{pa}} \). That is, since \((l^0, m^N) \in Q^0_{\text{pa}}\) there will be a smallest \( N \) such that \((l^N, m^N) \in Q^0_{\text{pa}} \). Moreover, by definition of \( Q^I_{\text{pa}} \) (6.12), we have that \( \Sigma_{\text{ma}}(m^N) = \emptyset \) and so the run cannot be extended further. The
resulting MA execution is infinite with a finite number of intervals in the hybrid time domain $\tau$, and it is non-blocking by Lemma 6.4.2. \hfill \Box

Before we can prove Theorem 6.4.1 we need one further preliminary result stating that because of the translational invariance of Assumption 6.2.1, the continuous part of an MA execution has a unique correspondence to a closed-loop trajectory of the system (6.1). The proof is straightforward and is omitted.

**Lemma 6.4.4.** Let $m \in M$, $x_0 \in I_{\text{sl}}(m)$, $y \in \mathbb{R}^p$, and $\tilde{x}_0 = x_0 + h_0^{-1}(y)$. Consider the trajectory $\phi(t, \tilde{x}_0)$ of (6.1) with the feedback control $u(x) = u_m(x - h_0^{-1}(y))$. Also consider the MA trajectory $\phi_{\text{sl}}(t, x_0)$ with feedback control $u_m(x)$. For all $t \geq 0$ such that $\phi_{\text{sl}}(t, x_0) \in I_{\text{sl}}(m)$,

$$\phi(t, \tilde{x}_0) = \phi_{\text{sl}}(t, x_0) + h_0^{-1}(y).$$

Finally we are ready to prove Theorem 6.4.1.

**Proof of Theorem 6.4.1.** We must show that (i) output trajectories of system (6.1) remain within $\mathcal{P}$, and (ii) output trajectories eventually reach and remain within the goal set $\mathcal{O}$. Let $\tilde{x}_0 \in \mathcal{X}^0$. There is a unique initial box $l_{\rho}^0$ such that $h(\tilde{x}_0) \in \text{int}(Y_{l_{\rho}})$. We define $x_0 := \tilde{x}_0 - h_0^{-1}(d \circ l_{\rho}^0)$ such that $h(x_0) \in \text{int}(Y^*)$. Now choose any $m^0 \in M$ such that $(l_{\rho}^0, m^0) \in Q^0_{\text{sl}}$. By Lemma 6.4.3, we may associate a unique MA execution $\chi$ and a unique PA run $\pi$ to $(l_{\rho}^0, m^0) \in Q^0_{\text{sl}}$ and $x_0 \in I_{\text{sl}}(m^0)$. Denote the hybrid time domain as $\tau = \{\mathcal{I}_0, \ldots, \mathcal{I}_N\}$ with $\mathcal{I}_k = [\tau_k, \tau_k']$ for $k = 0, \ldots, N - 1$ (with $\tau_0 = 0$) and $\mathcal{I}_N = [\tau_N, \infty)$. The last interval follows from the definition of $(l_{jN}, m^N) \in Q^1_{\text{sl}}$ (6.12), since $\Sigma_{\text{sl}}(m^N) = \emptyset$ and thus Assumption 6.3.1(vi) implies that we must have that $\mathcal{I}_N = [\tau_N, \infty)$. As in the proof of Lemma 6.4.3, denote the corresponding sequence of events as $\sigma^0, \ldots, \sigma^{N-1}$.

Using Lemma 6.4.4 with $y = d \circ l_{\rho}^0$, we have that $\phi(t, \tilde{x}_0) = \phi_{\text{sl}}(t, x_0) + h_0^{-1}(d \circ l_{\rho}^0)$. We claim that for all $k = 0, \ldots, N$ and $t \in \mathcal{I}_k$,

$$\phi(t, \tilde{x}_0) = \phi_{\text{sl}}(t, x_0) + h_0^{-1}(d \circ l_{jh}).$$

(6.15)

Clearly the result is true for $k = 0$.

We derive two facts to assist in proving this claim. Recall that by definition of the OTS edges, we have that for all $k = 0, \ldots, N - 1$, $\sigma^k = l_{j+1} - l_{jh}$. Furthermore, by rearranging, multiplying component-wise by $d$, and taking the preimage $h_0^{-1}$, we have the first fact: for all $k = 0, \ldots, N - 1$ that $h_0^{-1}(d \circ l_{j+1}) = h_0^{-1}(d \circ l_{jh}) + h_0^{-1}(d \circ \sigma^k)$. Also by definition of the reset map and MA execution, we get
the second fact: for all $k = 0, \ldots, N - 1$, $r_k(\phi_{\text{MA}}(\tau_k', x_0)) = \phi_{\text{MA}}(\tau_k', x_0) - h_{\circ}^{-1}(d \circ \sigma^k) = \phi_{\text{MA}}(\tau_{k+1}, x_0)$.

Returning to (6.15), by induction we assume that it is true for $0 \leq k < N$ and show that it is true for $k + 1$. Using the above facts and (6.15) for $k$ at $t = \tau_k' = \tau_{k+1}$ yields

$$
\phi(\tau_{k+1}, \tilde{x}_0) = \phi(\tau_k', \tilde{x}_0) = \phi_{\text{MA}}(\tau_k', x_0) + h_{\circ}^{-1}(d \circ l_{j_k}) = (\phi_{\text{MA}}(\tau_{k+1}, x_0) + h_{\circ}^{-1}(d \circ \sigma^k)) + h_{\circ}^{-1}(d \circ l_{j_k}) = \phi_{\text{MA}}(\tau_{k+1}, x_0) + h_{\circ}^{-1}(d \circ l_{j_{k+1}}).
$$

Applying Lemma 6.4.4 with $y = h_{\circ}^{-1}(d \circ l_{j_{k+1}})$ at the new initial condition $\phi_{\text{MA}}(\tau_{k+1}, x_0) \in I_{\text{MA}}(m^{k+1})$, we have that for $k + 1$ and for all $t \in \mathcal{I}_{k+1}$ that (6.15) holds. When $k + 1 = N$, the induction terminates and the claim is proven.

Using (6.15) and projecting to the output space we conclude that for all $k = 0, \ldots, N$ and $t \in \mathcal{I}_k$, $y(t, \tilde{x}_0) \in Y_{j_k}$. Since all the boxes are contained in $\mathcal{P}$ by construction, then for all $t \geq 0$ we have (i). Moreover, since $l_{j_N} \in L_{\text{OTS}}^\theta$ implies the goal box $Y_{j_N}$ is contained in $\mathcal{O}_\mathcal{P}$ and $\mathcal{I}_N = [\tau_N, \infty)$, we have (ii).

Remark 6.4.2. The above result does not depend on the method of construction of the admissible control policy $c \in \mathcal{C}$, nor does it require the control policy to be optimal. This allows for different path planning techniques on the PA such as offline non-deterministic Dijkstra methods as well as real-time non-deterministic A* methods [62].

### 6.5 Motion Primitives for Integrator Systems

A motion primitive is a canonical behavior in the output space that is realized by way of a closed-loop vector field in the state space. In turn, the closed-loop vector field is formed by invoking a pre-computed feedback controller associated with the motion primitive and applying it to the system (6.1). Typical motion primitives for robotic agents operating in a 2D workspace are Right, Left, Up, and Down.

Our design of motion primitives exploits three simplifications. First, we invoke Assumption 6.2.1 and the fact that all the boxes in the OTS are translations of each other to be able to design motion primitives over the canonical box $Y^*$ only. Second, we focus our design of motion primitives to robotic agents with either single or double integrator dynamics in each output coordinate. The latter simplification is justified by the fact that in many robotic applications, the system (6.1) is feedback linearizable to single or double integrator dynamics, or it is assumed a priori that such dynamics comprise the only relevant
behavior. Assumption 6.2.1 is satisfied for this model. The third simplification is that we design atomic motion primitives for a single output coordinate.

In what follows we will present atomic motion primitives for a single component of the output; that is, we take $p = 1$. For $p > 1$, we use the parallel composition of the design for $p = 1$. For both the single and double integrator, we consider an MA consisting of three atomic motion primitives, Hold ($\mathcal{H}$), Forward ($\mathcal{F}$), and Backward ($\mathcal{B}$). We use reach control theory to design these motion primitives [9, 1]. Details are given below.

### 6.5.1 Single Integrator

Suppose the nonlinear control system is the single integrator system:

$$
\dot{x}_1 = u_1, \quad y = x_1,
$$

(6.16)

where $x_1 \in \mathbb{R}$, $u_1 \in \mathbb{R}$, and the output $y$ is the position. In this case, $Y^* = [0, d]$. Equivalently $Y^* = \text{co} \{v_1^1, v_2^1\}$, the convex hull of two vertices $v_1^1 = 0$ and $v_2^1 = d$. Our goal is to design an affine feedback of the form

$$
u_m^1(x) = K_m^1 x + g_m^1,$$

(6.17)

for each motion primitive $m \in \{\mathcal{H}, \mathcal{F}, \mathcal{B}\} := M_{\mathcal{H}, \mathcal{F}, \mathcal{B}}$. If we assume that $u_1^*$ is the maximum control effort, then we choose $K_{\mathcal{H}}^1 = -2u_1^*/d$, $K_{\mathcal{B}}^1 = K_{\mathcal{F}}^1 = 0$, $g_{\mathcal{H}}^1 = g_{\mathcal{F}}^1 = u_1^*$, and $g_{\mathcal{B}}^1 = -u_1^*$. The behavior of these controllers is exhibited in the closed-loop vectors fields shown in Figure 6.7. The exit facets of each motion primitive, which are also the guard sets, are shown in green. The closed-loop behavior corresponds to our requirements: trajectories for motion primitive $\mathcal{H}$ remain within $[0, d]$ for all $t \geq 0$; trajectories for motion primitive $\mathcal{B}$ leave $[0, d]$ through $x_1 = 0$; and trajectories of motion primitive $\mathcal{F}$ leave $[0, d]$ through $x_1 = d$. 

![Figure 6.6: The maneuver automaton for the single integrator dynamics with $p = 1$. There are three motion primitives: Hold ($\mathcal{H}$), Forward ($\mathcal{F}$), and Backward ($\mathcal{B}$).](image-url)
Chapter 6. Motion Planning Framework for Integrator Systems

The MA is depicted in Figure 6.6. Its state space is $Q_{MA} = M_{HF,FB} \times \mathbb{R}$. The labels are $\Sigma = \{-1, 0, 1\}$. The set of augmented edges $E_{MA}$ are shown in Figure 6.6. The vector fields are given by $X_{MA}(m) = u_1^1 m(\cdot)$ as in (6.17). The invariant region for each $m \in M_{HF,FB}$ is $I_{MA}(m) = [0, d]$. The enabling conditions are constructed as follows. For an edge $e = (m, 1, m') \in E_{MA}$, $g_e = \{d\}$; for an edge $e = (m, -1, m') \in E_{MA}$, $g_e = \{0\}$. The reset conditions are then based entirely on the label $\sigma \in \Sigma$. It is quite trivial to prove that these motion primitives satisfy Assumption 6.3.1, so we omit the proof.

**Lemma 6.5.1.** The single integrator MA satisfies Assumption 6.3.1.

### 6.5.2 Double Integrator

Suppose the nonlinear control system is the double integrator system:

$$
\dot{x}_1 = x_2, \quad \dot{x}_2 = u_2, \quad y = x_1,
$$

(6.18)

where $x := (x_1, x_2) \in \mathbb{R}^2$, $u_2 \in \mathbb{R}$, and the output $y$ is the position. Each motion primitive’s invariant region is a polytopic set in the state space defined as the convex hull of vertices $v_k^2$, $k \in \{1, \ldots, 6\}$; see Figure 6.9. The vertices are determined by the segment length $d > 0$, and a pre-specified maximum control value $u_2^* > 0$. Let $\bar{u}_1 := \sqrt{d u_2^*}$. The vertices are $v_2^1 = (0, -\bar{u}_1)$, $v_2^2 = (0, 0)$, $v_2^3 = (0, \bar{u}_1)$, $v_2^4 = (d, -\bar{u}_1)$, $v_2^5 = (d, 0)$, and $v_2^6 = (d, \bar{u}_1)$. For each motion primitive $m \in M_{HF,FB}$, we define an affine feedback

$$
u_m^2(x) = K_m^2 x + g_m^2.
$$

(6.19)

Our specific choices are:

$$
K_{HF}^2 = \begin{bmatrix} 0 & -2u_2^*/\bar{u}_1 \end{bmatrix},

K_{FB}^2 = \begin{bmatrix} 0 & -2u_2^*/\bar{u}_1 \end{bmatrix},

K_{HF}^2 = \begin{bmatrix} -2u_2^*/d & -2u_2^*/\bar{u}_1 \end{bmatrix},

g_{HF}^2 = g_{FB}^2 = u_2^*, g_{FB}^2 = -u_2^*.
$$
Figure 6.8: The maneuver automaton for the double integrator dynamics with $p = 1$. There are three motion primitives: Hold ($H$), Forward ($F$), and Backward ($B$).

These controllers are derived using reach control theory [9, 1]. One first selects control values at the vertices of the polytopes so that trajectories remain in the invariant region (for the Hold primitive) or they exit the polytope through a certain facet and not through others; see Figure 6.9. Then the velocity vectors at the vertices are affinely extended to obtain affine feedbacks over the entire polytope, yielding the vector fields shown in Figure 6.10.

Now we construct the MA. The state space is $Q_{\text{MA}} = M_{\mathcal{B}, \mathcal{F}, \mathcal{H}} \times \mathbb{R}^2$. The labels are $\Sigma = \{-1, 0, 1\}$. The set of augmented edges $E_{\text{MA}}$ are shown in Figure 6.8. The augmented edges without the label 0 gives the set of edges $E_{\text{MA}}$, that is,

$$E_{\text{MA}} = \{(\mathcal{F}, 1, \mathcal{H}), (\mathcal{F}, 1, \mathcal{F}), (\mathcal{B}, -1, \mathcal{H}), (\mathcal{B}, -1, \mathcal{B})\}.$$  

The vector fields are given by $X_{\text{MA}}(m) = u^2_m(\cdot)$ as in (6.19), which are clearly globally Lipschitz. The invariants are given by the convex hull of vertices, as seen in Figure 6.9, and excluding the two points $(0, 0)$ and $(d, 0)$, so the invariants are clearly bounded. For example, $I_{\text{MA}}(\mathcal{H}) = \text{co} \{v^2_2, v^2_6\} \setminus \{(0, 0),(d, 0)\}$.

The enabling conditions are constructed by taking the convex hull of vertices of the exit facet and excluding again $(0, 0)$ or $(d, 0)$. Specifically, the edges $(\mathcal{F}, 1, \mathcal{H}), (\mathcal{F}, 1, \mathcal{F}) \in E_{\text{MA}}$ both have guard sets $g_e = \text{co} \{v^2_2, v^2_6\} \setminus \{(d, 0)\} = \{d\} \times \{0\}$, as shown highlighted in green on the invariant region of $\mathcal{F}$ in Figure 6.10, whereas $(\mathcal{B}, -1, \mathcal{H}), (\mathcal{B}, -1, \mathcal{B}) \in E_{\text{MA}}$ both have guard sets $g_e = \text{co} \{v^2_2, v^2_6\} \setminus \{(0, 0)\} = \{0\} \times \{-u_1\}$. The reset conditions are constructed according to their definition.

**Lemma 6.5.2.** The double integrator MA satisfies Assumption 6.3.1.

**Proof of Lemma 6.5.2.** (i) We must show that for all $m \in M_{\mathcal{H}, \mathcal{F}, \mathcal{B}}$, $0 \not\in \Sigma_{\text{MA}}(m)$. This is clearly true since there is no edge in $E_{\text{MA}}$ containing the label 0.

(ii) We must show that for all $e_1, e_2 \in E_{\text{MA}}$ such that $e_1 = (m_1, \sigma, m_2)$ and $e_2 = (m_1, \sigma, m_3)$, $g_{e_1} = g_{e_2}$. This is clearly true since we have designed $g_{(\mathcal{F}, 1, \mathcal{H})} = g_{(\mathcal{F}, 1, \mathcal{F})}$ and $g_{(\mathcal{B}, -1, \mathcal{H})} = g_{(\mathcal{B}, -1, \mathcal{B})}$. 
(iii) We must show that for all \( e_1, e_2 \in E_{\text{MA}} \) such that \( e = (m_1, \sigma_1, m_2) \) and \( e_2 = (m_1, \sigma_2, m_3) \), if \( \sigma_1 \neq \sigma_2 \), then \( g_{e_1} \cap g_{e_2} = \emptyset \). This is trivially true since for all \( m \in M_{\mathcal{P}, \mathcal{F}, \mathcal{H}}, |\Sigma_{\text{MA}}(m)| < 2 \).

(iv) We must show that for all \( e_1, e_2 \in E_{\text{MA}} \) such that \( e_1 = (m_1, \sigma_1, m_2) \) and \( e_2 = (m_2, \sigma_2, m_3) \), \( r_{e_1}(g_{e_1}) \cap g_{e_2} = \emptyset \). Using Assumption 6.3.1 (ii) above, we only need to check two cases, that is, \( e_1 = e_2 = (\mathcal{F}, 1, \mathcal{F}) \) and \( e_1 = e_2 = (\mathcal{B}, -1, \mathcal{B}) \). Both cases satisfy the condition because of the reset action on the first coordinate; for example, the first case gives \( r_{e_1}([d] \times (0, \bar{u}_1)) \cap [d] \times (0, \bar{u}_1) = \emptyset \).

(v) We must show that for all \( e = (m_1, \sigma, m_2) \in E_{\text{MA}}, r_e(g_e) \subset I_{\text{MA}}(m_2) \). This is easily verified for all four edges in \( E_{\text{MA}} \), for example, if \( e = (\mathcal{F}, 1, \mathcal{H}) \), clearly \( r_{e_1}([d] \times (0, \bar{u}_1)) \subset I_{\text{MA}}(\mathcal{H}) \).

(vi) We must show that for all \( m \in M_{\mathcal{P}, \mathcal{F}, \mathcal{H}}, \) if \( \Sigma_{\text{MA}}(m) = \emptyset \) then \( I_{\text{MA}}(m) \) is invariant. We have that only \( \Sigma_{\text{MA}}(\mathcal{H}) = \emptyset \). As can be seen in Figure 6.10, the closed-loop vector field does not allow trajectories to exit \( I_{\text{MA}}(\mathcal{H}) \), and therefore for all \( x_0 \in I_{\text{MA}}(\mathcal{H}) \), and for all \( t \geq 0, \phi_{\text{MA}}(t, x_0) \in I_{\text{MA}}(\mathcal{H}) \).

(vii) We must show that for all \( m \in M, \) if \( \Sigma_{\text{MA}}(m) \neq \emptyset \) then \( I_{\text{MA}}(m) \) forces all trajectories to exit in finite time through some guard. Consider \( \mathcal{F} \) with \( \Sigma_{\text{MA}}(\mathcal{F}) = \{1\} \). As can be seen in Figure 6.10, for all \( x_0 \in I_{\text{MA}}(\mathcal{F}) \), there exists \( T \geq 0 \) such that for all \( t \in [0, T], \phi_{\text{MA}}(t, x_0) \in I_{\text{MA}}(\mathcal{F}), \) and \( \phi_{\text{MA}}(T, x_0) \in [d] \times (0, \bar{u}_1) \). Since both \( g_{(\mathcal{F}, 1, \mathcal{F})} = g_{(\mathcal{F}, 1, \mathcal{F})} = [d] \times (0, \bar{u}_1) \), the assumption holds. A similar argument can be made for \( \mathcal{B}. \)

\[ \square \]

Remark 6.5.1. We noted in Remark 6.4.1 that Zeno executions do not arise for reach-avoid specifications that, by construction, involve only finite MA executions. However, one may be interested in analyzing whether an MA is non-Zeno in its own right, independently of the high level plan or control specification for which it is used. It can be verified rather easily that for \( p = 1 \), the single and double integrator MA designs we have presented above are non-Zeno. The situation is considerably more complicated when considering an MA that is a parallel composition of these MA’s or when considering an arbitrary MA. Generic conditions when hybrid systems have a Zeno execution have been studied in [54, 55, 56]. However, further study of this problem is needed in our context since existing results do not apply to all the situations that can arise in our MA.

Remark 6.5.2. In this section we have given polytopic sets and closed-loop vector fields on those sets to achieve \( \mathcal{B}, \mathcal{F}, \) and \( \mathcal{H} \). But we did not justify how these sets were obtained. One may ask whether Algorithm 5.3.1 of Chapter 5 is applicable. And how it related to the present results. In fact, Figure 5.1f provides the solution for the forward \( \mathcal{F} \) motion primitive based on Algorithm 5.3.1. The final polytope in Figure 5.1f is similar to the polytope we give in Figure 6.9. The only difference is that the polytope in Figure 6.9 is simpler - it has two fewer vertices at the bottom. Generally, the polytopes presented in this section can be derived from the algorithm in Chapter 5 after simplification. We chose to work
with simpler polytopes as these will be amenable to a generalization to higher dimensions in the next chapter. In contrast, the goal of Algorithm 5.3.1 was to find a maximal polytope, with no consideration for its complexity.

6.5.3 Triple Integrator

Suppose the nonlinear control system is the double integrator system:

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_3, \\
\dot{x}_3 &= u_3
\end{align*}
\]

(6.20)

where \( x := (x_1, x_2, x_3) \in \mathbb{R}^3 \), \( u_3 \in \mathbb{R} \), and the output \( y \) is the position. Each motion primitive’s invariant region is a polytopic subset constructed with vertices \( v^k_3 \), \( k \in \{1, \ldots, 10\} \), see Figure 6.11. The vertices are determined by the segment length \( d > 0 \), and maximum control value \( u^*_3 > 0 \), and variables \( \bar{u}_2 = (\sqrt{d}u^*_3/2)^{2/3} \), \( \bar{u}_1 = \sqrt{d}\bar{u}_2 \). Specifically, \( v^1_3 = (0, -\bar{u}_1, \bar{u}_2) \), \( v^2_3 = (0, 0, -\bar{u}_2) \), \( v^3_3 = (0, 0, 0) \), \( v^4_3 = (0, 0, \bar{u}_2) \), \( v^5_3 = (0, \bar{u}_1, -\bar{u}_2) \), \( v^6_3 = (d, -\bar{u}_1, \bar{u}_2) \), \( v^7_3 = (d, 0, -\bar{u}_2) \), \( v^8_3 = (d, 0, 0) \), \( v^9_3 = (d, 0, \bar{u}_2) \), and \( v^{10}_3 = (d, \bar{u}_1, -\bar{u}_2) \).
For each motion primitive, denoted as $m \in \mathcal{M}_{\bar{2}, \bar{3}, \bar{4}}$, we employ affine feedbacks

$$ u^3_m(x) = \begin{cases} 
K^3_{m,1}x + g^3_{m}, & x_2 < 0 \\
K^3_{m,2}x + g^3_{m}, & x_2 \geq 0
\end{cases} $$

(6.21)

We set $K^3_{\bar{2},1} = K^3_{\bar{3},2} = \begin{bmatrix} -2u^*_3/d & -(5/2)(u^*_3/\bar{u}_1) & -2u^*_3/\bar{u}_2 \end{bmatrix}$, $K^3_{\bar{2},1} = K^3_{\bar{3},2} = \begin{bmatrix} 0 & -2u^*_2/\sqrt{du^*_2} \end{bmatrix}$, $K^3_{\bar{3},1} = \begin{bmatrix} 0 & -(1/2)(u^*_3/\bar{u}_1) & -2u^*_3/\bar{u}_2 \end{bmatrix}$, $g^3_{\bar{2}} = g^3_{\bar{3}} = u^*_3$, and $g^2_{\bar{3}} = -u^*_3$. As in the case of the double integrator dynamics, these controllers can be derived by solving reach control problems on a triangulation of the polytopic subsets [9, 49] and imposing continuity constraints. The behaviour of these controllers is exhibited by the closed-loop vectors at the vertices shown in Figure 6.11.

**Lemma 6.5.3.** The triple integrator MA satisfies Assumption 6.3.1.

The proof follows closely that of Lemma 6.5.2. To give more insight into the construction of these polytopic subsets, as well as their closed-loop behaviour we further analyze the Forward mode for the triple integrator. The Forward mode is constructed from the union of three convex polytopes as shown.
Figure 6.11: The closed-loop vectors at the vertices for the Hold, Forward, and Backward motion primitives for triple integrator dynamics.
in Figure 6.12. Facets of the polytope which have each of their edges highlighted in green are exit facets, that is the facets from which the closed-loop trajectories originating within the polytope can leave. If a facet has at least one edge which is marked in black, that facet is blocked, and hence closed-loop trajectories cannot exit through it. From this we can see that the overall closed-loop behaviour of the forward mode causes trajectories to eventually enter the rightmost polytope in Figure 6.12. Once the trajectory enters this polytope, it can only exit through the exit facet given by \( \{v_8^3, v_9^3, v_{10}^3\} \). Since this face lies on the plane \( x_1 = d \), exiting through the face is associated with the event \( \sigma = 1 \).

Each of the polytopic subsets are designed precisely so that the closed-loop vector fields over them can be pieced together and faithfully represent the discrete transitions defined in the MA shown in Figure 6.6. For example, the transition from Forward to Hold, \((F, 1, H) \in E_{\text{MA}}\), is possible because when the trajectory leaves \( F \) through the exit facet (highlighted in green) on Figure 6.11, it activates the enabling and reset conditions given by \((F, 1, H)\). This resets the trajectory from \( \{v_8^3, v_9^3, v_{10}^3\} \) to \( \{v_3^3, v_4^3, v_5^3\} \) which is contained in \( H \).

The \( p \)-dimensional MA is obtained by parallel composing the single output MA shown in Figure 6.6 \( p \)-times, a standard operation on automata. The three single-output motion primitives, denoted as \( M_{\mathbb{H}, \mathcal{F}, \mathcal{H}} \), give rise to \( 3^p \) \( p \)-dimensional motion primitives, denoted as \( \prod_{i=1}^{p} M_{\mathbb{H}, \mathcal{F}, \mathcal{H}} \). For example, \((\mathcal{F}, \mathcal{H})\) would implement the motion primitive Right in 2D. The new set of labels for the transitions is given by \( \Sigma \) in \( p \)-dimensions. The composed transitions are obtained by considering all possible combinations of one-dimensional transitions in each output.
6.6 Applications

In this section, we apply our methodology both to motion planning problems for robotic manipulators, as well as quadrocopters. We begin with the robotic manipulators.

6.6.1 Robotic Manipulator Modeling

We consider planar robotic manipulators with two revolute joints. The end effector dynamics of such manipulators are known to be feedback linearizable [61]. The dynamics for the position \((x_e, y_e)\) of the end effector in the inertial frame after feedback linearization are given by two decoupled double integrators. Thus, we are able to use the atomic motion primitives \(B, F, H\) based on the double integrator model given by in Section 6.5. Note that since the output we are trying to control is given by the end effector position of each robot, \(p = 2j\) for \(j\) robotic manipulators. We therefore take the parallel composition of \(2j\) double integrator MAs to get the full MA for \(j\) robotic manipulators. For simplicity, we assume that the robotic manipulators do not have any width associated with their links, and that the obstacles are represented as polytopes.

6.6.2 Preparing the OTS

In order to ensure that the robotic manipulator does not come in contact with an obstacle, another robot, or itself, we remove locations in the OTS which correspond to these undesirable situations. In order to determine if a location in the OTS is unsafe, we perform a line segment intersection tests between:

- the links of each robotic manipulator and the links of every other robotic manipulator.
- the links of each robotic manipulator and each face of an obstacle.
- the links of each robotic manipulator and the boundaries of the workspace.

In order to ensure that the robot does not reach a singular configuration, we remove any locations in the OTS where the robot can be fully extended, or folded onto itself.

6.6.3 Learning Real Time A*

Since in application robotic manipulators require precision for their end effector location, the size of the boxes in the OTS may need to be small. This results in a large number of locations in the OTS, and in turn, the PA will have a large number of states. Because of this, it may be infeasible to employ offline path planning algorithms on the PA such as non-deterministic Dijkstra. We instead employ a real-time
algorithm called Learning Real Time A* (LRTA*) [62]. This algorithm begins by initializing the value function of each node $q \in Q_{pa}$ with a heuristic function $\text{Heur}(q)$, and then determines the next motion primitive based on a one step value function, as in (6.5). The algorithm is presented below.

**Algorithm 1** Algorithm for LRTA*

1: For all $q \in Q_{pa}$, $V(q) = \text{Heur}(q)$.
2: Take $q = (l, m) \in Q_{pa}$, the current state.
3: Execute motion primitive $m$. As a result, an event $\sigma \in \Sigma_{pa}(q)$ occurs. Let $l^* \in L_{ots}$ such that $(l, \sigma, l^*) \in E_{ots}$.
4: $m^* := \arg\min_{m' \in M(q, \sigma)} \{V(q')\}$.
5: $V(q) := \max\{V(q), \min_{m' \in M(q, \sigma)} \{V(q')\}\}.$
6: $q := (l^*, m^*)$, return to 2.

This algorithm ensures that if a goal state can be reached through any resolution of the non-determinism, it will be reached. For our application we use two different heuristic functions. Given $q = (l, m) \in Q_{pa}$, the first heuristic function, $\text{Heur}_1(l, m)$, is calculated as the sum of the number of boxes between the current box $l$ and the goal box in each output dimension. This heuristic does not take into account any obstacles, or collisions between the robotic manipulators, but can be calculated almost instantaneously. The second heuristic, $\text{Heur}_2(l, m)$, is calculated using a shortest path algorithm on the OTS. That is, given a state $(l, m)$ of the PA, we initialize the value function at $(l, m)$ with the value function of $l$ determined by a deterministic Dijkstra on the OTS. This heuristic takes into account obstacles and collisions between robotic manipulators, but may not be feasible if the OTS has many states and edges.

### 6.6.4 Robotic Manipulator Experimental Results

We perform three different experiments. The first experiment, shown in Figure 6.13, consists of a single robotic manipulator in which the objective is for the end effector to reach the goal location marked with dashed lines. The robotic manipulator must avoid wall obstacles, as well as avoid any singular configurations. The length of both links of the manipulator is 7 meters, and the workspace is partitioned into a $(15 \times 10)$ grid consisting of boxes having a width and length of 1 meter. We employ the second heuristic $\text{Heur}_2(l, m)$ since the size of the OTS is small. It can be seen that the robotic manipulator avoids the obstacle and reaches the goal location.

The second experiment, shown in Figure 6.14, consists of two robotic manipulators in which the objective is for the end effector to reach a goal box on the opposite side of the room as its base. In effect, they are trying to exchange end effector locations. Although there are no obstacles, the robotic manipulators must not collide with each other, or go beyond the boundaries of the room. The length
of both links of each manipulator is 3 meters, and the workspace is partitioned into a (100 × 40) grid consisting of boxes having a width and length of 0.1 meters. We employ the first heuristic Heur$_1(l, m)$ since the gridding is fine, and therefore the size of the OTS is large. The configuration of the robotic manipulators at different times is shown in Figure 6.14, where the grey lines show the trajectories of the end effectors. It can be seen that since collisions between manipulators is not taken into account with Heur$_1(l, m)$, the robotic manipulator on the right begins by moving away from its goal location. Over the timespan of 50 seconds, the two manipulators are not able to exchange end effector locations and therefore the objective is not achieved.

The third experiment, shown in Figure 6.15, is the same experiment as above except we employ the second heuristic Heur$_2(l, m)$. Although the computation of Heur$_2(l, m)$ is more demanding than that of Heur$_1(l, m)$, the calculation only needs to be done once. Also, since Heur$_2(l, m)$ takes into account collisions between manipulators, the desired objective is achieved in only 11 seconds.

6.6.5 Quadrocopter Modeling

The standard quadrocopter dynamics model is ubiquitous in the literature; see, for example, [58]. The vehicle dynamics are described by six degrees of freedom and are nonlinear. It is well known that this model is differentially flat [44]. As a result, the dynamics for the position $(x_w, y_w, z_w)$ in the world frame each reduce to a double integrator. Thus, we are able to use the motion primitive designs based on a double integrator model from Section 6.5.
Chapter 6. Motion Planning Framework for Integrator Systems

Figure 6.14: Two robotic manipulators attempt to exchange end effector positions while avoiding collisions using a LRTA* algorithm. Since the heuristic function \( \text{Heur}_2(l, m) \) does not take into account collisions between manipulators, the objective is not achieved after 50 seconds.

Figure 6.15: Two robotic manipulators exchange end effector positions while avoiding collisions using a LRTA* algorithm. Since the heuristic function \( \text{Heur}_1(l, m) \) takes into account collisions between manipulators, after 11 seconds the goal location is reached.

6.6.6 Interfacing Multiple Quadrocopters

In this section, we explain how to apply our methodology to perform a joint reach-avoid task among multiple quadrocopters. First, we discuss how to model a multi-vehicle system in our framework. Then
we highlight the offline and runtime aspects of our framework.

In the case of one quadrocopter, we grid the three-dimensional workspace into boxes to construct the OTS, assign a copy of the MA from Section 6.5 to each output $x_w, y_w,$ and $z_w,$ and construct the overall MA by parallel composition. To apply our framework to $N$ quadrocopters, a copy of the 3D workspace gridding must be associated with each vehicle, resulting in a total of $p = 3N$ outputs. The $p$-dimensional MA representing the asynchronous motion capabilities of the multi-vehicle system is obtained by parallel composing the single-output MA for each output.

The labelling of obstacle and goal boxes in $p = 3N$ dimensions requires a preliminary step in the case of multiple quadrocopters. Assuming a single 3D goal box for each vehicle, the corresponding $p$-dimensional goal box for the multi-vehicle goal is easily computed. Obstacle boxes must reflect real-world obstacles and pairwise collisions between two vehicles. Since obstacles are abstracted to $p$-dimensional boxes, a simple, exhaustive search over each discrete multi-vehicle configuration suffices to determine obstacle labels. A margin of safety is included to account for the quadrocopter size relative to the box size.

At this point, the multi-vehicle reach-avoid problem can be solved using our proposed methodology, following the steps shown in Figure 6.2. The output of the methodology is a hybrid controller, which is fully computed offline.

Once the hybrid controller is computed, the system can execute the reach-avoid task from any starting configuration. The task can be successfully completed if the starting configuration belongs to a valid initial condition of the hybrid controller. The runtime workflow is depicted in Figure 6.16, showing how the hybrid controller interfaces with the multi-vehicle system. Due to the simplicity of a box partition and assuming that the next motion primitive can be determined in constant time, each component requires a negligible amount of computation, resulting in sufficiently fast runtime performance even for a large number of vehicles and outputs.

### 6.6.7 Quadrocopter Experimental Results

Our experimental platform is the Parrot AR.Drone 2.0 interfaced with the ROS ardrone autonomy package. We used an external motion capture system to obtain the state estimates for our feedback controller (6.19), which was run at 70 Hz.

Due to limited space, we illustrate one interesting scenario where two quadrocopters must coordinate switching places through a narrow passage, see Figure 6.17. We constrain motion to the horizontal plane for both vehicles, but enable the use of non-deterministic motion primitives to exhibit simultaneous
Figure 6.16: Interface between multiple vehicles and the abstract framework with $p = 3N$ outputs. The internal state is updated via external state measurements (assumed to be given). The hybrid controller internal state consists of the joint state measurement of all the vehicles; this includes the current (joint) box, $l$, and the current (joint) motion primitive, $m$. The vehicle state is again used to compute the feedback controls.

motion. Additional video examples include a simple, single quadrocopter maneuver, see http://tiny.cc/quadrocopterPlanar, and several 2D and 3D scenarios with two quadrocopters, see http://tiny.cc/quad5scenes.

For the example shown here, the 3D workspace is partitioned into a $(5 \times 6 \times 1)$ grid consisting of boxes having side lengths of 1, 0.75, and 3 meters, respectively. The planar $(x_w,y_w)$-view is shown in Figure 6.17, where the red boxes represent the physical obstacles forming the passage. Note that the $z_w$-direction, consisting of just one box, can be stabilized using the Hold mode, so the overall number of outputs is effectively $p = 4$. For this scenario, it is too difficult to visualize the high-level control strategy, but it is analogous to the one shown for $p = 2$ in Figure 6.5. The total time to compute the discrete strategy was about 5 seconds.

A nominal experimental run is shown in Figure 6.17, depicting the $(x_w,y_w)$-trajectories of each quadrocopter. The trajectories are divided into several time slots in order to show how the vehicles coordinate moving through the passage. As a consequence of our non-deterministic motion primitives, both vehicles move simultaneously for most of the run. Notice that one vehicle (Vehicle 2 in this case) moves into a corner and waits to give space for the other vehicle to pass through the passage in order to maintain a sufficient safety margin.

In contrast, an experimental run with disturbances is shown in Figure 6.18. An Ecohouzng 16 High Velocity Air Circulator fan was used to simulate wind disturbances. Starting from the same configuration as the nominal run, Vehicle 2’s motion was delayed due to a persistent wind disturbance, see the annotation. As such, the quadrocopters automatically executed a significantly different path
Figure 6.17: Experimental results for the scenario where two quadrocopters must switch places through a narrow passage, where Vehicle 1 (blue) starts on the left and Vehicle 2 (orange) starts on the right. The grayed out trajectories show the full path followed by the vehicles; the color portion corresponds to the indicated time interval. The maneuver is shown over three time segments, with goals highlighted in green on the right. We observe that since Vehicle 2 gets to the passage first, it makes room and waits for Vehicle 1 to pass first. Noisiness in the trajectories is due to mutual aerodynamic effects, to be contrasted with Figure 6.19. Although these effects are not accounted for, our hybrid feedback controller safely completes the task.

Figure 6.18: The same scenario as in Figure 6.17 above, but now an unmodelled wind disturbance causes a delay in Vehicle 2. To compensate, Vehicle 1 makes room and waits for Vehicle 2 to pass first, opposite to the nominal case. The same hybrid controller was used as in the nominal run of Figure 6.17.
Figure 6.19: A single-vehicle run for a similar scenario as considered in Figures 6.17 and 6.18. The trajectory is smooth and efficient, which illustrates that the noisy trajectories in the multi-vehicle case arise from mutually caused aerodynamic effects. The blue trajectory represents the motion for the given time range from 0 to 15 seconds.

compared to the nominal run (Vehicle 1 moved into a corner this time). In Figures 6.17 and 6.18 mutual aerodynamic effects resulted in “wobbly” trajectories, which can be contrasted to the single vehicle run in Figure 6.19.

As long as disturbances are not too severe, the underlying feedback-based motion primitives will compensate for it and, if necessary, the high-level strategy will guide the vehicles along a new path to the goal configuration. Since our precomputed hybrid control strategy consists of a feedback at both the discrete high-level of motion primitive assignment and at the continuous low-level over each box, we achieve a robust maneuver that does not require any online recomputation. Notice that there is no notion of time or timed trajectories in our approach.
This chapter develops an algorithm to design atomic motion primitives as in the Motion Planning chapter for an $n$-integrator system whose output $y$ is its first coordinate. Suppose $Y^* = [0, d]$ is an interval in the output space. The three atomic motion primitives that we wish to design are \textit{Hold (H)}, \textit{Forward (F)}, \textit{Backward (B)} which have the following behaviours respectively: $y$ remains in $[0, d]$, $y$ leaves the interval $[0, d]$ at the face $y = d$, $y$ leaves $[0, d]$ at $y = 0$. We now begin formalizing the above discussion.

Consider the $n$-th order integrator system $\dot{x} = f_n(x, u)$

$$
\begin{align*}
\dot{x}_1 &= x_2 \\
& \vdots \\
\dot{x}_{n-1} &= x_n \\
\dot{x}_n &= u \\
y &= x_1,
\end{align*}
$$

(7.1)

where $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}$ is the input, and $y \in \mathbb{R}$ is the output. Let $\phi(\cdot, x_0)$ and $y(\cdot, x_0)$ denote the state and output trajectories of (7.1) starting at initial condition $x_0 \in \mathbb{R}^n$ and under some control $u$.

If $W \subset \mathbb{R}$ is a set in the output space, we define the pre-image of $W$ with respect to the output map to be

$$y^{-1}(W) := \{ x \in \mathbb{R}^n \mid x_1 \in W \}.$$  

(7.2)
Let $Y^* := [0, d]$, $d > 0$ be the non-empty subset of the output space over which the motion primitives are defined. Given $\mathcal{P} \subset \mathbb{R}^n$, and a continuous piecewise affine feedback $u(x)$,

$$
\text{Out}(\mathcal{P}) = \{ x_0 \in \mathcal{P} \mid \exists T \geq 0, \gamma > 0 \text{ s.t. } \phi(T, x_0) \in \partial(\mathcal{P}) \text{ and } \phi(t, x_0) \notin \mathcal{P}, \forall t \in (T, T + \gamma) \} \quad (7.3)
$$
denotes the set of initial conditions in $\partial(\mathcal{P})$ for which trajectories immediately leave $\mathcal{P}$. Since we are designing the motion primitives over $Y^*$ to be used in a so called maneuver automaton (MA), we need reset conditions to ensure that the trajectories of the MA remain within $Y^*$. Given $X \subset \mathbb{R}^n$, let $r_F(X) = \{ x - (d, 0, \ldots, 0) \mid x \in X \}$ and $r_B(X) = \{ x + (d, 0, \ldots, 0) \mid x \in X \}$ denote the respective reset maps for the Forward and Backward motion primitives on the set $X$. There are no reset conditions for the Hold motion primitive.

**Problem 7.0.1.** We are given the system (7.1) and the non-empty output subset $Y^* = [0, d] \subset \mathbb{R}$. Construct polytopes $\mathcal{P}_H^n, \mathcal{P}_F^n, \mathcal{P}_B^n \subset \mathbb{R}^n$, and associated feedback controllers $u_H(x), u_F(x), u_B(x)$ such that:

(i) **Hold:** for every $x_0 \in \mathcal{P}_H^n$ and for all $t \geq 0$, $y(t, x_0) \in Y^*$,

(ii) **Forward:** for every $x_0 \in \mathcal{P}_F^n$ there exists $T \geq 0$ and $\gamma > 0$ such that $y(t, x_0) \in Y^*$ for all $t \in [0, T]$, $y(T, x_0) = d$, and $y(t, x_0) \notin Y^*$ for all $t \in (T, T + \gamma)$,

(iii) **Backward:** for every $x_0 \in \mathcal{P}_B^n$ there exists $T \geq 0$ and $\gamma > 0$ such that $y(t, x_0) \in Y^*$ for all $t \in [0, T]$, $y(T, x_0) = 0$, and $y(t, x_0) \notin Y^*$ for all $t \in (T, T + \gamma)$,

Problem 7.0.1(i)-(iii) define the desired behaviour of the motion primitives $\mathcal{H}, \mathcal{F}, \text{ and } \mathcal{B}$ to be constructed. These motion primitives will form the discrete components the MA shown in Figure 7.1. Problem 7.0.1(iv) imposes additional conditions on the structure of the polytopes $\mathcal{P}_H^n, \mathcal{P}_F^n, \mathcal{P}_B^n$ to ensure these transitions are well defined. That is, at the continuous level, trajectories of the MA faithfully represent the transitions in Figure 7.1. Problem 7.0.1(iv)a) ensures that transitions from $\mathcal{H}$ to both $\mathcal{F}$ and $\mathcal{B}$ are well defined, while Problem 7.0.1(iv)b) ensures that transitions from $\mathcal{F}$ and $\mathcal{B}$ to $\mathcal{H}$ are well defined.

### 7.1 Preliminaries

In this section we provide the preliminary background material and terminology which will be used to solve Problem 7.0.1.
Consider an $\kappa$-dimensional polytope

$$ P = \text{co}\{v_1, \ldots, v_q\} $$

with vertex set $V := \{v_1, \ldots, v_q\}$ and facets $\mathcal{F}_0, \ldots, \mathcal{F}_r$, where $\mathcal{F}_0$ is the exit facet of $P$. Consider the hyperplane representation of $P$

$$ P = \{x \in \mathbb{R}^\kappa \mid h_i \cdot x \leq \bar{h}_i, i \in 0, \ldots, r\}, $$

where $h_i$ is the unit normal vector of the facet $\mathcal{F}_i$ pointing outside the polytope. Define the index sets $I := \{1, \ldots, q\}, J := \{1, \ldots, r\}$, and $J(x) := \{j \in J \mid x \in \mathcal{F}_j\}$. That is, $J(x)$ is the set of indices of the restricted facets in which $x$ is a point. For each $x \in P$, define the closed convex cone

$$ C(x) := \{\bar{x} \in \mathbb{R}^\kappa \mid h_j \cdot \bar{x} \leq 0, j \in J(x)\}. \quad (7.4) $$

Consider two vertices $v, w \in \mathbb{R}^\kappa$ of a polytope $P$. We would like to define a notion of the closed-loop vector field at $v$ pointing along the vector from $v$ to $w$. Since the vector $w - v$ points inside the polytope at $v$, the invariance conditions must hold at $v$. Given $u \in \mathbb{R}$, the notation

$$ f_\kappa(v, u) \to w \iff \exists \tau > 0 \text{ s.t. } w = v + \tau f_\kappa(v, u). \quad (7.5) $$

Recall in Chapter 2 we introduced the notion of a Bouligand tangent cone and Nagumo’s Theorem (Theorem 2.3.3). Also recall that in Chapter 5 we introduced the definition of a projection of polytopes (Definition 5.2.1) and how faces of polytopes map through projections of polytopes (Lemma 5.2.2).

Finally, we present Barbalat’s Lemma, a standard result from nonlinear control theory.
Lemma 7.1.1. [83] If the differentiable function \( f(t) \) has a finite limit as \( t \to \infty \), and is such that \( \dot{f} \) exists and is bounded, then \( \dot{f}(t) \to 0 \) as \( t \to \infty \).

7.2 \( \mathcal{H} \) Motion Primitive

We begin by solving Problem 7.0.1(i) and do so by induction. That is, we assume that we have a polytope \( \mathcal{P}_{\mathcal{H}}^\kappa \) of dimension \( \kappa \) which solves Problem 7.0.1(i), and we input \( \mathcal{P}_{\mathcal{H}}^\kappa \) into an algorithm to construct a polytope \( \mathcal{P}_{\mathcal{H}}^{\kappa+1} \) of dimension \( \kappa + 1 \) which solves Problem 7.0.1(i) and continue until \( \kappa = n \). We then provide a polytope which initializes the induction process.

Lemma 7.2.1. Given a polytope \( \mathcal{P} = \text{co}\{v_1, \ldots, v_q\} = \{x \in \mathbb{R}^\kappa \mid h_i \cdot x \leq \bar{h}_i, i = 0, \ldots, r\} \), a vertex \( v \in \{v_1, \ldots, v_q\} \), an input \( u \in \mathbb{R} \), and \( w \in \mathcal{P} \), if \( f_\kappa(v, u) \to w \) then \( f_\kappa(v, u) \in \mathcal{C}(v) \).

Proof. Since \( f_\kappa(v, u) \to w \) there exists \( \tau > 0 \) such that \( w = v + \tau f_\kappa(v, u) \). Let \( k \in J(v) \), that is \( h_k \cdot v = \bar{h}_k \). Since \( w \in \mathcal{P} \), for all \( i \in \{1, \ldots, r\} \), \( h_i \cdot w \leq \bar{h}_i \), and hence \( h_k \cdot w \leq \bar{h}_k \). Therefore

\[
    h_k \cdot f_\kappa(v, u) = (1/\tau)(h_k \cdot (w - v))
    = (1/\tau)(h_k \cdot w - h_k \cdot v)
    = (1/\tau)(h_k \cdot w - \bar{h}_k)
    \leq 0,
\]

which shows that \( f_\kappa(v, u) \in \mathcal{C}(v) \). \( \square \)

Let \( \mathcal{P}_{\mathcal{H}}^\kappa = \text{co}\{v_1^\kappa, \ldots, v_{2\kappa}^\kappa\} \subset \mathbb{R}^\kappa \) be a polytope where \( v_i^\kappa \in \mathbb{R}^\kappa, i = 1, \ldots, 2\kappa \). Let \( u_1^\kappa, \ldots, u_{2\kappa}^\kappa \) be the associated controls at the vertices. We assume the following properties:

Properties 7.2.1.

(i) \( v_1^\kappa = (0, 0, \ldots, 0) \in \mathbb{R}^\kappa \)

(ii) \( \{v_1^\kappa, \ldots, v_{\kappa+1}^\kappa\} \) are affinely independent

(iii) For \( i \in \{1, \ldots, \kappa\} \), \( (v_i^\kappa)_1 = 0 \), and \( v_{\kappa+i}^\kappa = -v_i^\kappa + (d, 0, \ldots, 0) \), where \( (d, 0, \ldots, 0) \in \mathbb{R}^\kappa \)

(iv) For \( i \in \{1, \ldots, \kappa\} \), \( u_{\kappa+i}^\kappa = -u_i^\kappa \)

(v) For \( i \in \{1, \ldots, 2\kappa - 1\} \), \( f_\kappa(v_i^\kappa, u_i^\kappa) \to v_{i+1}^{\kappa} \), and \( f_\kappa(v_{2\kappa}^\kappa, u_{2\kappa}^\kappa) \to v_1^\kappa \)
These properties will be used as sufficient conditions for Problem 7.0.1(i). Properties 7.2.1(i) and Properties 7.2.1(iii) ensure that \( y(P_{\mathcal{H}}) \subset [0,d] \). Properties 7.2.1(v) ensures that the invariance conditions hold at each vertex, and therefore \( P_{\mathcal{H}} \) is invariant. Together, these are sufficient for the solvability of Problem 7.0.1(i). We have additional properties for our method of solving Problem 7.0.1. Properties 7.2.1(ii) ensures that \( P_{\mathcal{H}} \) is full dimensional. The transformation in Properties 7.2.1(iii) is motivated by Lemma 7.3.5. We will use this transformation, along with Properties 7.2.1(iv) to focus our design on the motion primitives \( H \) and \( F \). Motion primitive \( B \) will then be obtained by applying this transformation to \( F \).

We now show that such control values for polytope \( P_{\mathcal{H}} \) can be realized by a fixed affine feedback.

**Lemma 7.2.2.** Suppose the polytope \( P_{\mathcal{H}} = \text{co}\{v_1^\kappa, \ldots, v_{2\kappa}^\kappa\} \) and associated control values \( u_1^\kappa, \ldots, u_{2\kappa}^\kappa \) satisfy Properties 7.2.1, then there exists a fixed affine feedback \( u_{\mathcal{H}}^\kappa(x) = F_{\mathcal{H}}^\kappa x + g_{\mathcal{H}}^\kappa \) such that \( u_i^\kappa = u_{\mathcal{H}}^\kappa(v_i^\kappa), i = 1, \ldots, 2\kappa \).

**Proof.** By Properties 7.2.1(ii), \( \{v_1^\kappa, \ldots, v_{2\kappa}^\kappa\} \) are affinely independent. Using the method of [6], one can find unique \( F_{\mathcal{H}}^\kappa \) and \( g_{\mathcal{H}}^\kappa \) corresponding to the affine feedback \( u_{\mathcal{H}}^\kappa(x) = F_{\mathcal{H}}^\kappa x + g_{\mathcal{H}}^\kappa \) on \( \text{co}\{v_1^\kappa, \ldots, v_{2\kappa}^\kappa\} \).

We are left to show that for \( i = \kappa + 2, \ldots, 2\kappa, u_i^\kappa = F_{\mathcal{H}}^\kappa v_i^\kappa + g_{\mathcal{H}}^\kappa \). Using the fact that for \( i = 1, \ldots, \kappa + 1, u_i^\kappa = F_{\mathcal{H}}^\kappa v_i^\kappa + g_{\mathcal{H}}^\kappa \) and Properties 7.2.1(i), (iii), and (iv) we have that for \( i = 1, \ldots, \kappa \)

\[
    u_{\kappa+i}^\kappa = \begin{cases} 
    -u_i^\kappa & \text{if } \kappa+i \leq \kappa + 1 \\
    F_{\mathcal{H}}^\kappa(-v_i^\kappa) - g_{\mathcal{H}}^\kappa & \text{if } \kappa+i = \kappa + 2 \\
    F_{\mathcal{H}}^\kappa(v_{\kappa+i}^\kappa - (d,0,\ldots,0)) - g_{\mathcal{H}}^\kappa & \text{if } \kappa+i = \kappa + 3 \\
    F_{\mathcal{H}}^\kappa v_{\kappa+i}^\kappa - F_{\mathcal{H}}^\kappa v_{\kappa+1}^\kappa - g_{\mathcal{H}}^\kappa & \text{if } \kappa+i = \kappa + 4 \\
    F_{\mathcal{H}}^\kappa v_{\kappa+i}^\kappa + F_{\mathcal{H}}^\kappa v_{1}^\kappa + g_{\mathcal{H}}^\kappa & \text{if } \kappa+i = \kappa + 5 \\
    F_{\mathcal{H}}^\kappa v_{\kappa+i}^\kappa + g_{\mathcal{H}}^\kappa & \text{if } \kappa+i = \kappa + 6 
\end{cases}
\]

which completes the proof. \(\square\)

We give an example which shows these properties in practice.

**Example 7.2.1.** We give an example of a polytope which satisfies Properties 7.2.1 for \( \kappa = 2 \) and we refer to Figure 7.2. Let \( u^* > 0 \). The vertices and their control values are given by \( v_1^1 = (0,0), v_2^1 = (0,u^*), v_3^1 = (d,0), v_4^1 = (0,-u^*), v_1^2 = u^*, v_2^2 = \frac{-(u^*)^2}{d}, v_3^2 = -u^*, \text{ and } v_4^2 = \frac{(u^*)^2}{d} \). Clearly Properties 7.2.1(i)-(iv) hold. By looking at Figure 7.2 it is clear that Property 7.2.1(v) holds, and that the dynamics are directed
in a cycle. The affine feedback $u^2_H(x) = \left[ -2u^* \frac{d}{d} - 2 \frac{(u^*)^2 + du^*}{d} \right] x + u^*$ satisfies the control values at all four vertices.

\[
\text{Hold (} \mathcal{H} \text{)}
\]

Figure 7.2: A polytope of dimension $\kappa = 2$ which satisfies Properties 7.2.1.

We now show that Properties 7.2.1 are sufficient for solving Problem 7.0.1(i).

**Theorem 7.2.3.** A polytope $\mathcal{P}_H^\kappa$ which satisfies Properties 7.2.1 solves Problem 7.0.1(i).

**Proof.** By Properties 7.2.1(ii), $\mathcal{P}_H^\kappa$ is full dimensional. By Properties 7.2.1(v) for $i \in \{1, \ldots, 2n - 1\}$, $f_n(v^n_i, u^n_i) \rightarrow v^n_{i+1}$, and $f_n(v^n_{2n}, u^n_{2n}) \rightarrow v^n_1$. By Lemma 7.2.1, each vertex of $\mathcal{P}_H^\kappa$ satisfies the invariance conditions. Since the invariance conditions hold at the vertices under an affine feedback, then by convexity the invariance conditions hold for all $x \in \mathcal{P}_H^\kappa$ [6].

Since there are no exit facets, for each $x \in \mathcal{P}_H^\kappa$, the invariance conditions $C(x)$ coincide with the Bouligand tangent cone $T_{\mathcal{P}_H^\kappa} (x)$ [89]. By Theorem 2.3.3, $\mathcal{P}_H^\kappa$ is positively invariant, and hence for every $x_0 \in \mathcal{P}_H^\kappa$, $\phi(t, x_0) \in \mathcal{P}_H^\kappa$ for all $t \geq 0$. By Properties 7.2.1(iii), for each vertex $v^n_i \in \{v^n_1, \ldots, v^n_{2n}\}$, $(v^n_i) \in [0, d]$. Therefore, $\mathcal{P}_H^\kappa \subset y^{-1}([0, d])$, and for all $x_0 \in \mathcal{P}_H^\kappa$, $y(t, x_0) \in [0, d]$ for all $t \geq 0$. This solves Problem 7.0.1(i). □

Algorithm 2 is now presented. As stated above, the purpose of the algorithm is to construct a polytope of dimension $\kappa + 1$ which maintains the properties in Properties 7.2.1. In particular, the construction in Algorithm 2 will ensure that the vertices have control values such that the closed-loop vector field is directed at the vertex with following index (i.e. $f_{\kappa+1}(v_1^{\kappa+1}, u_1^{\kappa+1}) \rightarrow v_1^{\kappa+1}$) and the last vector is directed at the first vector, which closes the loop.
Algorithm 2 Algorithm for $\mathcal{H}$ Motion Primitive

1: Let $\mathcal{P}^{\kappa}_{\mathcal{H}} = \text{co}\{v^1_1, \ldots, v^\kappa_{2\kappa}\}$ with associated inputs at the vertices $u^1_1, \ldots, u^\kappa_{2\kappa} \in \mathbb{R}$ satisfy Properties 7.2.1.

2: Let $\tilde{\kappa} = \kappa + 1$

3: Define $v^\tilde{\kappa}_1 = (v^\kappa_1, 0) \in \mathbb{R}^{\tilde{\kappa}}$

4: for $i \in \{2, \ldots, \tilde{\kappa}\}$ do

5: $v_i^\tilde{\kappa} = (u^\kappa_{i-1}, u^\kappa_{i-1}) \in \mathbb{R}^{\tilde{\kappa}}$

6: end for

7: for $i \in \{1, \ldots, \tilde{\kappa}\}$ do

8: $v_{\kappa+i}^\tilde{\kappa} = -v_i^\tilde{\kappa} + (d, 0, \ldots, 0)$, where $(d, 0, \ldots, 0) \in \mathbb{R}^{\tilde{\kappa}}$

9: select $u_i^\tilde{\kappa}$ such that $f_{\kappa}(v_i^\tilde{\kappa}, u_i^\tilde{\kappa}) \rightarrow v_{i+1}^\tilde{\kappa}$

10: $u_{\kappa+i}^\tilde{\kappa} = -u_i^\tilde{\kappa}$

11: end for

12: return $\mathcal{P}^\tilde{\kappa}_{\mathcal{H}} = \text{co}\{v_1^\tilde{\kappa}, \ldots, v_{2\tilde{\kappa}}^\tilde{\kappa}\}$ with associated inputs at the vertices $u_1^\tilde{\kappa}, \ldots, u_{2\tilde{\kappa}}$

We will further analyze Algorithm 2 through Example 7.2.2. Before we do so, we need a polytope which satisfies Properties 7.2.1 in order to initialize Algorithm 2. We construct $\mathcal{P}^1_{\mathcal{H}} = \text{co}\{v_1^1, v_2^1\}$, where $v_1^1 = 0$ and $v_2^1 = d$. We assign controls $u_1^1 = u^*$, $u_2^1 = -u^*$ for some $u^* > 0$.

Lemma 7.2.4. $\mathcal{P}^1_{\mathcal{H}} = \text{co}\{v_1^1, v_2^1\}$ with control values $u_1^1 = u^*$, $u_2^1 = -u^*$ for some $u^* > 0$ satisfies Properties 7.2.1.

Proof. (i) $v_1^1 = 0, v_2^1 = d$. (ii) Clearly $v_1^1 = 0$ and $v_2^1 = d$ are affinely independent. (iii) $v_1^1 = (v_1^1)_1 = 0$, and $v_2^1 = d = -v_1^1 + d$. (iv) $u_1^1 = u^* = -u_2^1$. (v) Since $u^* > 0$ and $d > 0$, $T = d/u^* > 0$ is such that $0 + \tau f_1(0, u^*) = \tau u^* = d$, and hence $f_1(v_1^1, u_1^1) \rightarrow v_2^1$. Similarly $d + \tau f_1(d, -u^*) = d - \tau u^* = 0$, and hence $f_1(v_2^1, v_1^1) \rightarrow v_1^1$. It can be seen that $u_{\mathcal{H}}^1(x) = -\frac{e^{\tau x}}{T} x + u^*$ achieves the desired control values at the vertices.

We now give an example which goes through Algorithm 2 for $\kappa = 1$ and $\kappa = 2$.

Example 7.2.2. We start with $\kappa = 1$ and inputs $\mathcal{P}^1_{\mathcal{H}} = \text{co}\{v_1^1, v_2^1\}$ and $u_1^1, u_2^1$.

2: $\tilde{\kappa} = \kappa + 1 = 2$.

3: $v_2^2 = (v_1^1, 0) = (0, 0)$

4-6: $v_3^2 = (v_1^1, u_1^1) = (0, u^*)$

7-11: $v_4^2 = -v_3^2 + (d, 0) = (d, 0), u_1^2 = u^*, u_3^2 = -u^*, v_5^2 = -v_3^2 + (d, 0) = (d, -u^*), u_2^2 = \frac{-(u^*)^2}{d}$.

The polytope $\mathcal{P}^2_{\mathcal{H}}$ is shown in Figure 7.2.

We now show $\kappa = 2$ with inputs $\mathcal{P}^2_{\mathcal{H}} = \text{co}\{v_1^2, v_2^2, v_3^2, v_4^2\}$ and $u_1^2, u_2^2, u_3^2, u_4^2$.

2: $\tilde{\kappa} = \kappa + 1 = 3$.

3: $v_3^3 = (v_1^2, 0) = (0, 0, 0)$
4-6: \( v_2^3 = (v_1^2, u_1^2) = (0, 0, u^*) \), \( v_3^3 = (v_2^2, u_2^2) = (0, u^* - \frac{(u^*)^2}{d}) \)

7-11: \( v_4^3 = -v_1^3 + (d, 0, 0) = (d, 0, 0) \), \( u_3^3 = u^* \), \( u_4^3 = -u^* \), \( v_3^3 = -v_2^3 + (d, 0, 0) = (d, 0, -u^*) \),
\( u_3^3 = \frac{d(u^*)^2 - (u^*)^3}{dx^2} \), \( u_3^3 = \frac{(u^*)^3 - 3du^*}{dx^2} \), \( u_3^3 = \frac{u^*}{d(u^*)^2} \), \( u_3^3 = \frac{1}{d(u^*)^2} \),
\( u_3^3 = \frac{u^*}{d(u^*)^2} \).

The polytope \( \mathcal{P}_{\mathcal{H}^3} \) is shown in Figure 7.3.

For each vertex \( v^\kappa \) with associated input \( u^\kappa \), the algorithm generates a new vertex \( v^{\kappa+1} = (v^\kappa, u^\kappa) \in \mathbb{R}^{\kappa+1} \). The algorithm also adds two new vertices \( v_{14}^\kappa+1 \) and \( v_{15}^\kappa+1 \). We use the following lemma to determine the control value \( u^{\kappa+1} \) for the newly constructed vertex \( v^{\kappa+1} \).

**Lemma 7.2.5.** Suppose \( v^\kappa, \dot{v}^\kappa \in \mathbb{R}^\kappa \) with associated inputs \( u^\kappa, \dot{u}^\kappa \in \mathbb{R} \) are such that \( f_\kappa(v^\kappa, u^\kappa) \to \dot{v}^\kappa \). Let \( v^{\kappa+1} = (v^\kappa, u^\kappa) \), and \( \dot{v}^{\kappa+1} = (\dot{v}^\kappa, \dot{u}^\kappa) \). There exists \( u^{\kappa+1} \in \mathbb{R} \) such that \( f_{\kappa+1}(v^{\kappa+1}, u^{\kappa+1}) \to \dot{v}^{\kappa+1} \).

**Proof.** Since \( f_\kappa(v^\kappa, u^\kappa) \to \dot{v}^\kappa \), there exists \( \tau > 0 \) such that \( \dot{v}^\kappa = v^\kappa + \tau f_\kappa(v^\kappa, u^\kappa) \). Let \( u^{\kappa+1} = (1/\tau)(\dot{v}^\kappa - u^\kappa) \) and note that by (7.1)

\[
\begin{align*}
    f_{\kappa+1}(v^{\kappa+1}, u^{\kappa+1}) &= ((v^{\kappa+1})_2, \ldots, (v^{\kappa+1})_{\kappa+1}, u^{\kappa+1}) \\
    &= ((v^\kappa)_2, \ldots, (v^\kappa)_\kappa, u^\kappa, u^{\kappa+1}) \\
    &= (f_\kappa(v^\kappa, u^\kappa), u^{\kappa+1})
\end{align*}
\]

Therefore since \( v^\kappa + \tau f_\kappa(v^\kappa, u^\kappa) = \dot{v}^\kappa \), and since \( u^\kappa + \tau u^{\kappa+1} = \dot{u}^\kappa \),

\[
\begin{align*}
    v^{\kappa+1} + \tau f_{\kappa+1}(v^{\kappa+1}, u^{\kappa+1}) &= (v^\kappa, u^\kappa) + \tau(f_\kappa(v^\kappa, u^\kappa), u^{\kappa+1}) \\
    &= (\dot{v}^\kappa, \dot{u}^\kappa) \\
    &= \dot{v}^{\kappa+1},
\end{align*}
\]

Figure 7.3: A polytope of dimension \( \kappa = 3 \) which satisfies Properties 7.2.1.
and hence $f_{k+1}(v^k, u^{k+1}) \rightarrow v^{k+1}$.

We now show that Algorithm 2 is well constructed.

**Theorem 7.2.6.** The output $\mathcal{P}_x^\kappa = \text{co}\{v_1^\kappa, \ldots, v_{2\kappa}^\kappa\}$ with associated inputs at the vertices $u_1^\kappa, \ldots, u_{2\kappa}^\kappa$ of Algorithm 2 satisfies Properties 7.2.1.

**Proof.** We show that $\mathcal{P}_x^\kappa = \text{co}\{v_1^\kappa, \ldots, v_{2\kappa}^\kappa\}$ satisfies Properties 7.2.1.

(i) By construction in line 3 of Algorithm 2, $v_1^\kappa = (v_1^i, 0)$. By Properties 7.2.1(i) $v_1^i = (0, 0, \ldots, 0) \in \mathbb{R}^\kappa$, and hence $v_1^\kappa = (0, 0, \ldots, 0) \in \mathbb{R}^\kappa$.

(ii) Consider the $\tilde{k} + 1$ points $v_1^\kappa, \ldots, v_{\tilde{k}+1}^\kappa$. By construction of $v_1^\kappa, \ldots, v_{\tilde{k}}^\kappa$ in line 5 and by construction of $v_{\tilde{k}+1}^\kappa$ in line 8 we have

$$
\begin{align*}
v_1^\kappa &= (v_1^\kappa, 0) = (0, 0, \ldots, 0), \\
v_i^\kappa &= (v_{i-1}^\kappa, u_{i-1}^\kappa), \quad i = 2, \ldots, \tilde{k}, \\
v_{\tilde{k}+1}^\kappa &= -v_1^\kappa + (d, 0, \ldots, 0) = -(v_1^\kappa, 0) + (d, 0, \ldots, 0) = (v_{\tilde{k}+1}^\kappa, 0).
\end{align*}
$$

We will show that if

$$
\lambda_1 v_1^\kappa + \cdots + \lambda_{\tilde{k}+1} v_{\tilde{k}+1}^\kappa = 0
$$

(7.7)

with $\lambda_i \in \mathbb{R}$ such that $\lambda_1 + \cdots + \lambda_{\tilde{k}+1} = 0$, then $\lambda_1 = \cdots = \lambda_{\tilde{k}+1} = 0$. From (7.6) and (7.7), the first $\kappa$ components give $(\lambda_1 + \lambda_2)v_1^\kappa + \lambda_3 v_2^\kappa + \cdots + \lambda_{\tilde{k}+1} v_{\tilde{k}+1}^\kappa = 0$. By Properties 7.2.1 for $\mathcal{P}_x^\kappa$, $v_1^\kappa, \ldots, v_{\tilde{k}+1}^\kappa$ are affinely independent in $\mathbb{R}^\kappa$, and therefore $(\lambda_1 + \lambda_2) + \lambda_3 + \cdots + \lambda_{\tilde{k}+1} = 0$ implies that $(\lambda_1 + \lambda_2) = \lambda_3 = \cdots = \lambda_{\tilde{k}+1} = 0$. We are left to show that $(\lambda_1 + \lambda_2) = 0$ implies that $\lambda_1 = \lambda_2 = 0$.

Equation (7.7) with $\lambda_3 = \cdots = \lambda_{\tilde{k}+1} = 0$, becomes $\lambda_1 v_1^\kappa + \lambda_2 v_2^\kappa = 0$. Also, by (7.6), $v_1^\kappa = (0, 0, \ldots, 0)$ and $v_2^\kappa = (v_1^\kappa, u_1^\kappa)$. Thus, we have that $\lambda_2 v_2^\kappa = (\lambda_2 v_1^\kappa, \lambda_2 u_1^\kappa) = (0, 0)$. That is, $\lambda_2 u_1^\kappa = 0$. Now we claim that $u_1^\kappa \neq 0$. Suppose $u_1^\kappa = 0$, then $f_\kappa(v_1^\kappa, u_1^\kappa) = f_\kappa((0, 0, \ldots, 0), 0) = (0, \ldots, 0)$. By construction of $u_1^\kappa$ there exists $\tau > 0$ such that $v_2^\kappa = v_1^\kappa + \tau f_\kappa(v_1^\kappa, u_1^\kappa)$. Since $f_\kappa(v_1^\kappa, u_1^\kappa) = (0, \ldots, 0)$, $v_2^\kappa = v_1^\kappa$. Therefore $v_1^\kappa$ and $v_2^\kappa$ are not affinely independent, which contradicts the affine independence of $v_1^\kappa, v_2^\kappa, \ldots, v_{\tilde{k}+1}^\kappa$. It must be that $u_1^\kappa \neq 0$. Therefore since $\lambda_2 u_1^\kappa = 0$, $\lambda_2 = 0$, and $\lambda_1 + \lambda_2 = \lambda_1 = 0$, which shows affine independence.

(iii) By (7.6) and since $\mathcal{P}_x^\kappa$ satisfies Properties 7.2.1(iii), for $i \in \{1, \ldots, \tilde{k}\}$, $(v_i^\kappa)_1 = 0$. By construction of $v_{\tilde{k}+1}^\kappa$ in line 8, $v_{\tilde{k}+1}^\kappa = -v_1^\kappa + (d, 0, \ldots, 0)$ for $i \in \{1, \ldots, \tilde{k}\}$.

(iv) Holds by construction of $u_{\tilde{k}+1}^\kappa, \ldots, u_{2\kappa}^\kappa$, in line 10.

(v) We first show that there exists $u_1^\kappa$ such that $f_\kappa(v_1^\kappa, u_1^\kappa) \rightarrow v_2^\kappa$. That is $\exists u_1^\kappa, \tau > 0$ such that $v_2^\kappa = v_1^\kappa + \tau f_\kappa(v_1^\kappa, u_1^\kappa)$. By (7.6), $v_1^\kappa = (0, 0, \ldots, 0)$ and $v_2^\kappa = (v_1^\kappa, u_1^\kappa) = (0, \ldots, 0, u_1^\kappa)$. Setting $u_1^\kappa = u_1^\kappa$
which completes the proof for (v).

Since $\mathcal{P}_{\mathcal{F}}$ satisfies Properties 7.2.1(v), by Lemma 7.2.5 for $i \in \{2, \ldots, \hat{k}\}$ there exist $\hat{u}_{i}^\kappa$ such that $f_{\kappa}(v_{i}^\kappa, u_{i}^\kappa) \rightarrow v_{i+1}^\kappa$. Therefore for $i \in \{2, \ldots, \hat{k}\}$, there exists $\tau > 0$ such that $v_{i}^\kappa + \tau f_{\kappa}(v_{i}^\kappa, u_{i}^\kappa) = v_{i+1}^\kappa$.

By (iii), for $i \in \{1, \ldots, \hat{k}\}$, $u_{i}^\kappa = -v_{k+1}^\kappa + (d, 0, \ldots, 0)$, and $u_{i}^\kappa = -u_{k+1}^\kappa$, and so by the dynamics of (7.1),

\[
 f_{\kappa}(v_{i}^\kappa, u_{i}^\kappa) = f_{\kappa}(-v_{k+1}^\kappa + (d, 0, \ldots, 0), -u_{k+1}^\kappa) = -f_{\kappa}(v_{k+1}^\kappa, u_{k+1}^\kappa). 
\]

This implies that for $i \in \{1, \ldots, \hat{k} - 1\}$,

\[
 v_{k+1}^\kappa + \tau f_{\kappa}(v_{k+1}^\kappa, u_{k+1}^\kappa) = -v_{i}^\kappa + (d, 0, \ldots, 0) - \tau f_{\kappa}(v_{i}^\kappa, u_{i}^\kappa) \\
 = -v_{i+1}^\kappa + (d, 0, \ldots, 0) \\
 = v_{i+1}^\kappa 
\]

Therefore we have shown that for $i \in \{1, \ldots, 2\hat{k} - 1\}$, $f_{\kappa}(v_{i}^\kappa, u_{i}^\kappa) \rightarrow v_{i+1}^\kappa$. It remains to show that $f_{\kappa}(v_{2\hat{k}}^\kappa, u_{2\hat{k}}^\kappa) \rightarrow v_{1}^\kappa$.

Let $\tau > 0$ be such that $v_{2\hat{k}}^\kappa + \tau f_{\kappa}(v_{2\hat{k}}^\kappa, u_{2\hat{k}}^\kappa) = v_{1}^\kappa$. Then we have as above

\[
 v_{2\hat{k}}^\kappa + \tau f_{\kappa}(v_{2\hat{k}}^\kappa, u_{2\hat{k}}^\kappa) = -v_{2\hat{k}}^\kappa + (d, 0, \ldots, 0) - \tau f_{\kappa}(v_{2\hat{k}}^\kappa, u_{2\hat{k}}^\kappa) \\
 = -v_{1}^\kappa + (d, 0, \ldots, 0) \\
 = v_{1}^\kappa 
\]

which completes the proof for (v).

Finally we show that the above control values can be realized by a single affine feedback. \hfill \Box

## 7.3 $\mathcal{F}$ and $\mathcal{B}$ Motion Primitives

We now construct a polytope which solves Problem 7.0.1(ii) and (iii). The process will be almost identical to the construction of $\mathcal{P}_{\mathcal{F}}$ above with two main differences. The first difference is that the polytope $\mathcal{P}_{\mathcal{F}}$ that we construct will have an exit facet on the plane $x_1 = d$. The second difference is that the closed-loop vectors at the vertices will be directed at each other in a chain instead of a cycle. That is the closed-loop vector field at vertex $v_{2\kappa}^\kappa$ will be directed at a fixed point outside of $\mathcal{P}_{\mathcal{F}}$ instead of being directed at $v_{1}^\kappa$.

Let $\mathcal{P}_{\mathcal{F}} = \text{co}\{v_{1}^\kappa, \ldots, v_{2\kappa}^\kappa\} \subset \mathbb{R}^n$ be a polytope where $v_{i}^\kappa \in \mathbb{R}^n$, $i = 1, \ldots, 2\kappa$. Let $\mathcal{F}_{0}^\kappa = \text{co}\{v_{\kappa+1}^\kappa, \ldots, v_{2\kappa}^\kappa\}$ be the exit facet. Let $u_{1}^\kappa, \ldots, u_{2\kappa}^\kappa$ be the associated controls at the vertices. We assume the following properties:

**Properties 7.3.1.**
(i) \( v_1^\kappa = (0,0,\ldots,0) \in \mathbb{R}^\kappa \)

(ii) \( \{v_1^\kappa,\ldots,v_{\kappa+1}^\kappa\} \) are affinely independent

(iii) For \( i \in \{1,\ldots,\kappa\} \), \((v_i^\kappa)_j = 0\) and \( v_{\kappa+i}^\kappa = v_i^\kappa + (d,0,\ldots,0) \), where \((d,0,\ldots,0) \in \mathbb{R}^\kappa \)

(iv) For \( i \in \{1,\ldots,\kappa\} \), \( u_{\kappa+i}^\kappa = u_i^\kappa \)

(v) For \( i \in \{1,\ldots,2\kappa - 1\} \), \( f_\kappa(v_i^\kappa,u_i^\kappa) \rightarrow v_{i+1}^\kappa \) and \( f_\kappa(v_{2\kappa}^\kappa,u_{2\kappa}^\kappa) \rightarrow (2d,0,\ldots,0) \)

(vi) The polytope \( \mathcal{P}_\mathcal{H}^\kappa \) is simple, that is each vertex lies on \( \kappa \) edges (and \( \kappa \) facets)

These properties will be used as sufficient conditions for Problem 7.0.1(ii). Properties 7.3.1(i) and Properties 7.3.1(iii) ensure that \( y(\mathcal{P}_\mathcal{H}^\kappa) \subset [0,d] \). Properties 7.3.1(v) and Properties 7.3.1(vi) ensure that the invariance conditions hold at the vertices. Therefore if trajectories leave \( \mathcal{P}_\mathcal{H}^\kappa \), they do so via the exit facet \( \mathcal{F}_0^\kappa \). The transformations in Properties 7.3.1(iii) and Properties 7.3.1(iv) are different from those in Properties 7.2.1(iii) and Properties 7.2.1(iv). They will ensure that all trajectories leave \( \mathcal{P}_\mathcal{H}^\kappa \). This along with the invariance conditions are sufficient for solving Problem 7.0.1(ii).

We now show that such control values for polytope \( \mathcal{P}_\mathcal{H}^\kappa \) can be realized by a fixed affine feedback.

**Lemma 7.3.1.** Suppose the polytope \( \mathcal{P}_\mathcal{H}^\kappa = \text{co}\{v_1^\kappa,\ldots,v_{2n}^\kappa\} \) and associated control values \( u_1^\kappa,\ldots,u_{2n}^\kappa \) satisfy Properties 7.3.1, then there exists a fixed affine feedback \( u^\kappa(x) = F_\mathcal{H}^\kappa x + g_\mathcal{H}^\kappa \) such that \( u_i^\kappa = u^\kappa(v_i^\kappa) \), \( i = 1,\ldots,2\kappa \).

**Proof.** The proof follows the same logic as the proof for Lemma 7.2.2. \( \square \)

**Theorem 7.3.2.** A polytope \( \mathcal{P}_\mathcal{H}^\kappa \) which satisfies Properties 7.3.1 solves Problem 7.0.1(ii).

**Proof.** We begin by showing that the invariance conditions hold. By Properties 7.3.1(ii), \( \mathcal{P}_\mathcal{H}^\kappa \) is full dimensional. By Properties 7.2.1(v) for \( i \in \{1,\ldots,2n-1\} \), \( f_n(v_i^n,u_i^n) \rightarrow v_{i+1}^n \). By Lemma 7.2.1, each of these vertices satisfy the invariance conditions. We have left to show that the invariance conditions hold at \( v_{2n}^n \). We do this by showing that \( v_{2n}^n \) satisfies the invariance conditions of the \( n - 1 \) facets it shares with \( v_1^n \), and that it lies on \( \mathcal{F}_0^n \) (which does not impose any invariance conditions). Since \( \mathcal{P}_\mathcal{H}^\kappa \) is simple by Properties 7.2.1(vii), \( v_{2n}^n \) only lies on \( n \) facets, and hence this will show the invariance conditions hold at \( v_{2n}^n \).

Suppose \( n = 1 \), then \( v_{2n}^n \in \mathcal{F}^1_1 \). Since \( \mathcal{P}_\mathcal{H}^\kappa \) is simple, \( v_{2n}^n \) lies on exactly one facet, \( \mathcal{F}^1_0 \), and hence the invariance conditions hold.

Suppose \( n > 1 \). For \( i \in \{1,\ldots,n\} \), let \( v_i^{n-1} = ((v_i^n)_2,\ldots,(v_i^n)_n) \in \mathbb{R}^{n-1} \) and \( S^{n-1} = \text{co}\{v_1^{n-1},\ldots,v_n^{n-1}\} \). It can be seen that \( v_1^{n-1},\ldots,v_n^{n-1} \) are affinely independent since \( v_1^n,\ldots,v_n^n \) are affinely independent by Properties 7.3.1(ii). This shows that \( S^{n-1} \) is a simplex in \( \mathbb{R}^{n-1} \). We create a projection
of polytopes which projects $P_\mathfrak{F}$ onto the simplex $S^{n-1}$. The projection simply omits the first coordinate. Let $\text{Proj}: P_\mathfrak{F} \rightarrow S^{n-1}$, where $\text{Proj}(v^n) = ((v^n)_2, \ldots, (v^n)_n)$. Clearly $\text{Proj}(P_\mathfrak{F}) = S^{n-1}$, and therefore $\text{Proj}$ is a projection of polytopes. By Lemma 5.2.2, for every face $F^{n-1}$ of $S^{n-1}$, $F^n := \text{Proj}^{-1}(F^{n-1})$ is a face of $P_\mathfrak{F}$. We show that if $F^{n-1}$ is a facet, then $F^n$ is also a facet. Let $F^{n-1}$ be a facet, then there exists $w_1^{n-1}, \ldots, w_{n-1}^{n-1} \in F^{n-1}$ which are affinely independent. Therefore $(0, w_1^{n-1}), \ldots, (0, w_{n-1}^{n-1}), (d, w_{n-1}^{n-1}) \in F^n$ are affinely independent, which shows that $F^n$ is facet.

Since $S^{n-1}$ is a simplex, $v_1^n$ lies on $n-1$ facets $F_1^{n-1}, \ldots, F_{n-1}^{n-1}$. Since $\text{Proj}(v_1^n) = \text{Proj}(v_2^n) = v_1^n$, $v_1^n$ and $v_2^n$ both lie on the $n-1$ facets $\text{Proj}^{-1}(F_1^{n-1}), \ldots, \text{Proj}^{-1}(F_{n-1}^{n-1})$. We will show that $v_2^n$ satisfies the invariance conditions for these shared facets. Let $h_j$ be the outward normal of a shared facet $\text{Proj}^{-1}(F_j^{n-1})$, then by construction of $v_2^n$, and the fact that $v_1^n$ satisfies the invariance conditions for $\text{Proj}^{-1}(F_j^{n-1})$

$$h_j : f_n(v_2^n, u_2^n) = h_j : f_n(v_1^n + (d, 0, \ldots, 0), u_2^n)$$

$$= h_j : f_n(v_1^n, u_1^n)$$

$$\leq 0.$$

Clearly $v_1^n \notin F_0^n$ and $v_2^n \in F_0^n$, which shows that $F_0^n$ is not a shared facet of $v_1^n$ and $v_2^n$. Since $P_\mathfrak{F}$ is simple, $v_2^n$ lies on $n$ facets: the $n - 1$ for which we have shown the invariance conditions hold, and $F_0^n$. Therefore the invariance conditions hold for $v_2^n$. Since the invariance conditions hold at the vertices by affine feedback, the invariance conditions hold for all $x \in P_\mathfrak{F}^n$ [6].

If we can now show that trajectories leave $P_\mathfrak{F}^n$, they must leave through $F_0^n$, solving Problem 7.0.1(ii). Suppose $n = 1$. Since $f_1(v_1^1, u_1^1) \rightarrow v_2^1$, and since $v_1^1 = 0$ and $v_2^1 = d$, $u_1^1 = u_2^1 = u^*$ for some $u^* > 0$ and $u_2^1(x) = u^*$. Let $x_0 \in P_1^1$. The solution of (7.1) is $\phi(t, x_0) = x_0 + \int_0^t u^*d\tau = x_0 + u^*t$. Since $u^* > 0$, Problem 7.0.1(ii) is solved.

Suppose $n > 1$. By a similar argument as for the $n = 1$ case above it can be shown that $P_\mathfrak{F}^n \subset \{x \in \mathbb{R}^n \mid x_2 \geq 0\}$. Suppose by way of contradiction that there exists $x_0 \in \mathbb{R}^n$ such that $\phi(t, x_0) \in P_\mathfrak{F}^n$ for all $t \geq 0$. Since $P_\mathfrak{F}^n \subset \{x \in \mathbb{R}^n \mid x_2 \geq 0\}$, it must be that $(\phi(t, x_0))_2 \rightarrow 0$ as $t \rightarrow \infty$, since otherwise $(\phi(t, x_0))_1 \rightarrow \infty$ as $t \rightarrow \infty$ (and hence leave $P_\mathfrak{F}^n$) by the dynamics of (7.1). Since $\phi(t, x_0) \in P_\mathfrak{F}^n$ for all $t \geq 0$ and since $u_2^n(x)$ is an affine feedback, $\phi(t, x_0)_2$ exists and is bounded, and hence by Barbalat’s Lemma[83], if $(\phi(t, x_0))_2 \rightarrow 0$ as $t \rightarrow \infty$, $\phi(t, x_0)_2 \rightarrow 0$ as $t \rightarrow \infty$. Continuing this argument for each coordinate we have that $\phi(t, x_0) \rightarrow 0$ as $t \rightarrow \infty$. By the dynamics of (7.1), for all $i \in \{2, \ldots, n\}$, $(\phi(t, x_0))_i \rightarrow 0$ as $t \rightarrow \infty$. Since $P_\mathfrak{F}^n$ is closed $\phi(t, x_0) \rightarrow \{x \in P_\mathfrak{F}^n \mid x_2 = \cdots = x_n = 0\}$ as $t \rightarrow \infty$. This is the set $O_{\mathfrak{F}^n} = \text{co}\{v_1^n, v_{n+1}^n\}$. By a similar argument shown for the $n = 1$ case, for all $x \in O_{\mathfrak{F}^n}$,
and hence \( f_0 + (i) \). Proof. Properties 7.3.1.

Algorithm 3 is now presented. The purpose of the algorithm is to construct a polytope of dimension \( \kappa + 1 \), which maintains the properties in Properties 7.3.1. In particular, the construction in Algorithm 3 will maintain that the vertices have control values such that the closed-loop vector field is directed at vertex with the next index (i.e. \( f_{\kappa+1}(v_{i}^{\kappa+1}, u_{i}^{\kappa+1}) \rightarrow v_{i+1}^{\kappa+1}) \).

Algorithm 3 Algorithm for \( \mathcal{F} \) Motion Primitive

1: Let \( \mathcal{P}_{\mathcal{F}}^{\kappa} = \text{co}\{v_1^{\kappa}, \ldots, v_{2\kappa}^{\kappa}\} \) with associated inputs at the vertices \( u_1^{\kappa}, \ldots, u_{2\kappa}^{\kappa} \in \mathbb{R} \) satisfy Properties 7.3.1
2: Let \( \tilde{\kappa} = \kappa + 1 \)
3: Define \( v_1^{\tilde{\kappa}} = (v_1^{\kappa}, 0) \in \mathbb{R}^{\tilde{\kappa}} \)
4: for \( i \in \{2, \ldots, \tilde{\kappa}\} \) do
5: \( v_i^{\tilde{\kappa}} = (v_{i-1}^{\kappa}, v_{i-1}^{\kappa}) \in \mathbb{R}^{\tilde{\kappa}} \)
6: end for
7: for \( i \in \{1, \ldots, \tilde{\kappa}\} \) do
8: \( v_{i+1}^{\tilde{\kappa}} = v_i^{\tilde{\kappa}} + (d, 0, \ldots, 0) \), where \( (d, 0, \ldots, 0) \in \mathbb{R}^{\tilde{\kappa}} \)
9: select \( u_i^{\kappa} \) such that \( f_{\tilde{\kappa}}(v_i^{\kappa}, u_i^{\kappa}) \rightarrow v_{i+1}^{\tilde{\kappa}} \)
10: \( u_{i+1}^{\kappa} = u_i^{\kappa} \)
11: end for
12: return \( \mathcal{P}_{\mathcal{F}}^{\kappa} = \text{co}\{v_1^{\kappa}, \ldots, v_{2\kappa}^{\kappa}\} \) with associated inputs at the vertices \( u_1^{\kappa}, \ldots, u_{2\kappa}^{\kappa} \)

In order to initialize Algorithm 3, we need to have a polytope which satisfies Properties 7.3.1. We construct \( \mathcal{P}_{\mathcal{F}}^{1} = \text{co}\{v_1^{1}, v_2^{1}\} \), where \( v_1^{1} = 0 \) and \( v_2^{1} = d \). We assign controls \( u_1^{1} = u^{*}, u_2^{1} = u^{*} \) for some \( u^{*} > 0 \).

**Lemma 7.3.3.** \( \mathcal{P}_{\mathcal{F}}^{1} = \text{co}\{v_1^{1}, v_2^{1}\} \) with associated controls \( u_1^{1} = u^{*}, u_2^{1} = u^{*} \) for some \( u^{*} > 0 \), satisfies Properties 7.3.1.

**Proof.** (i) \( v_1^{1} = 0, v_2^{1} = d \). (ii) Clearly \( v_1^{1} = 0 \) and \( v_2^{1} = d \) are affinely independent. (iii) \( v_2^{1} = (v_1^{1})_1 = 0, \) and \( v_2^{1} = d = v_1^{1} + d \). (iv) \( u_1^{1} = u^{*} = u_2^{1} \). (v) Since \( u^{*} > 0 \) and \( d > 0 \), \( T = d/u^{*} > 0 \) is such that \( 0 + \tau f_1(0, u^{*}) = Tu^{*} = d \), and hence \( f_1(v_1^{1}, u_1^{1}) \rightarrow v_2^{1} \). Similarly \( d + T f_1(d, -u^{*}) = d + Tu^{*} = 2d \), and hence \( f_1(v_2^{1}, v_2^{1}) \rightarrow 2d \). (vi) \( \mathcal{F}_{0}^{1} = v_2^{1} \), which is a facet of \( \mathcal{P}_{\mathcal{F}}^{1} \). (vii) Since \( \mathcal{P}_{\mathcal{F}}^{1} \) is a simplex, it is simple. □

We now show that Algorithm 2 is well constructed.
Theorem 7.3.4. The output $\mathcal{P}_f^\xi = \text{co}\{v^\xi_1, \ldots, v^\xi_{2k}\}$ with associated inputs at the vertices $u^\xi_1, \ldots, u^\xi_{2k}$ of Algorithm 3 satisfies Properties 7.3.1.

Proof. We show that $\mathcal{P}_f^\xi = \text{co}\{v^\xi_1, \ldots, v^\xi_{2k}\}$ satisfies Properties 7.3.1.

(i) By construction in line 3 of Algorithm 3, $v^\xi_1 = 0$. By Properties 7.3.1(i) $v^\xi_1 = (0, \ldots, 0) \in \mathbb{R}^\kappa$, and hence $v^\xi_1 = (0, \ldots, 0) \in \mathbb{R}^\kappa$.

(ii) Since the construction of $v^\xi_1, \ldots, v^\xi_{2k}$ in both Algorithm 2 and Algorithm 3 is identical, the proof is the same as that in Theorem 7.2.6(ii).

(iii) By construction of $v^\xi_1, \ldots, v^\xi_{2k}$ in lines 3 and 5, and since $\mathcal{P}_f^\xi$ satisfies Properties 7.3.1(iii), for $i \in \{1, \ldots, \hat{k}\}$, $(v^\xi_i)_1 = 0$. By construction in line 8, for $i \in \{1, \ldots, \hat{k}\}$, $v^\xi_{i+1} = v^\xi_i + (d, 0, \ldots, 0)$.

(iv) Holds by construction of $u^\xi_1, \ldots, u^\xi_{2k}$, in line 10.

(v) We first show that there exists $u^\xi_1$ such that $f_\kappa(v^\xi_1, u^\xi_1) \rightarrow v^\xi_2$. That is $\exists u^\xi_1, \tau > 0$ such that $v^\xi_2 = v^\xi_1 + f_\kappa(v^\xi_1, u^\xi_1)$. By Properties 7.3.1(i), $v^\xi_1 = (0, \ldots, 0)$. By construction of $v^\xi_2$ in line 5, $v^\xi_2 = (v^\xi_1, u^\xi_1) = (0, \ldots, 0, u^\xi_1)$. Setting $u^\xi_1 = u^\xi_1$ gives $v^\xi_1 + f_\kappa(v^\xi_1, u^\xi_1) = (0, \ldots, 0) + (0, \ldots, 0, u^\xi_1) = v^\xi_2$, and hence $f_\kappa(v^\xi_1, u^\xi_1) \rightarrow v^\xi_2$.

Since $\mathcal{P}_f^\xi$ satisfies Properties 7.3.1, by Lemma 7.2.5 for $i \in \{2, \ldots, \hat{k}\}$ there exist $u^\xi_i$ such that $f_\kappa(v^\xi_i, u^\xi_i) \rightarrow v^\xi_{i+1}$. Therefore for $i \in \{2, \ldots, \hat{k}\}$, there exists $\tau > 0$ such that $v^\xi_i + \tau f_\kappa(v^\xi_i, u^\xi_i) = v^\xi_{i+1}$.

By (iii), for $i \in \{1, \ldots, \hat{k}\}$, $v^\xi_i = v^\xi_{i+i} + (d, 0, \ldots, 0)$, and $u^\xi_i = u^\xi_{i+i}$, and so by the dynamics of (7.1), $f_\kappa(v^\xi_i, u^\xi_i) = f_\kappa(v^\xi_{i+i} + (d, 0, \ldots, 0), u^\xi_{i+i}) = f_\kappa(v^\xi_{i+i}, u^\xi_{i+i})$. This implies that for $i \in \{1, \ldots, \hat{k} - 1\}$,

$$v^\xi_{i+i} + \tau f_\kappa(v^\xi_{i+i}, u^\xi_{i+i}) = v^\xi_i + (d, 0, \ldots, 0) + \tau f_\kappa(v^\xi_i, u^\xi_i) = v^\xi_{i+1} + (d, 0, \ldots, 0) = v^\xi_{i+i+1}$$

Therefore we have shown that for $i \in \{1, \ldots, 2\hat{k} - 1\}$, $f_\kappa(v^\xi_i, u^\xi_i) \rightarrow v^\xi_{i+1}$. It remains to show that $f_\kappa(v^\xi_{2k}, u^\xi_{2k}) \rightarrow (2d, 0, \ldots, 0)$.

Let $\tau > 0$ be such that $v^\xi_{i+i} + \tau f_\kappa(v^\xi_{i+i}, u^\xi_{i+i}) = v^\xi_{i+1}$. Then we have as above

$$v^\xi_{2k} + \tau f_\kappa(v^\xi_{2k}, u^\xi_{2k}) = v^\xi_{2k} + (d, 0, \ldots, 0) + \tau f_\kappa(v^\xi_{2k}, u^\xi_{2k}) = v^\xi_{k+1} + (d, 0, \ldots, 0) = (2d, 0, \ldots, 0)$$
which completes the proof for (v).

(vi) For \( i \in \{1, \ldots, \kappa \} \), let \( v_i^{\kappa - 1} = ((v_i^\kappa)_2, \ldots, (v_i^\kappa)_\kappa) \in \mathbb{R}^{\kappa - 1} \) and \( S = \co\{v_1^{\kappa - 1}, \ldots, v_{\kappa}^{\kappa - 1}\} \). Since \( v_1^\kappa, \ldots, v_\kappa^\kappa \) are affinely independent by (ii), \( v_1^{\kappa - 1}, \ldots, v_{\kappa}^{\kappa - 1} \) are affinely independent. This shows that \( S \) is a simplex in \( \mathbb{R}^{\kappa - 1} \) and therefore simple. Note that \( \mathcal{P}_\phi^{\kappa} = [0, d] \times S \) and hence \( \mathcal{P}_\phi^{\kappa} \) is created by extending \( S \) along the \( x_1 \) axis. This shows that each vertex \( v_i^{\kappa} \in \mathcal{P}_\phi^{\kappa} \subset \mathbb{R}^\kappa \) lies on one more edge than a vertex \( v_i^{\kappa - 1} \in S \subset \mathbb{R}^{\kappa - 1} \) and hence, \( \mathcal{P}_\phi^{\kappa} \) simple.

We conclude by showing that \( \mathcal{F}_0^{\kappa} \) is a facet. By (iii), for each \( i \in \{1, \ldots, \kappa \} \), \( (v_i^\kappa)_1 = 0 \) and for each \( i \in \{\kappa + 1, \ldots, 2\kappa\} \), \( (v_i^\kappa)_1 = d \). Therefore for all \( x \in \text{int} (\mathcal{P}_\phi^{\kappa}) \), \( x_1 \in (0, d) \). For each \( i \in \{\kappa + 1, \ldots, 2\kappa\} \), \( v_i^\kappa \) lies on the plane \( x_1 = d \), and hence \( \mathcal{F}_0^{\kappa} := \co\{v_{\kappa + 1}^\kappa, \ldots, v_{2\kappa}^\kappa\} \) is a face. Also, since \( v_1^\kappa, \ldots, v_\kappa^\kappa \) are affinely independent, and since for \( i \in \{1, \ldots, \kappa\} \), \( v_i^{\kappa + i} = v_i^\kappa + (d, 0, \ldots, 0) \), \( v_{\kappa + 1}^\kappa, \ldots, v_{2\kappa}^\kappa \) are affinely independent. Therefore \( \mathcal{F}_0^{\kappa} \) has dimension \( \kappa - 1 \) and is a facet. \( \square \)

Before we present Theorem 7.3.6 for solving Problem 7.0.1(iii), we require the following result relating the solvability of Problem 7.0.1(ii) and (iii). It relies on a transformation \( \tilde{x} = -x + (d, 0, \ldots, 0) \in \mathbb{R}^n \) which will also be used in the construction of the polytope \( \mathcal{P}_\phi^{\kappa} \).

**Lemma 7.3.5.** Let \( \mathcal{P} \subset \mathbb{R}^n \) be compact and \( u(x) \) be such that \( f_u(x, u(x)) \) is locally Lipschitz. Let \( x_0 \in \mathcal{P} \) and \( \tilde{x}_0 = -x_0 + (d, 0, \ldots, 0) \). Consider the trajectory \( \phi(t, x_0) \) of (7.1) with input \( u(t, x) \), and the trajectory \( \tilde{\phi}(t, \tilde{x}_0) \) of \( \tilde{x} = f_u(\tilde{x}, u) \) with input \( u(\tilde{\phi}(t, \tilde{x}_0) + (d, 0, \ldots, 0)) \). Then for all \( t \) such that \( \phi(t, x_0) \in \mathcal{P} \)

\[
\phi(t, x) = -\tilde{\phi}(t, \tilde{x}_0) + (d, 0, \ldots, 0).
\] (7.8)

**Proof.** \( f_u(x, u(x)) \) is locally Lipschitz and therefore for all \( t \) such that \( \phi(t, x_0) \in \mathcal{P} \) there exists a unique solution to the initial value problem

\[
\frac{d}{dt} \phi(t, x_0) = f_u(\phi(t, x_0), u(\phi(t, x_0)))
\]

\[
\phi(0, x_0) = x_0.
\]

By the dynamics of (7.1), for all \( t \) such that \( \phi(t, x_0) \in \mathcal{P} \)

\[
\frac{d}{dt} \left(-\tilde{\phi}(t, \tilde{x}_0) + (d, 0, \ldots, 0)\right) = -f_u(\tilde{\phi}(t, \tilde{x}_0), u(\phi(t, x_0)))
\]

\[
= f_u(-\tilde{\phi}(t, \tilde{x}_0) + (d, 0, \ldots, 0), u(\phi(t, x_0))),
\]

with initial condition given by \( \tilde{\phi}(0, \tilde{x}_0) + (d, 0, \ldots, 0) = x_0 \). By uniqueness of solutions, for all \( t \) such that \( \phi(t, x_0) \in \mathcal{P} \), \( \phi(t, x_0) = -\tilde{\phi}(t, \tilde{x}_0) + (d, 0, \ldots, 0) \). Note that since \( \tilde{\phi}(t, x) = -\phi(t, x_0) + (d, 0, \ldots, 0) \),
We now show that if we solve Problem 7.0.1(ii), we can also solve Problem 7.0.1(iii).

**Theorem 7.3.6.** Suppose \( P_{\mathcal{F}}^n \) and \( u_{\mathcal{F}}^n(x) \) solve Problem 7.0.1(ii) and \( f_n(x,u_{\mathcal{F}}(x)) \) is locally Lipschitz. Then \( P_{\mathcal{F}}^n := \{ -\dot{x} + (d,0,\ldots,0) \mid \dot{x} \in P_{\mathcal{F}}^n \} \) and \( u_{\mathcal{F}}^n(x) := u_{\mathcal{F}}^n(-x+(d,0,\ldots,0)) \) solve Problem 7.0.1(iii).

**Proof.** Let \( x_0 \in P_{\mathcal{F}}^n \), then \( x_0 = -\dot{x}_0 + (d,0,\ldots,0) \) for some \( \dot{x}_0 \in \mathcal{P}_{\mathcal{F}}^n \). Let \( u_{\mathcal{F}}^n(x) \) be the input of (7.1) with initial condition \( x_0 \). Let \( u_{\mathcal{F}}^n(\dot{x}) \) be the input of \( \dot{x} = f_n(\dot{x},u) \) with state trajectory \( \dot{\phi}(t,\tilde{x}_0) \). Since \( u_{\mathcal{F}}^n(\dot{x}) = u_{\mathcal{F}}^n(\dot{x}) \), by Lemma 7.3.5, for all \( t \) such that \( \tilde{\phi}(t,\tilde{x}_0) \in P_{\mathcal{F}}^n \), \( \phi(t,x_0) = -\tilde{\phi}(t,\tilde{x}_0) + (d,0,\ldots,0) \). Therefore \( \tilde{y}(t,\tilde{x}_0) = \tilde{\phi}(t,\tilde{x}_0) = (-\phi(t,x_0))_1 + d = -y(t,x_0) + d \). Note that that \( \{-y+d \mid y \in Y^*\} =: -Y^* + d = Y^* \).

By assumption \( P_{\mathcal{F}}^n \) and \( u_{\mathcal{F}}(\dot{x}) \) are such that for every \( \dot{x}_0 \in \mathcal{P}_{\mathcal{F}}^n \) there exists \( T \geq 0 \) and \( \gamma > 0 \) such that \( \tilde{y}(t,\tilde{x}_0) \in Y^* \) for all \( t \in [0,T] \), \( \tilde{y}(T,\tilde{x}_0) = d \), and \( \tilde{y}(t,\tilde{x}_0) \not\in Y^* \) for all \( t \in (T,T+\gamma) \). Therefore \( y(t,x_0) \in -Y^* + d = Y^* \) for all \( t \in [0,T] \), \( y(T,x_0) = -d + d = 0 \), and \( y(t,x_0) \not\in -Y^* + d = Y^* \) for all \( t \in (T,T+\gamma) \).

To validate the results above, each of the algorithms were implemented and tested in Matlab to verify that the polytopes indeed achieve their objective. The plots below show the performance for integrator chains of dimension \( n = 1,\ldots,12 \) for both the Hold and Forward motion primitives. The plots show the output \( y = x_1 \) over time. For each of these examples, the initial condition was the origin and \( d = 10 \). Therefore, for the Hold motion primitive the output should remain within \([0,10]\) for all time, and for the Forward mode the output should leave \([0,10]\) through the face \( x_1 = 10 \) at some finite time. It can be seen that integrator chains of higher order take longer to stabilize for the Hold motion primitive and take longer to leave the Forward polytope, as would be expected.

In order to integrate the development of this chapter into Chapter 6, one would need to show that the \( n \)-th order integrator and associated motion primitives satisfy Assumption 6.3.1. Since Assumption 6.3.1(i)-(vi) are concerned with the discrete nature of the MA, these assumptions hold. Assumption 6.3.1(vii) is concerned with the flow condition of the RCP. This difficult theoretical result is beyond the scope of this thesis. The flow condition has been shown to hold for every initial condition and value of \( n \) that we have run in simulation.
Chapter 8

Conclusion

This dissertation began by motivating the study of problems with complex control specifications. We chose one method for addressing such specifications for affine systems, namely the RCP. While significant advancements in the RCP have been made, we observed a gap in the theory. That gap was an extension of the RCP to the output space, the ORCP. We addressed this gap by formulating and solving the ORCP using two different methods, as well as using this new theory towards a framework for motion planning of robotic systems. We now highlight the main contributions of this dissertation.

The first contribution of this dissertation extended regulatory theory to affine systems. Using this extension, a disturbance rejection problem was formulated for the RCP. The approach was to encode the desired RCP behaviour on each simplex in an exosystem. While this method for solving the ORCP has good practical value, the formulation does not guarantee that the transient behaviour will remain within the simplex, and could break the safety constraints of the RCP.

In light of the drawbacks of the first contribution, a second problem formulation was constructed which would guarantee the safety constraints imposed on the output space. This second contribution was the formulation of the ORCP on a simplex, which removed the disturbance, and relied on existing RCP techniques on polytopes in the state space. The solution took the simplex in the output space and lifted it to a polytope in the full state space. A modified viability algorithm was used to ensure that the invariance conditions on this full dimensional polytope were solvable. It was then shown that solving the RCP on the full dimensional polytope solves the ORCP on the simplex in the output space.

With a solid foundation for solving the ORCP established, we wanted to apply this theory towards real world applications. This lead to the third contribution, a modular framework for motion planning of robotic systems. We developed a modular, hierarchical framework for motion planning of heterogenous
agents in known environments. It consists of several modules. An output transition system (OTS) models the allowable motions of the agents by partitioning their workspace into boxes. A set of motion primitives is designed based on the RCP on polytopes. A maneuver automaton (MA) captures constraints on successive motion primitives. Finally, a control policy is generated based on the synchronous product of the OTS and the discrete part of the MA. Overall we obtain a two-level control design which is highly robust, modular, and conceptually elegant. We presented a specific maneuver automaton for single, double and triple integrator systems, and illustrated its effectiveness to coordinate motion between quadrocopters as well as robotic manipulators.

The final contribution was a generalization of the motion primitives constructed in the motion planning framework to \( n \)-th order integrator systems. This generalization complements the aforementioned motion planning framework, and allows the control of more complex integrator systems such as robotic manipulators with flexible materials which are known to be feedback linearizable to quadruple integrators.

We now discuss some future directions to extend the work in this dissertation. The first future direction would be to attempt solving the ORCP in a different manner. Since the dynamics of the output in general rely on the full state dynamics, an equivalent formulation of the ORCP on a simplex is a formulation of the RCP on a polyhedra in the full state space. Although defining equivalent invariance and flow conditions for polyhedra would be one method for solving the ORCP, solving the RCP on polytopes has proven to be challenging, and generalizing this discussion to polyhedra would be non-trivial.

A more promising future direction would be to extend the motion planning framework. These extensions can be viewed as either extending the low-level design, or extending the high-level design. An extension at the low-level that is already being pursued is to extend the basic Forward, Backward, Hold motion primitives to more complex motion primitives. This could include having different nominal speeds for the Forward and Backward modes, or developing motion primitives which are designed explicitly for \( p = 2, 3, \ldots \) instead of composing atomic motion primitives. Another low-level extension could be to design motion primitives for dynamical systems which do not satisfy the translational invariance assumption, such as unicycle dynamics. The first, most obvious, extension for the high-level design would be to extend the reach-avoid objective to a more general LTL specification. This can be achieved by taking the synchronous product of the product automaton with a Büchi Automaton. Solving the motion planning problem on this new automaton would satisfy the LTL specification. Another high-level extension would be to alter the problem formulation such that each mobile robot solves its own motion planning framework. Other mobile robots would now be seen as obstacles, and therefore the product
automaton of each robot would be dynamic. What would be left is to determine conditions for when motion planning with a dynamic product automaton is solvable.
Bibliography


