

FRACTIONAL CONTROL OF MULTIVEHICLE SYSTEMS AND
RELATIVE ORBITS

by

David Yaylali

Copyright © David Yaylali 2018

A Thesis Submitted to the Faculty of the

DEPARTMENT OF AEROSPACE AND MECHANICAL ENGINEERING

In Partial Fulfillment of the Requirements

For the Degree of

MASTERS OF SCIENCE

WITH A MAJOR IN AEROSPACE ENGINEERING

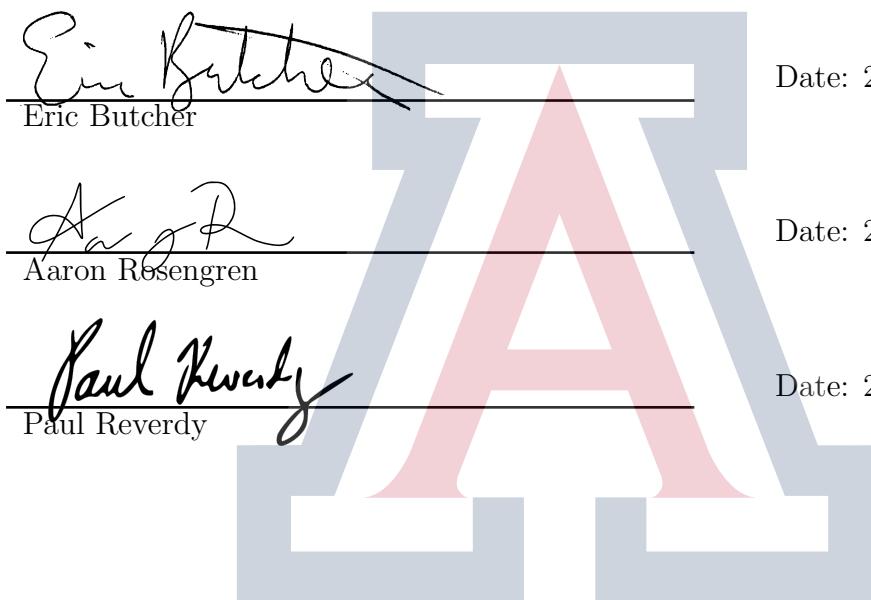
In the Graduate College

THE UNIVERSITY OF ARIZONA

2018

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Master's Committee, we certify that we have read the thesis prepared by David Yailali, titled Fractional Control of Multivehicle Systems and Relative Orbits and recommend that it be accepted as fulfilling the thesis requirement for the Degree of Masters of Science.



Eric Butcher

Date: 28 November 2018

Aaron Rosengren

Date: 28 November 2018

Paul Reverdy

Date: 28 November 2018

Final approval and acceptance of this thesis is contingent upon the candidate's submission of the final copies of the thesis to the Graduate College.

I hereby certify that I have read this thesis prepared under my direction and recommend that it be accepted as fulfilling the Masters of Science thesis requirement.

Date: 28 November 2018

Eric Butcher
Professor
Aerospace and Mechanical Engineering

ACKNOWLEDGEMENTS

I am immensely grateful to my research adviser Eric Butcher for welcoming me into the aerospace program here at the University of Arizona, introducing me to the fascinating fields of control theory and astrodynamics, and taking me on as a research student. I am also grateful to the faculty and staff here at the Department of Aerospace and Mechanical Engineering, including Aaron Rosengren, Paul Reverdy, Jini Kandyil, Mathieu Joerger, and Cholik Chan. For enriching conversation and friendship, I also thank the graduate students I have worked with, including Mohammad Maadani, Ethan Burnett, Jingwei Wang, B. Malladi, and Arman Dabiri.

Lastly and most importantly, I give my deepest gratitude to my wife Hannah for the love, support, and encouragement throughout my program. May this work, and the many late nights away from you that made this possible, allow our ship to come in.

TABLE OF CONTENTS

LIST OF FIGURES	6
ABSTRACT	10
CHAPTER 1 INTRODUCTION	11
1.1 CONTROL SYSTEMS FOR SECOND-ORDER SYSTEMS	15
1.1.1 State-Space Representation	16
1.1.2 The State Transition Matrix	19
1.1.3 PD and PID Control	24
1.1.4 The Linear Quadratic Regulator as an Optimal Controller	29
1.1.5 Vehicle Control in Three Dimensions	33
1.2 MULTIAGENT SYSTEM FORMALISM AND CONTROL	36
1.2.1 Communication Topology and Algebraic Graph Theory	37
1.2.2 General Second-Order Consensus Protocols	41
1.2.3 Stability of the Consensus Protocol	47
1.2.4 Generalization to Three Dimensions	55
1.3 FRACTIONAL CALCULUS AND CONTROL	57
1.3.1 Motivation	58
1.3.2 Formal Definitions	61
1.3.3 Fractional Controllers	68
1.3.4 Numerical Methods: The Fractional Chebyshev Collocation Method	73
1.4 REVIEW OF ORBITAL MECHANICS	84
1.4.1 State Vectors for Orbiting Satellites	85
1.4.2 Dynamics of Orbital Motion	88
1.4.3 Relative Orbits	89
1.5 CONCLUDING REMARKS	91

TABLE OF CONTENTS – *Continued*

CHAPTER 2 FRACTIONAL COOPERATIVE CONTROL	94
2.1 PID CONSENSUS PROTOCOLS	94
2.1.1 Eigenanalysis of the Closed-loop System and Stability	96
2.2 FRACTIONAL PID CONSENSUS CONTROL	104
2.2.1 Equations of Motion in Terms of the Fractional Pseudostate	105
2.2.2 Closed-loop Equations of Motion	108
2.3 STABILITY OF PID $^\alpha$ CONSENSUS CONTROL	111
2.3.1 Eigenanalysis of the PID $^\alpha$ Controlled System	111
2.3.2 Proof of Stability for PID $^\alpha$ Controlled Systems	112
2.3.3 Stability Conditions on PID $^\nu$ Controller Gains	119
2.4 NUMERICAL SIMULATIONS AND RESULTS	122
2.4.1 Performance Comparisons Between PID and PID $^\alpha$ Controllers	122
CHAPTER 3 RELATIVE ORBIT CONTROL	128
3.1 RELATIVE ORBITS	128
3.1.1 Relative Orbits in the LVLH Frame	130
3.1.2 Transforming Between Different Relative Orbit Descriptions	132
3.1.3 Dynamics and Linearized Equations of Motion	135
3.1.4 Relative Orbits in HCW Dynamics	138
3.1.5 Relative Orbit Control	143
3.2 FRACTIONAL CONTROL OF RELATIVE ORBITS	151
3.2.1 Decoupling the HCW Equations Using Fractional Control	153
3.2.2 LQR-Designed Controller Using Fractional Derivatives	157
3.2.3 Numerical Survey Techniques in Fractional Controller Design	160
3.3 CONSENSUS AND FORMATION CONTROL FOR HCW DYNAMICS . .	164
3.3.1 Arranging Spacecraft Formations Using Consensus Protocols	169
CHAPTER 4 CONCLUSIONS	175
REFERENCES	178

LIST OF FIGURES

1.1 Trajectory and control signal for the PD-controlled double integrator. Compared are the trajectories corresponding to a critically damped oscillator and the trajectories from the LQR-optimized solution using Q and R matrices in Eq. (1.26). 1.2 Computed cost for the PD-controlled double integrator as a function of k_D , for $k_P = 1$. We see that the minimum cost (red star) is achieved at the LQR computed value $k_D = 1.73$ 1.3 Pictoral representation of a directed communication graph for a 6-agent system. Note that the lack of arrow-heads between nodes ν_1 and ν_3 imply that $(\nu_1, \nu_3) \in E$ and $(\nu_3, \nu_1) \in E$, <i>i.e.</i> , this edge is undirected. 1.4 Fractional derivative $D^\alpha x$ (left), as defined in Eq. (1.57), for various fractional orders α . Included for reference is the standard first-order derivative, $D^1 x = 1$. This fractional derivative is defined using the Euler gamma function (right). 1.5 The behavior of the solutions to $D^\alpha x(t) = -x(t)$, given by $E_\alpha(-t^\alpha)$, for various choices of α . The creeping behavior of the solutions is seen as $\alpha \rightarrow 0$ 1.6 Illustration of Matignon's theorem, Eq. (1.73). 1.7 Collocation points as horizontal projections of equally spaced points around the unit circle (left). At right we show the distribution of these collocation points as we increase N 1.8 A few select Chebyshev polynomials $T_n(x)$. The density of the extremal points (CGL points) increases near the domain endpoints $\{-1, 1\}$ 1.9 Two methods of characterizing the state of an orbiting satellite. At left we show the axes of the ECI cartesian coordinate system, and at right we show the Keplerian orbital elements. Note that orbital elements a and e are not shown. 2.1 Hypersurfaces bounding the stability regions in $\{k_P, k_I, k_D\}$ parameter space for a N -agent system of double-integrators under PID consensus control. Regions below the surfaces are stable. These surfaces correspond to the non-zero Laplacian matrix eigenvalue μ_i , with $\mu_i \in \mathbb{R}$ (left) and $\mu_i = 1 + i$ (right). Note that the actual stability region for the system is the intersection of the regions corresponding to $\mu_2, \mu_3, \dots, \mu_N$ 	33 33 38 60 67 72 76 77 86 99
--	--

LIST OF FIGURES – *Continued*

2.2	Communication graphs for three benchmark 5-agent systems. System A is an undirected complete graph, describing a fully connected topology. System B is a directed graph (digraph) representing a leader-follower communication topology. System C is a 5-cycle graph representing a cyclic communication topology.	101
2.3	Simulation of System A under PID consensus control. From left to right, we show values of k_I corresponding to stable, marginally stable, and unstable systems as determined from Eq. (2.12).	103
2.4	Same as Fig. 2.3, but simulating response of System B. We simulate for a much longer time window due to the slower response of this system. These plots also confirm the stability conditions on the PID gains.	104
2.5	Same as Figs. 2.3 and 2.4, but simulating response of System C.	104
2.6	Illustrating the stability conditions on gains for a fully-connected system under fractional PID $^\alpha$ control. For this 3-agent system, we have taken derivative order $\alpha_D = 0.5$ and chosen $k_I = 2$ and $K_D = 1$. Using Eq. (2.57), we find that the system will be stable for $k_P > 0.33$. We show choices of k_P at, above, and below this stability limit. In the right panels we show the location of the eigenvalues of the system matrix A_{CL} corresponding to these choices of k_P . These plots agree with our result that the system is asymptotically stable if the non-zero eigenvalues of A_{CL} satisfy $ \arg \lambda_i > \alpha \frac{\pi}{2}$	121
2.7	Comparisons between integer- and fractional order consensus for the three systems in Eq. (2.16). In the left panels we show the system response from an integer order PID controller in red and the response from a fractional order PID $^\alpha$ controller (using the approximately optimal choice of α_D) in blue. In the right panels we show the settling time and overshoot for the system as a function of fractional derivative order α_D	124
2.8	Contours of settling time as a function of k_I and k_D for System A and taking $k_P = 1$. At left we show the contours for the fractional PID $^\alpha$, taking $\alpha_D = 0.75$, and at right we show the same results using integer order controller. After surveying over α_D , we have found that the minimum settling time of $t_s = 0.862$ s is achieved with a fractional controller using $\alpha_D = 0.75$, $k_D = 0.6$, and $k_I = 0.25$; this point is marked with a purple star. The minimum settling time for the integer-order controller is $t_s = 1.067$ s.	125
2.9	Same as Fig. 2.8, but showing contours of the geometric mean of settling time and integrated control effort. In this case we find the optimal value to be 5.78 using the fractional controller with $\alpha_D = 0.90$, $k_D = 0.7$, $k_I = 0.2$. The minimum value using the integer-order controller is 5.83.	126

LIST OF FIGURES – *Continued*

3.1	The LVLH coordinate frame for an orbiting spacecraft.	131
3.2	Relative orbit trajectory for the initial conditions in Table 3.1. We plot the trajectory from integrating the HCW equations (purple) along with the trajectory found by propagating the Keplerian motion separately for chief and deputy and subsequently transforming to the LVLH frame.	139
3.3	Position error of HCW-propagated relative orbit trajectory compared to true relative orbit found by propagating chief and deputy Keplerian orbits separately. We plot trajectories using calibrated initial conditions $\tilde{\mathbf{x}}$ (purple) and exact initial conditions \mathbf{x} (teal).	143
3.4	Relative orbit trajectory under the decoupling PD controller described by Eqs. (3.23) and (3.24). Gains are chosen so that each component is critically damped. At top we show the 3D trajectory from two different view points, and in the bottom row we show each component of the trajectory separately. The uncontrolled relative orbit is shown for reference as the dotted lines.	147
3.5	Relative orbit trajectory under the LQR-designed controller described by Eqs. (3.23) and (3.31). The uncontrolled relative orbit is shown for reference as the dotted lines.	149
3.6	Comparison of trajectory (left) and control effort (right) for closed-loop system using decoupled PD control design and LQR control design. Control signals are in units of km/s ²	150
3.7	Trajectories under decoupled fractional PDD ^{α} control, for equal fractional orders $\alpha_x = \alpha_y = \alpha_z$. We show various values of fractional order, with $\alpha = 1$ corresponding to the standard integer-order controller.	156
3.8	Position components of the trajectories in Fig. 3.7.	156
3.9	Trajectories under the tuned fractional PDD ^{α} control, $\boldsymbol{\alpha} = [1.00, 0.85, 0.85]$, as compared to the integer-order PD-controlled trajectory.	157
3.10	Components of relative position (left) and control effort (right) for relative orbit trajectories under fractional control. We set $\alpha_y = 1$ and vary the order of the x -component derivative, α_x , using the LQR designed gains. The uncontrolled trajectory is shown as a dotted line.	158
3.11	Same as Fig. 3.10, but setting $\alpha_x = 1$ and varying the order of the y -component derivative, α_y	159
3.12	Relative performance of the optimal PD ^{α} controller as compared to the optimal integer-order controller. At left we show the results for the relative orbit defined in Table 3.1; the right plot corresponds to an alternative set of initial conditions as given in the text. In both cases, the optimal controller corresponds to the integer-order case $\alpha_x = \alpha_y = 1$	161

LIST OF FIGURES – *Continued*

3.13	Settling time as a function of controller gains k_{P_w} and k_{D_w} . Optimal choices of gains are marked with a red star. At left we show the optimal slice of α_w parameter space; the right panel corresponds to standard PD-type control.	163
3.14	Trajectories which optimize settling time for both integer- and fractional-order control (left) along with corresponding control signals (right).	163
3.15	LVLH-frame free trajectories of the two deputy relative orbit configuration defined by Eqs. (3.50) and (3.51).	166
3.16	LVLH-frame controlled trajectories (<i>c.f.</i> , Fig. 3.15). We show the trajectory at three different times during the propagation to show how consensus reached. We plot the free trajectories of both deputies as dotted lines for reference.	167
3.17	Free (left) and consensus-controlled (right) trajectories when Deputy 2 has non-zero initial drift in the LVLH frame. The consensus relative orbit now drifts relative to the LVLH frame. In the figure at right we also show the uncontrolled trajectories of both deputies as dotted lines for reference.	168
3.18	Illustration of the LISA gravitational wave observatory.	169
3.19	Placement of SC1 into the circular relative orbit using leader-follower consensus control. Solid line represents the physical deputy while dotted line represents the virtual (leader) deputy. Units are in 10^6 km.	173
3.20	Evolution of the LISA constellation under the leader-follower consensus control laws of the form Eq. (3.55). The black dots indicate the positions of the constellation after 6 complete orbits of the LVLH frame. We see that these controllers successfully bring the constellation from the payload origin to the points maximally separated along a circular relative orbit.	174
3.21	Radial distances of the three LISA constellation members from the LVLH origin (left). Shown at right are the relative distances between the constellation members. In both cases we see that these distances become constant (and equal) as the arrangement is formed.	174

ABSTRACT

Cooperative control protocols can be formulated for systems comprising multiple independent agents which can share information. In this work I will consider the cooperative control problem for multi-agent systems whose agents obey second-order Newtonian dynamics. Specifically, I will explore consensus and cooperative control laws for arrangements of both free point-mass bodies and point-mass spacecraft in orbit about a celestial body. In the latter case, the linearized equations of motion for two or more bodies in orbit will be used, allowing us to frame cooperative control laws in the standard formalism of algebraic graph theory. One of the primary novelties explored in this work is the usage of non-integer order integral and derivative operators in the feedback controllers for cooperative multiagent systems. These fractional operators provide additional degrees of freedom in controller design, and therefore afford more freedom in shaping the controlled system's trajectories. Among the main results presented in this work, we prove the stability of certain fractional consensus controllers and show that these controllers can outperform standard integer-order controllers in terms of some important performance measures.

Portions of the work presented in this thesis were accepted for presentation and publication within the proceedings of the 2019 AIAA Guidance, Navigation, and Control Conference and the 29th AAS/AIAA Space Flight Mechanics Meeting, cited as [1] and [2], respectively. Much of the initial work on relative orbit control was carried out during the Summer Faculty Fellowship Program at the Air Force Research Lab in Albuquerque, NM during the summer of 2018.

CHAPTER 1

INTRODUCTION

The opening ceremony of the 2018 Winter Olympics Games featured a carefully choreographed dance of 1,218 airborne illuminated drones. The appearance of this fantastical display, broadcast on the world stage, indicates that the technology of cooperative automated vehicles is beginning to flourish. There are indications that similar technology will become central in more practical applications including automated transportation, cooperative search and rescue, spacecraft formation flying, large aperture space telescopes, and space-based gravitational interferometers (for these and other applications, see [3, 4, 5] and the extensive lists of citations therein). With a little imagination, one can see this technology becoming ubiquitous and indispensable in the near future.

Perhaps the most important dynamical model in the study of such coordinated systems is that of the point-mass, described in the language of control theory as a double integrator second-order system. This prototypical system captures the essential dynamics of each vehicle, describing inertia through the notational rephrasing of Newton’s second law of motion

$$F = ma \quad \iff \quad \begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u,$$

where the double-integrator description on the right recasts force as a control signal u . The ability of one vehicle to move in coordination with another then relies on properly encoding information about the entire multivehicle system into the control u . This is broadly known as cooperative control.

Central to many of these applications is the utilization of consensus control protocols for double-integrator systems [6, 7]. Consensus in this context is the convergence of all vehicles to some common values in position and velocity, or to the convex hull defined by reference values in the case of a leader/follower consensus scenario [8]. Consensus protocols rely on

information sharing between the agents making up the system. A powerful approach to this problem utilizes algebraic graph theory to capture and characterize the communication topology and information flow between the agents [4].

For consensus control of double-integrator systems, the feedback control typically contains information on the relative positions and velocities of the agents. The control protocol for one agent can thus depend on the distance and relative velocity to all other agents with which it shares information. A common consensus control protocol involves feedback of the errors in the position differences between agents which share information as well as the relative velocities to build control protocols which drive these differences and errors to zero. These two components essentially make up a proportional-derivative (PD) consensus controller, whose performance can be tuned based on the relative weight (specified by gains) of the position and velocity feedback terms in the control protocol. Framing the standard consensus control protocol as a PD controller immediately suggests the possibility of adding a third term to the controller proportional to the integral of the position differences. These *PID consensus controllers* will be one of the central topics explored in Chapter 2 of this work. The additional tunable gain associated with the integral term affords one more freedom in shaping the response of the controller, and thus the speed and manner in which the agents in the system reach consensus.

The ability to easily tune performance is perhaps the main reason why PID controllers are so widely used in industrial and GNC applications. Despite this freedom, certain performance specifications resulting from PD/PID control are inherently coupled. For instance, one may want to design the controller to minimize both rise time and overshoot, but this could lead to unacceptably large control inputs. From a degree-of-freedom standpoint, in best-case scenarios one can only independently tune three performance criteria by adjusting these three gains independently. We therefore may be motivated to add a degree of freedom to our control protocol, so that we have more freedom in shaping the response. One innovative way of introducing extra degrees of freedom is by adjusting the action of the derivatives and integrals themselves.

Fractional calculus offers ways to assign meaning to non-integer orders of the differ-

entiation and integration operators [9]. In other words, fraction calculus explores the generalization

$$\left(\frac{d}{dx} \right)^{\text{integer}} \rightarrow \left(\frac{d}{dx} \right)^{\text{real or complex}}$$

Although presently a rather obscure subject, a vision of fractional calculus was in fact evident to the pioneers of the ordinary, integer-order calculus including Leibniz, and some years later Euler. It was not until the latter half of the 20th century, however, when applications of these formal concepts began to appear. Natural and mechanical systems which are more accurately or efficiently modeled by fractional-order differential equations of motion include lossy transmission lines, heat diffusion into semi-infinite solids, damping of a plate suspended in viscous liquid, fluid flow and rheology, and certain electric circuits (see [10, 11, 12] for reviews). Bagley and Calico found that viscoelastically damped structures exhibit behavior which is effectively described by fractional differential equations [13], and developed a fractional state-space formalism which simplified the description of such systems and suggested the use of fractional feedback control.

Fractional control strategies were developed in the late 20th century, led in large part by investigations by Oustaloup [14] and Podlubney [15]. In some cases, one seeks to control a system whose equations of motion involve non-integer-order derivatives or integrals; in others, one may seek to design controllers for integer-order systems using fractional derivatives or integrals of the system's state. In both cases, the closed-loop equations of motion for the system become fractional in nature. Investigations by Matignon [16], among others, led to important contributions in the understanding of stability of such systems. One of Podlubney's larger contributions was, as alluded to above, the generalization of PID controllers to fractional order $\text{PI}^{\mu}\text{D}^{\nu}$ controllers [15], with integral and derivative control of order μ and ν , respectively, where $0 < \mu, \nu \leq 1$. Integer-order PID control is obtained with $\mu, \nu = 1$. The fractional orders therefore act as additional "knobs" one can tune in order to shape the trajectory of the controlled closed-loop system. Moreover, in terms of specific performance measures, these fractional $\text{PI}^{\mu}\text{D}^{\nu}$ controllers can outperform their integer-order counterparts (see, *e.g.*, [1, 2, 17]).

A fractional $\text{PI}^{\mu}\text{D}^{\nu}$ controller was used for spacecraft attitude control in [18, 19, 20],

where it was demonstrated that the fractional controller performed better in terms of certain performance criteria such as the simultaneous optimization of both settling time and overshoot. Consensus control for double-integrator systems using integer-order PD controllers was considered in [7], while proportional-type consensus protocols for fractional-order systems was considered in [21]. Feedback of the integral of an error signal was incorporated into consensus controllers in [22], allowing a multiagent system to achieve consensus in the presence of persistent disturbances. To the best of my knowledge, however, utilization of integer- or fractional-order PID consensus controllers for double-integrator systems has not yet been considered. The majority of Chapter 2 will thus be dedicated to this topic.

Although the double integrator serves as a solid foundation upon which to study systems in inertial reference frames, there are many cases where noninertial dynamics cannot be ignored. Such is the case in orbital mechanics when studying the relative motion of spacecraft. For instance, when first learning the subject one is confronted with the extremely counterintuitive fact that in order to move “forward” (so as to catch up to a spacecraft which is ahead of you in an orbit) one must actually thrust “backward”. This is necessary in order to reduce the size of ones orbit, resulting in a faster orbital velocity. Therefore in this case the laws of motion seem reversed. In fact they are not reversed; the unfamiliar motion arises as a symptom of observing motion in a noninertial frame. In the close vicinity of a spacecraft in orbit, these noninertial dynamics give rise to “fictitious forces” which can essentially be captured by altering the form of the (three-dimensional) double integrator matrix. The dynamics, however, remain second-order, and thus the controllers introduced above can be applied without substantial modification. We will therefore explore the application of both fractional and consensus controllers to relative orbit dynamics in Chapter 3.

This first chapter, however, will essentially serve to introduce the reader to the fundamental theory and necessary formalism needed to discuss consensus and fractional control of second-order systems. Although the intended reader is expected to have a basic familiarity with control theory for dynamical systems, we will in Sec. 1.1 briefly review these concepts in the specific context of second-order dynamical systems. Section 1.2 introduces

the theory and formalism of cooperative control for multiagent systems; central to this topic is the characterization of the communication topology shared between the individual agents, which is efficiently captured by so-called communication graphs and their associated eigenstructure. Section 1.3 introduces fractional calculus, and the utilization of fractional integral- and derivative-operators in feedback control laws. The topics in Sec. 1.2 and 1.3 will be merged in the following chapters in the formulation of fractional consensus control laws. In Sec. 1.4 we will review Keplerian orbital mechanics, which will be foundational in our construction of relative orbit controllers.

This stylistic structure of this chapter is highly pedagogical, and a high level of detail is given in the exposition. As a result, this introduction comprises a substantial fraction of this thesis. I will therefore conclude this chapter in Sec. 1.5 by restating (for the benefit of the reader) the central topics which will be explored in the main body of this work, and the main results which will be presented therein.

1.1 CONTROL SYSTEMS FOR SECOND-ORDER SYSTEMS

This work is concerned with the dynamics and control of second-order systems. By “second-order systems,” we mean systems which obey Newtonian dynamics as summarized by Newton’s second law of motion,

$$\mathbf{F} = m\mathbf{a}. \quad (1.1)$$

Here, of course, \mathbf{F} represents the force vector acting upon a body of mass m and \mathbf{a} is the resulting acceleration vector of the body. The terminology “second-order” refers to the fact that Newton’s law is a second-order (vector) differential equation, as acceleration is the second-order time-derivative of position, $\mathbf{a} = d^2\mathbf{r}(t)/dt^2$.

In less formal terms, second-order systems are synonymous with systems which have *inertia*. As such systems constitute the primary physical entities of classical physics, the importance of second-order systems need not be stressed. In the closely-related context of engineering, and more specifically aerospace engineering, the primary second-order entities under consideration include *vehicles* such as aircraft, rockets, and spacecraft.

Control systems for second-order systems serve to impart an additional force to the dynamics of the system for the purposes of altering (or *guiding*) the natural trajectory of the system to some desired trajectory. A controlled second-order system can therefore be written as

$$m\mathbf{a} = \mathbf{F} + \mathbf{F}_{\text{control}},$$

where \mathbf{F} describes the natural or environmental forces on the body (*e.g.*, gravity, drag, *etc.*) and $\mathbf{F}_{\text{control}}$ describes the forces applied by the controller. These latter forces can be assumed to be provided by some internal mechanism within the body such as rocket or jet thrusters, but in many cases can be provided by manipulating the external environment (*by*, *e.g.*, deploying sails or altering airflow by adjusting control surfaces). In control theory it is common to divide out the mass of the body (or re-scale our units) such that the controlled dynamics take the form

$$\mathbf{a} = \frac{d^2\mathbf{r}}{dt^2} = \frac{\mathbf{F}}{m} + \frac{\mathbf{F}_{\text{control}}}{m} \quad \Rightarrow \quad \ddot{\mathbf{r}} = \mathcal{F}(\mathbf{r}, \dot{\mathbf{r}}, t) + \mathbf{u}, \quad (1.2)$$

where \mathcal{F} is a vector function containing the natural dynamics of the system and \mathbf{u} is known as the *control signal* provided by the controller. We have assumed here that the natural dynamics can be written as a function only of the body's current position, velocity, and time, though \mathcal{F} can in general include more complicated dynamics (*e.g.*, position relative to other external bodies).

1.1.1 State-Space Representation

Let us now express the second-order dynamics in Eq. (1.2) as a system of first-order equations:

$$\ddot{\mathbf{r}} = \mathcal{F}(\mathbf{r}, \dot{\mathbf{r}}, t) + \mathbf{u} \quad \Rightarrow \quad \begin{cases} \dot{\mathbf{r}} = \mathbf{v}, \\ \dot{\mathbf{v}} = \mathcal{F}(\mathbf{r}, \dot{\mathbf{r}}, t) + \mathbf{u}. \end{cases} \quad (1.3)$$

When the uncontrolled dynamics of a second-order system are *linear*, the vector function $\mathcal{F}(\mathbf{r}, \dot{\mathbf{r}}, t)$ can be expressed in the general form

$$\mathcal{F}(\mathbf{r}, \dot{\mathbf{r}}, t) = \mathbf{f}(t)\mathbf{r} + \mathbf{g}(t)\dot{\mathbf{r}}.$$

where \mathbf{f} and \mathbf{g} are, in general, time-dependent (3×3) matrix functions. This allows us to write Eq. (1.3) as a first-order *matrix* equation:

$$\begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & I_3 \\ \mathbf{f}(t) & \mathbf{g}(t) \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ I_3 \end{bmatrix} \mathbf{u}, \quad (1.4)$$

where I_3 is the 3×3 identity matrix and $\mathbf{0}$ is the 3×3 matrix of zeros. In 3-dimensional Newtonian dynamics, the full configuration of the system at any time is characterized by the position \mathbf{r} and velocity \mathbf{v} ; this information thus fully characterizes the *state* of the system at any time, and allows us to predict the future behavior of the system at any later time through integration of the equations of motion Eq. (1.4). We therefore call the 6×1 column vector $[\mathbf{r}^T \ \mathbf{v}^T]^T$ the *state vector* \mathbf{x} of the system; the equations of motion are thus referred to as the *state equation* for the system.

We pause here to briefly comment on notation. Throughout this work, n -dimensional vectors will always be taken to be $n \times 1$ column-arrays when appearing in matrix equations. Additionally, when there is no ambiguity we will not explicitly specify the dimension of sub-matrices. For example, in Eq. (1.4), since we have already defined \mathbf{f} and \mathbf{g} as 3×3 matrices, and since I_3 is also 3×3 , there is no ambiguity as to the dimension of $\mathbf{0}$. When ambiguity exists or when it is helpful for clarity, we will specify the dimensions of such matrices as, *e.g.*, $\mathbf{0}_{m \times n}$.

The general state equation for a linear second-order system, Eq. (1.4), can be written in standardized notation as

$$\dot{\mathbf{x}} = A(t)\mathbf{x} + B\mathbf{u}, \quad (1.5)$$

where $A(t)$ is known as the *state matrix* or *system matrix*, and B is referred to as the *input matrix*. The specific form of the input matrix (along with the top row of the system matrix) as given in Eq. (1.4) is the defining characteristic of Newtonian (second-order) dynamics; namely, that imparted forces give rise to accelerations. In general, the system matrix can be time-dependent (as written above); in this case, Eq. (1.5) describes the dynamics of a *linear, time-varying* (LTV) system. In cases where the dynamics do not explicitly depend on time, A is time-independent and the state equation

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \quad (1.6)$$

is referred to as a *linear, time-invariant* (LTI) system. In fact, in the absence of external or unaccounted-for forces, most linear systems encountered in nature are LTI (this is related to the fact that the Hamiltonians for systems which conserve energy are time-independent).

Let us consider the case of a free point-mass particle. In this case, no external forces act upon the particle, so that

$$\mathcal{F} = \mathbf{0} \quad \Rightarrow \quad \mathbf{f}(t) = \mathbf{0}, \quad \mathbf{g}(t) = \mathbf{0},$$

and the state equation reduces to

$$\begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & I_3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ I_3 \end{bmatrix} \mathbf{u}. \quad (1.7)$$

In control theory, this is known as the *3D double-integrator* system, and is defined by the system and input matrices

$$A = \begin{bmatrix} \mathbf{0} & I_3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{0} \\ I_3 \end{bmatrix}. \quad (1.8)$$

Of course since the sub-matrices within this system are diagonal, the individual \mathbf{r} and \mathbf{v} components of the state equation decouple so that each component is described by the 2-dimensional state-equation with state and input matrices given by

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (1.9)$$

This is simply referred to as the *double-integrator* system, with the specifier “1D” omitted. In both 1D and 3D cases, the designation “double-integrator” can always be thought of as synonymous with the dynamics of free point-mass particles.

The double integrator is ubiquitous in aerospace control theory applications; as it accounts for inertia, it typically can be used as a starting point in the construction of more complicated systems. In this way we can study the motion of vehicles under the influence of applied control forces. The double integrator will therefore be one of the prototypical systems studied in this work. In addition to this, we will also consider systems of multiple orbiting spacecraft. We will show that under certain conditions the dynamics describing the relative motion are time-independent and can be linearized. Therefore in this case the dynamics is also described by a second-order LTI system of the form Eq. (1.6), with input

matrix B taking the form as given in Eq. (1.8) and system matrix A possessing off-diagonal elements.

1.1.2 The State Transition Matrix

Let us now consider a general LTI dynamical system as described by the state equation

$$\dot{\mathbf{x}} = A\mathbf{x}, \quad (1.10)$$

where \mathbf{x} is a n -dimensional state vector and A is the system matrix describing the dynamics of the system. For now, we assume this is an uncontrolled system (so that this state equation follows from Eq. (1.6) after setting $\mathbf{u} = \mathbf{0}$); we will see below how A can in many cases “absorb” the dynamics induced by a controller. Given knowledge of the complete state at some initial time t_0 , $\mathbf{x}(t_0) \equiv \mathbf{x}_0$, the state equation Eq. (1.10) can be solved to provide the solution $\mathbf{x}(t)$ at any later time. The solution $\mathbf{x}(t)$ will be referred to as the *trajectory* of the system.

For systems whose states are described by real scalars $x \in \mathbb{R}$, the above state equation reduces to the first-order (scalar) differential equation

$$\dot{x} = ax,$$

where a is now a scalar quantity describing the dynamics of the scalar state $x(t)$. Given an initial condition $x(t_0) = x_0$, the solution to this differential equation is

$$x(t) = x_0 e^{a(t-t_0)}.$$

Similarly, for an n -dimensional state $\mathbf{x} \in \mathbb{R}^n$ with initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$, the solution to the state equation Eq. (1.10) can be expressed as [23],

$$\mathbf{x}(t) = \exp((t-t_0)A)\mathbf{x}_0, \quad (1.11)$$

where the right-hand side involves the *matrix exponential*,¹ defined for $n \times n$ matrices as

$$\exp M \equiv \sum_{k=0}^{\infty} \frac{M^k}{k!},$$

¹There is no ambiguity in using the same notation “ $\exp(\cdot)$ ” here, since the matrix exponential reduces to the standard exponential for scalar arguments.

with M^k representing the k -fold matrix power of M ,

$$M^0 = I_n, \quad M^k = \underbrace{MM \cdots M}_{k \text{ times}}.$$

Therefore, the matrix exponential of a general $n \times n$ matrix is itself an $n \times n$ matrix. The matrix exponential in Eq. (1.11) is known as the *state transition matrix* (STM),

$$\Phi(t, t_0) \equiv \exp((t - t_0)A), \quad (1.12)$$

or equivalently written as $\Phi(t, t_0) \equiv e^{(t-t_0)A}$.

Although there are alternative (and, often, more convenient) ways of deriving the STM, the matrix exponential definition of Φ is useful when the system dynamics Eq. (1.10) is written in *modal* form. Let us assume for now that the eigenvalues λ_k of the system matrix A are non-degenerate. In this case, the system matrix can be diagonalized as

$$P^{-1}AP = \Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}, \quad (1.13)$$

where P is the change-of-basis matrix whose columns are the (right) eigenvectors of A :

$$A\mathbf{v}_i = \lambda_i \mathbf{v}_i, \quad i = 1, 2, \dots, n \quad \Leftrightarrow \quad P = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_n].$$

Inverting Eq. (1.13) as $A = P\Lambda P^{-1}$, the STM can be evaluated as the matrix exponential

$$\Phi(t, t_0) = e^{(t-t_0)A} = e^{(t-t_0)P\Lambda P^{-1}} = \sum_{k=0}^{\infty} \frac{(t-t_0)^k (P\Lambda P^{-1})^k}{k!}.$$

The matrix product within the sum, however, simplifies as

$$(P\Lambda P^{-1})^0 = I_n, \quad (P\Lambda P^{-1})^k = (P\Lambda P^{-1})(P\Lambda P^{-1}) \cdots (P\Lambda P^{-1}) = P\Lambda^k P^{-1},$$

so that the STM can be written as

$$\Phi(t, t_0) = e^{(t-t_0)P\Lambda P^{-1}} = Pe^{(t-t_0)\Lambda}P^{-1},$$

and since Λ is diagonal, this is written in closed form as

$$\Phi(t, t_0) = P \begin{bmatrix} e^{\lambda_1(t-t_0)} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2(t-t_0)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n(t-t_0)} \end{bmatrix} P^{-1}. \quad (1.14)$$

Note that by defining the *modal coordinates* $\mathbf{z}(t) = P^{-1}\mathbf{x}(t)$, the original system is equivalent to a system of n decoupled first-order equations,

$$\dot{\mathbf{x}} = A\mathbf{x} \Rightarrow P^{-1}\dot{\mathbf{x}} = P^{-1}APP^{-1}\mathbf{x} \Rightarrow \dot{\mathbf{z}} = \Lambda\mathbf{z} \Rightarrow \begin{cases} \dot{z}_1 = \lambda_1 z_1 \\ \dot{z}_2 = \lambda_2 z_2 \\ \vdots \\ \dot{z}_n = \lambda_n z_n \end{cases}$$

The closed-form expression for the STM given in Eq. (1.14) can be generalized to cases in which two or more eigenvalues of A are degenerate. In this case, the modal decomposition performed above will lead to a block matrix of the form

$$A = PJP^{-1} = P \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_m \end{bmatrix} P^{-1}, \quad (1.15)$$

where all omitted elements are understood to be zero. Here, J_k , $k = 1, 2, \dots, m \leq n$ are *Jordan blocks* of the form

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix},$$

where λ_i corresponds to a degenerate eigenvalue. The matrix J is known as the *Jordan matrix*, and can be written concisely as

$$J = J_1 \oplus J_2 \oplus \cdots \oplus J_m.$$

The number of Jordan blocks and their corresponding dimensions are determined by the algebraic and geometric multiplicities of the Eigenvalues λ_k . For instance, if a matrix has

4 degenerate eigenvalues, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = p$ of geometric multiplicity 1, the Jordan matrix will contain 1 Jordan block

$$J_p = \begin{bmatrix} p & 1 & 0 & 0 \\ 0 & p & 1 & 0 \\ 0 & 0 & p & 1 \\ 0 & 0 & 0 & p \end{bmatrix}.$$

If, on the other hand, the geometric multiplicity is determined to be 2, the Jordan matrix will contain *two* Jordan blocks,

$$J_{p_1} = \begin{bmatrix} p & 1 \\ 0 & p \end{bmatrix}, \quad J_{p_2} = \begin{bmatrix} p & 1 \\ 0 & p \end{bmatrix}.$$

In both cases, the algebraic multiplicity is equal to 4, the number of degenerate eigenvalues.

The geometric multiplicity of λ_i is equal to the number of linearly independent eigenvectors associated with eigenvalue λ_i . We will illustrate by example: Consider the matrix

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

It can be easily determined that the eigenvalues are all equal to 1,

$$\lambda_1 = \lambda_2 = \cdots = \lambda_5 = 1 \equiv \lambda,$$

so that the algebraic multiplicity of $\lambda = 1$ is 5. The eigenvectors \mathbf{v}_i are given by the null-space of $(A - \lambda I)$,

$$(A - \lambda I)\mathbf{v}_i = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{v}_i = \mathbf{0} \quad \Rightarrow \quad \mathbf{v}_1 = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

We therefore find 3 linearly independent eigenvectors associated with $\lambda = 1$. This implies that the geometric multiplicity of $\lambda = 1$ is 3, and the Jordan matrix associated with A will therefore be built from three Jordan blocks. To determine the dimensions of the

Jordan blocks, we determine the number of generalized eigenvectors \mathbf{x} associated with each eigenvector:

$$(A - \lambda I)\mathbf{x} = \mathbf{v}_1$$

$$(A - \lambda I)\mathbf{x} = \mathbf{v}_2$$

$$(A - \lambda I)\mathbf{x} = \mathbf{v}_3$$

It can easily be seen that the first and second equations do not have solutions, so that \mathbf{v}_1 and \mathbf{v}_2 do not have generalized eigenvectors. The Jordan matrix therefore has two Jordan blocks of dimension 1 and a third Jordan block of dimension 3:

$$J_1 = [1], \quad J_2 = [1], \quad J_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

There are therefore two generalized eigenvectors associated with the third eigenvector \mathbf{v}_3 . These are given by

$$\begin{aligned} (A - \lambda I)\mathbf{v}_4 &= \mathbf{v}_3 & \Rightarrow \quad \mathbf{v}_4 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{v}_5 = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}. \\ (A - \lambda I)\mathbf{v}_5 &= \mathbf{v}_4 \end{aligned}$$

The change-of-basis matrix P which transforms between A and J is now build from the eigenvectors and generalized eigenvectors,

$$P = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3 \ \mathbf{v}_4 \ \mathbf{v}_5] \quad \Rightarrow \quad P^{-1}AP = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = J_1 \oplus J_2 \oplus J_3 = J.$$

We now return to the STM for an LTI system, Eq.(1.12), in the general case where eigenvalues of A can be degenerate. The modal expansion of this STM is given by

$$\Phi(t, t_0) = e^{(t-t_0)A} = Pe^{(t-t_0)J}P^{-1} = P \begin{bmatrix} e^{(t-t_0)J_1} & & & & \\ & e^{(t-t_0)J_2} & & & \\ & & \ddots & & \\ & & & & e^{(t-t_0)J_m} \end{bmatrix} P^{-1}. \quad (1.16)$$

The problem of determining a closed-form solution for Φ has now been reduced to finding the closed-form expression for the individual block matrices $\exp((t - t_0)J_k)$. We in fact find that, for a Jordan block J_k of dimension n ,

$$e^{\tau J_k} = e^{\lambda_k \tau} \begin{bmatrix} 1 & \tau & \frac{\tau^2}{2!} & \cdots & \frac{\tau^{n-1}}{(n-1)!} \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \frac{\tau^2}{2!} \\ & & & 1 & \tau \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}. \quad (1.17)$$

Note that for $\operatorname{Re} \lambda_k < 0$, this exponential vanishes asymptotically (*i.e.*, as $t \rightarrow \infty$). This fact is equivalent to the more familiar statement that an LTI system $\dot{\mathbf{x}} = A\mathbf{x}$ is stable if and only if all eigenvalues of A are in the left-half complex plane (the matrix A is *Hurwitz*).

We can therefore find the trajectory of $\mathbf{x}(t)$ of an LTI system $\dot{\mathbf{x}} = A\mathbf{x}$ analytically, given an initial condition \mathbf{x}_0 , by computing the STM via Eqs. (1.16) and (1.17). In the case where all eigenvalues of A are non-degenerate, the Jordan blocks in Eq. (1.16) are one-dimensional and the exponentials are simply $e^{\lambda_k(t-t_0)}$.

1.1.3 PD and PID Control

“The career of a young theoretical physicist consists of treating the harmonic oscillator in ever-increasing levels of abstraction.”

– Sidney Coleman

We will now make a first attempt to build a controller for a second-order system. Let us take our dynamical system to be a double integrator (a point mass in 1-dimension), and let us imagine that we wish to stabilize this system about $\mathbf{x}^* = [r^* \ r^*]^T = \mathbf{0}$. That is, we would like to build a controller which brings the system to rest at the origin regardless of the initial position or velocity.

We can imagine how one could build this controller in practice: simply attach a spring to the mass, and hammer the other end of the spring to the table-top at the point we define to be the origin. Moreover, in order to damp out the resulting oscillatory motion, we could

cover the surface of the table with sandpaper or some viscous substance. We have, of course, simply built a damped harmonic oscillator which evolves according to the equation

$$m\ddot{r} = -kr - b\dot{r},$$

where k characterizes the stiffness of the spring we have picked out of our toolbox and b characterizes the “grit” of the sandpaper we have decided to use. In the context of the LTI state description, however, what we have done is add to the double integrator a control signal:

$$\underbrace{\ddot{r} = 0}_{\text{double integrator}} + \underbrace{\left(-\frac{k}{m}r - \frac{b}{m}\dot{r}\right)}_{\text{control signal } u}.$$

We can therefore write the state equation as

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left(-\frac{k}{m}r - \frac{b}{m}\dot{r}\right).$$

We now realize that the control signal is just a linear function of position and velocity, and thus a linear function of the state. It can be written as

$$u = -k_P r - k_D \dot{r} = [k_P \ k_D] \begin{bmatrix} r \\ \dot{r} \end{bmatrix} = -[k_P \ k_D] \mathbf{x}, \quad (1.18)$$

where we have defined $k_P = k/m$ as the gain of the position-*proportional* term, and $k_D = b/m$ as the gain of the position-*derivative* term. These are indeed controller gains, since they depend on which spring and sandpaper grit we have chosen; in other words, k_P and k_D are *tunable*. The form of the controller given in Eq. (1.18) is appropriately referred to as a *proportional-derivative* (PD) controller. Due to its simplicity and stability (as will be revealed below), these types of controllers are ubiquitous in controller design and are used in a wide variety of industrial applications.

We can now rewrite our state equation utilizing the form of u in Eq. (1.18) as

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} [k_P \ k_D] \mathbf{x} \\ &= \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ k_P & k_D \end{bmatrix} \right) \mathbf{x} \\ &= \underbrace{\begin{bmatrix} 0 & 1 \\ -k_P & -k_D \end{bmatrix}}_{A_{CL}} \mathbf{x}.\end{aligned}$$

The matrix A_{CL} can therefore be thought of in two subtly different ways. In one sense, it is simply the dynamical system matrix for a damped-harmonic oscillator. In another sense, which is more relevant to our current purposes, it is the *closed-loop* system matrix of a double integrator under PD control (hence the notation A_{CL}). An important quality of PD-controlled double integrators is that the eigenvalues of A_{CL} are

$$\lambda_{\pm} = \frac{1}{2}(-k_D \pm \sqrt{k_D^2 - 4k_P}),$$

which are in the left-half complex plane for all $k_P, k_D > 0$. The closed-loop matrix is therefore Hurwitz for all choices of positive gains, implying that the system is stable about the equilibrium point $\mathbf{x}^* = \mathbf{0}$ and the system comes to rest at the origin. This is of course the expected behavior of the tabletop spring-sandpaper controller we have built in this example.

The ability to write the controlled dynamical system above in the closed-loop form $\dot{\mathbf{x}} = A_{CL}\mathbf{x}$ relied solely on the ability to write our controller in the form $u = -K\mathbf{x}$, with K a general matrix (of compatible dimension with u and \mathbf{x}). Controllers of this form are known as *full-state feedback* controllers since the controller “feeds back” the complete current state of the system to the dynamics. For a *general* LTI system under full-state feedback,

$$\left. \begin{array}{l} \dot{\mathbf{x}} = A\mathbf{x} + Bu \\ u = -K\mathbf{x} \end{array} \right\} \Rightarrow \dot{\mathbf{x}} = (A - BK)\mathbf{x} = A_{CL}\mathbf{x}.$$

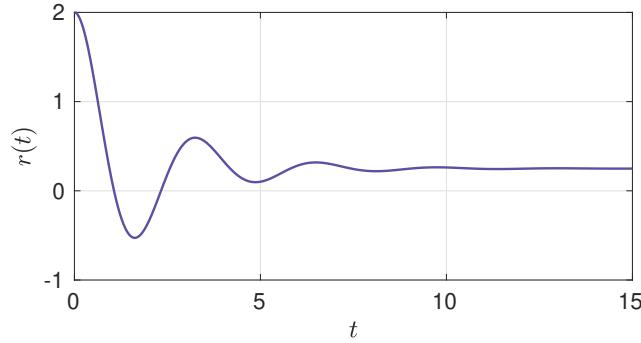
“PD” → “PID”: Including integral control

Let us now assume that there is some unaccounted-for disturbing force affecting our system. In our one-dimensional tabletop example above, we can imagine that we have inadvertently

mis-leveled our table. This results in a constant disturbing force to our system proportional to the projection of the gravitational acceleration vector along the surface of the (mis-leveled) tabletop. The state equation describing the dynamics of our PD-controlled system, now affected by this disturbance, is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -k_P & -k_D \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ a_g \end{bmatrix},$$

where a_g is the acceleration arising from the gravitational force pulling the mass down the slant of the table. The controller is unaware of this acceleration, since we have designed it assuming that the table is level. The response of the PD-controlled system, with $k_P = 4$, $k_D = 1$, $a_d = 1$, and initial condition $\mathbf{x} = [2, 0]$ is shown below:

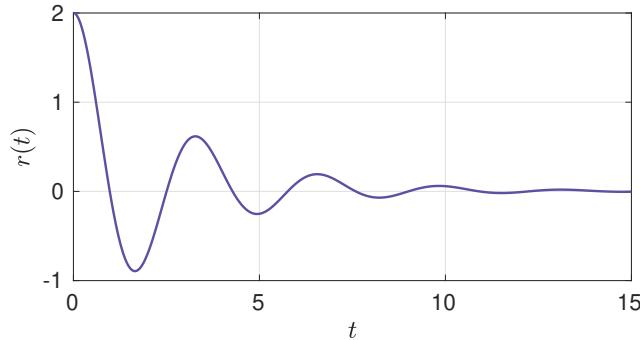


We see that the PD controller indeed brings the system to rest, but the steady-state position that is reached is offset from the desired equilibrium point $\mathbf{x}^* = [0 \ 0]^T$. This is an example of *steady-state error*, and is the result of the controller not taking into account disturbances to the assumed dynamics.

To mitigate steady state error, we realize that the time integral of the position in the above example will monotonically increase once the steady-state is reached. We are therefore motivated to include this time integral as an additional term to our controller:

$$u = -k_P r - k_D \dot{r} - k_I \int_{t_0}^t r(\tau) d\tau. \quad (1.19)$$

The addition of the integral term serves to cancel out any steady-state error in position. Adding the integral control term with $k_I = 1$ to the example considered above, the resulting trajectory becomes



We see that the integral term removes the steady state error caused by the tilt of our table, and the desired steady state $\mathbf{x}^* = [0 \ 0]^T$ is reached.

Equation (1.19) is an example of a *proportional-integral-derivative* (PID) controller. The closed-loop dynamics of a PID controlled system can be put into the form of an LTI state equation by introducing the auxiliary state component $\xi(t)$ and defining a new state as

$$\xi(t) = \int_{t_0}^t r(\tau) d\tau, \quad \mathbf{z} = [\xi \ r \ v]^T.$$

By definition, the initial condition $\xi(t_0)$ always vanishes. In the case of a free point-mass, the LTI state equation for this modified state is given by

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u.$$

This is an obvious generalization of the double integrator LTI system, and one could perhaps call this system a *triple integrator*.² Substituting the explicit form of the PID (full state feedback) controller, this becomes the closed-loop equation

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -k_I & -k_P & -k_D \end{bmatrix} \mathbf{z}.$$

For stability of the controlled system about the equilibrium point $\mathbf{z}^* = [0 \ 0 \ 0]^T$, the closed-loop system matrix above must be Hurwitz. This is equivalent to the statement that the eigenvalues λ_k of the above matrix, determined by the roots of the characteristic equation

$$\lambda^3 + k_D\lambda^2 + k_P\lambda + k_I = 0,$$

²To my knowledge this terminology is not widely used in the literature.

must be in the left-half complex plane. By applying the Routh-Hurwitz theorem³ (or equivalently by *tracing* the stability region by placing λ along the imaginary axis $\lambda = i\rho$, $\rho \in \mathbb{R}$), one finds the conditions on the gains which lead to system stability as

$$k_I < k_P k_D$$

for all gains positive. Therefore, unlike the PD controller, improper choice of controller gains can lead to an unstable closed-loop system.

These controllers are also ubiquitous in industrial settings; as illustrated above, one of their primary benefits is to remove steady state error induced by un-modeled disturbances. In addition, one benefit of PID controllers over PD controllers is an additional tunable parameter afforded by the integral gain K_I . Though this tunable gain provides more freedom in trajectory shaping, it also somewhat complicates controller design. We will confront this issue, and propose ways to optimally tune PID controllers, throughout this work.

1.1.4 The Linear Quadratic Regulator as an Optimal Controller

In the previous section, we introduced PD and PID full-state feedback controllers and (given certain conditions on the controller gains) proved their stability for double integrator systems. We have not, however, discussed how one goes about choosing the various gains. There are various ways in which this can be done. For instance, given that the stability of the closed-loop system requires all eigenvalues of A_{CL} to be in the left-half complex plane, one can “reverse engineer” the gains by first setting real parts of all eigenvalues to negative values. Generally, as the distance between these eigenvalues and the imaginary axis is increased, the response time of the system becomes smaller. This methodology is broadly known as *pole placement*, since the eigenvalues of A_{CL} correspond to the poles of the system’s transfer function.⁴ Another method to determine gains is to place requirements on a certain response

³The Routh-Hurwitz theorem provides an algorithmic test on a polynomial which determines whether all roots of this polynomial lie in the left-half complex plane. See, *e.g.*, [23].

⁴Briefly, transfer functions map the input to the output for a dynamical system, and can be found through Laplace transform of a system’s dynamical equations of motion. As we will not explicitly use transfer functions in this work, we point the reader to [23] for elaboration on this topic.

measure, such as *settling time* or *overshoot*; for second-order systems these can be directly related to the control gains.

In many cases, however, it is desirable to *simultaneously* optimize both the response characteristics and the control signal required to achieve this response. The methods mentioned above are not naturally suited for this purpose. For instance, through pole placement one can guarantee that a controller brings a system to equilibrium in a time characterized by some time constant τ , but this may lead to a controller which is impractical for a given application. In the case of vehicle control, this can correspond to controllers which are not fuel-efficient.

The simultaneous optimization of a system's response and the control signal used to generate that response is one of the central issues addressed in the broad field of *optimal control theory*. For our purposes, we will consider a specific problem in optimal control theory, which we frame as follows: Given a dynamical system

$$\dot{x}(t) = f(x, u, t), \quad (1.20)$$

where x is the system's state and u is some control signal, find the function $u(t)$ which optimizes some function

$$J\{u; x_f, t_f\} = \int_{t_0}^{t_f} L(x, u, t) dt + S(x_f, t_f). \quad (1.21)$$

We will call J the *cost function*, but formally (as indicated by the bracketed notation) J is a *functional*; that is, it is a function of functions, as its value depends on the functional form of $x(t)$ and $u(t)$. Note that in the context of this problem the dynamical equation implies that the trajectory $\mathbf{x}(t)$ is also a functional, as it depends on the particular control which optimizes J ,

$$\mathbf{x} = \mathbf{x}\{\mathbf{u}(t)\}.$$

Formally, then, the optimal control problem is an example of *constrained optimization* problem; we wish to optimize the cost function Eq. (1.21) under the constraint that the trajectory satisfies the dynamical equations Eq. (1.20). Much of the groundwork in solving these types of problems was laid out in the 17th and 18th century by prominent names

including Fermat and Lagrange, and these developments progressed in tandem with the formulation of *variational calculus*. Beautiful as it is, a sufficient discussion of these topics is beyond the narrative scope of this work. I therefore, somewhat regretfully, must refer the reader elsewhere for further discussion and for derivation of the following results. A particularly clear and insightful presentation of these topics can be found in Goldstein's well-known treatise on classical mechanics [24].

In what follows we will primarily be interested in LTI systems,

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{u} \in \mathbb{R}^m, \quad (1.22)$$

under full-state feedback control $\mathbf{u} = -K\mathbf{x}$, and we will be interested in finding the control \mathbf{u} which minimizes the particular cost function

$$J = \int_0^\infty (\mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}) dt. \quad (1.23)$$

where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are positive-definite matrices. Being a quadratic function of \mathbf{x} and \mathbf{u} , this constitutes what is known as the *linear quadratic regulator* (LQR) problem. Minimizing J in this case corresponds to, loosely speaking, simultaneously minimizing the time integral of the trajectory and control signal. Depending on how we choose Q and R , we can *weight* the various components of the state and control which we wish to minimize. The matrices Q and R are therefore design choices, and these can be chosen depending on the desired application. To roughly illustrate, in the example of unmanned spacecraft control, optimizing fuel consumption is typically more critical than optimizing the trajectory taken by the spacecraft. Therefore in these cases one typically puts more *weight* on the control term within the cost function by increasing elements of R relative to the elements of Q .

The solution to this optimal control problem is given by

$$\mathbf{u}(t) = -R^{-1}B^T P \mathbf{x}, \quad (1.24)$$

where the $n \times n$ matrix P is the solution to the continuous-time *algebraic Riccati equation* (CARE),

$$A^T P + PA - PBR^{-1}B^T P + Q = 0. \quad (1.25)$$

Once again, derivation of this result will take us too far off our desired path, so we will refer to reader in this case to texts discussing constrained optimization in the context of control theory, such as [25].

To briefly illustrate, let us consider the PD-controlled double integrator,

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad u = -K\mathbf{x},$$

with $K = [k_P \ k_D]$, and imagine we wish to optimize the cost function of Eq. (1.23) with

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1. \quad (1.26)$$

That is, we wish to optimize the trajectory $\{r(t), v(t)\}$ and the control effort $u(t)$ democratically. Given these selections for Q and R , the solution to the CARE along with the resulting optimal gain matrix is given by

$$P = \begin{bmatrix} 1.7321 & 1 \\ 1 & 1.7321 \end{bmatrix} \Rightarrow K = [k_P \ k_D] = [1 \ 1.7321].$$

We will compare the trajectories and control signals of this optimal solution to a slightly different choice of gains, $K = [1 \ 2]$, corresponding to a critically-damped harmonic oscillator. The trajectories $r(t)$ and control signals $u(t)$ for these two cases are shown in Fig. 1.1. Computing the cost function for both cases, we find

$$J_{LQR} = 6.9288,$$

$$J_{\text{crit}} = 7.0002.$$

In this case, we see the optimized trajectory indeed gives a lower cost compared to the critically damped trajectory.

To more completely demonstrate this optimization, we can choose $k_P = 1$ and survey over values of k_D . The resulting cost as a function of k_D is shown in Fig. 1.2. We indeed see that the cost is minimized at the LQR determined gain $k_D = 1.73$.

In this example, the LQR designed controller performs almost identically to the controller designed for critical damping. It therefore may seem that we do not gain much using the optimization, especially considering that the optimal gain was easily found from a simple

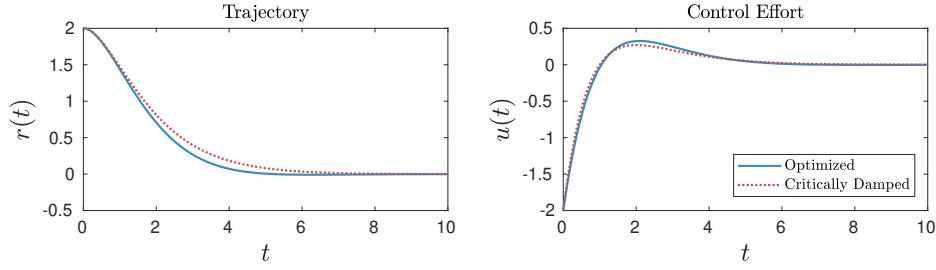


Figure 1.1: Trajectory and control signal for the PD-controlled double integrator. Compared are the trajectories corresponding to a critically damped oscillator and the trajectories from the LQR-optimized solution using Q and R matrices in Eq. (1.26).

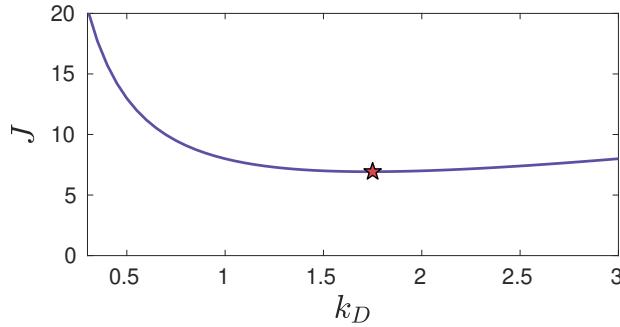


Figure 1.2: Computed cost for the PD-controlled double integrator as a function of k_D , for $k_P = 1$. We see that the minimum cost (red star) is achieved at the LQR computed value $k_D = 1.73$.

survey. As we will see, however, the LQR solution Eq. (1.24) is extremely convenient when the dimension of the dynamical system becomes large. In these cases, a large number of gains must be tuned, and often there is no obvious guiding principle which allows us to tune these gains manually. Accordingly, the LQR design strategy determines these gains for us automatically.

1.1.5 Vehicle Control in Three Dimensions

We will now discuss control protocols for second-order systems in three dimensions. Generally this introduces a large amount of complexity since the orientation and rotational inertia of the vehicle being controlled must be taken into account. For most systems, however, this *rigid-body* rotational motion is decoupled from the translational motion of the body's center-

of-mass (this follows from Euler's laws of motion); that is, we can discuss and analyze the rotational and translational motion separately. In this work we will only focus on translational motion in three dimensions. The bodies⁵ discussed in this work will therefore always be treated as *point masses*, and I will make an effort to point out along the way any issues that arise when this idealization encounters issues in real world applications.

The position vector $\mathbf{r} = [r_x, r_y, r_z]^T$ and velocity vector $\mathbf{v} = [v_x, v_y, v_z]^T$ describing the full configuration of a second-order system can be collected in the state vector $\mathbf{x} = [\mathbf{r}^T \ \mathbf{v}^T]^T \equiv [\mathbf{r}; \mathbf{v}]$. As introduced in Sec. 1.1.1, in the absence of external forces this state vector now evolves according to the 3D double integrator state equation,

$$\left. \begin{array}{l} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{u} \end{array} \right\} \Rightarrow \mathbf{x} = \begin{bmatrix} \mathbf{0} & I_3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ I_3 \end{bmatrix} \mathbf{u},$$

where $\mathbf{u} = [u_x, u_y, u_z]^T$ now is the control vector collecting the controller accelerations acting in each of the three dimensions.

Although the generalization to 3D translational motion at this point may seem trivial, the controller can induce coupling between the components in each of the three dimensions. As a direct example, consider the full-state feedback controller $\mathbf{u} = -K\mathbf{x}$. The gain matrix K now has dimension 3×6 , and mimicking the notation introduced in above we can write this as $K = [K_P \ K_D]$ with proportional and derivative gain matrices K_P and K_D both of dimension 3×3 . The closed-loop state equation then becomes

$$\dot{\mathbf{x}} = \underbrace{(A - BK)}_{A_{CL}} \mathbf{x} = \begin{bmatrix} \mathbf{0} & I_3 \\ -K_P & -K_D \end{bmatrix} \mathbf{x}. \quad (1.27)$$

If K_P and K_D are both chosen to be diagonal,

$$K_P = \text{diag}\{k_{Px}, k_{Py}, k_{Pz}\}, \quad K_D = \text{diag}\{k_{Dx}, k_{Dy}, k_{Dz}\},$$

⁵Note that it is common in dynamics and control literature (especially in cooperative control literature) to reserve the term “body” for an extended-mass system with rotational inertia; in other tribes this is typically referred to as a “rigid body”. I will make an effort to respect the former convention, but in some cases I will revert to a physicist’s terminology and use “body” and “point-mass” as synonyms. Since we will not discuss rotation in this work, no confusion should arise.

then as reflected by the block-diagonal form of A_{CL} the closed-loop system again decouples, with the component in each dimension evolving as three separate 1D damped harmonic oscillators:

$$\ddot{x} + k_{Dx}\dot{x} + k_{Px}x = 0, \quad \ddot{y} + k_{Dy}\dot{y} + k_{Py}y = 0, \quad \ddot{z} + k_{Dz}\dot{z} + k_{Pz}z = 0.$$

We therefore again find that the controller stabilizes the equilibrium ($\mathbf{r}, \mathbf{v} \rightarrow \mathbf{0}$) as long as all nonzero components of K (the diagonal components of K_P and K_D) are positive.

The closed-loop dynamics becomes more complicated, however, if we introduce a non-zero off-diagonal component in our controller's gain matrices K_P and K_D . As an example, let us add to the above diagonal gain matrices the off-diagonal components $[K_P]_{12} = k_{Pxy}$ and $[K_D]_{31} = k_{Dzx}$. The closed-loop equations of motion now become

$$\begin{aligned} \ddot{x} + k_{Dx}\dot{x} + k_{Px}x &= k_{Pxy}y \\ \ddot{y} + k_{Dy}\dot{y} + k_{Py}y &= 0 \\ \ddot{z} + k_{Dz}\dot{z} + k_{Pz}z &= k_{Pzx}\dot{x}. \end{aligned}$$

Furthermore, the stability of the system now relies on the eigenvalues of the *entire* closed-loop system matrix A_{CL} , as defined in Eq. (1.27), to lie in the left-hand side of the complex plane; that is, we require the matrix A_{CL} to be Hurwitz. Since these eigenvalues now depend on 18 potentially non-zero gains (the elements of K), the stability conditions on the gains become much more intricate.

To illustrate, consider the full-state feedback gain matrices

$$K_P = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix}, \quad K_D = \begin{bmatrix} 1 & 3 & 0 \\ 4 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

The eigenvalues of the closed-loop system are then given by

$$A_{CL} = \begin{bmatrix} \mathbf{0} & I_3 \\ -K_P & -K_D \end{bmatrix} \Rightarrow \text{eig } A_{CL} = \begin{cases} -0.6721 \\ -2.7912 \pm 0.3267i \\ +2.3386 \\ +0.4580 \pm 0.5951i \end{cases}$$

So, although all individual gains are positive (indicating positive damping and spring-constant forces), the equilibrium of the closed-loop system is unstable.

One might wonder whether the introduction of spatial-component coupling gains (off-diagonal elements of K_P and K_D) affords one any advantage. In the case of the 3D double integrator, since each spatial component is naturally decoupled, one can optimize each component separately. Optimal gain matrices in this case are therefore typically diagonal. If, however, the *natural* dynamics which we are studying couple the different spatial components, then diagonal gain matrices will often lead to *sub-optimal* system response. In these cases, introducing nonzero spatial-component coupling gains can often lead to better system response. As we will see in Chapter 3, spacecraft relative motion is an example of such a system.

1.2 MULTIAGENT SYSTEM FORMALISM AND CONTROL

Let us consider a collection of N individual physical entities, each of which evolve according to some specified dynamics. Let us further assume that these entities can communicate with each other, so that one agent can in some manner be made aware of the dynamical state of other entities. The shared information can thus be used in the control protocols used by each entity, serving to couple the closed-loop dynamics of the entire system. In the standardized terminology we will use throughout this work, each physical entity in this system is known as an *agent*, and such systems (where single-agent dynamics are coupled through inter-agent communication) are known generally as *multiagent systems*.

In this section we review the formalism of multiagent systems and consensus control protocols. Throughout this section (and indeed throughout much of this work) we will denote the number of agents by N and the number of states per agent n ; for example, a 3-agent second-order system has $N = 3$ and $n = 2$ (position and velocity). We also use the following shorthand for column-wise concatenation of p general column vectors w_i :

$$[w_1; w_2; \dots; w_p] \equiv [w_1^T \ w_2^T \ \dots \ w_p^T]^T$$

1.2.1 Communication Topology and Algebraic Graph Theory

Let us consider a collection of N agents, each of which evolves according to the same second-order dynamics. We assume that the agents can communicate with each other according to a prescribed constant communication network, so that the control signal for agent i can depend on its own state as well as those of its neighbors with which it is in communication. The communication flow between the agents is encoded in the *communication graph* \mathcal{G} of the system, and is efficiently characterized using the formalism of *algebraic graph theory*. We will review the necessary fundamentals of algebraic graph theory in this section; for more complete treatments, we refer the reader to [4, 5].

Let each agent (identified by the index $i \in \mathcal{N} = \{1, 2, \dots, N\}$) be associated with a system *vertex* (or, *node*) ν_i . The set of all vertices is denoted by \mathcal{V} . If agent i (node ν_i) provides information to agent j (node ν_j), we say that there exists an *edge* (ν_i, ν_j) . The set of all edges in the communication topology is denoted by \mathcal{E} , and we have $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. By definition, $(\nu_i, \nu_i) \notin \mathcal{E}$ so that there are no trivial “self-loops” in the communication topology. Pictorially, we can represent each node ν_i as a point and the edge (ν_i, ν_j) as an arrow from ν_i to ν_j , as illustrated for a specific 6-agent system in Fig. 1.3. In this example, the index, node, and edge set are respectively given by

$$\mathcal{N} = \{1, 2, 3, 4, 5, 6\}$$

$$\mathcal{V} = \{\nu_1, \nu_2, \nu_3, \nu_4, \nu_5, \nu_6\}$$

$$\mathcal{E} = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 1), (3, 5), (4, 6), (5, 6), (6, 2)\}$$

where, at little risk of ambiguity, we have here written the edges (ν_i, ν_j) more succinctly as (i, j) .

Thus, the communication topology for a given system is completely characterized by the *communication graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. If for any edge $(\nu_i, \nu_j) \in \mathcal{E}$ we also have $(\nu_j, \nu_i) \in \mathcal{E}$, the communication graph is said to be *bidirectional*; otherwise it is said to be *directed*. If edges exist such that one node can communicate to all other nodes in the system, the graph is said to have a *spanning tree*, and the graph is said to be *connected*; the first node along this communication path is known as the *root* node. Additionally, if every possible edge exists

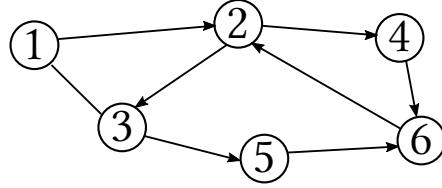


Figure 1.3: Pictoral representation of a directed communication graph for a 6-agent system. Note that the lack of arrow-heads between nodes ν_1 and ν_3 imply that $(\nu_1, \nu_3) \in E$ and $(\nu_3, \nu_1) \in E$, i.e., this edge is undirected.

(i.e., $(\nu_i, \nu_j) \neq 0$, $\forall i \neq j \in \mathcal{N}$), then the graph is said to be *complete* and the communication topology is *fully connected*; in this case, all nodes in the system are root nodes.

To illustrate these properties we again consider the communication graph in Fig. 1.3. This is a directed graph, since, e.g., $(\nu_1, \nu_2) \in \mathcal{E}$ but $(\nu_2, \nu_1) \notin \mathcal{E}$. Additionally, this graph is connected since it contains a spanning tree; specifically, it contains the spanning tree

$$1 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 2 \rightarrow 4$$

with node ν_1 being the root node; other spanning trees exist for this graph as well. The fact that $(\nu_2, \nu_1) \notin \mathcal{E}$ means, however, that this graph is *not* complete, and the topology is not fully-connected.

Multi-agent system matrices

Let us associate with each edge $(\nu_j, \nu_i) \in \mathcal{E}$ a weight $a_{ij} > 0$ (note the order of indices). These weights a_{ij} characterize, in some rough sense which will be clarified in the next section, the relative “importance” of the communication flow between the nodes in a graphs. If agent j does not communicate with agent i , then $(\nu_j, \nu_i) \notin \mathcal{E}$ and we set $a_{ij} = 0$. Note that for bidirectional graphs, $a_{ij} > 0 \Rightarrow a_{ji} > 0$. Bidirectional graphs with the additional condition $a_{ij} = a_{ji}$ are said to be *undirected*.

We also define the weighted *in-degree* of a node ν_i as $d_i = \sum_{j=1}^N a_{ij}$ and the weighted *out-degree* as $d_i^o = \sum_{j=1}^N a_{ji}$. If all non-zero weights are set to unity, then the in-degree counts the number of nodes which provide information to ν_i and the out-degree counts the number of nodes to which ν_i provides information.

Let us now populate the elements of an $N \times N$ matrix \mathcal{A} with the weights a_{ij} , which we will refer to as the *adjacency matrix*. Since by definition $(\nu_i, \nu_i) \notin \mathcal{E}$ (the graph contains no self-loops), the diagonal elements of \mathcal{A} are zero, $a_{ii} = 0$. Note that for undirected graphs, $a_{ij} = a_{ji}, \forall i, j \in \mathcal{N}$ and \mathcal{A} is symmetric. We also populate the diagonal elements of an $N \times N$ matrix \mathcal{D} with the in-degrees d_i : $\mathcal{D} \equiv \text{diag}(d_1, d_2, \dots, d_N)$. The adjacency and in-degree matrices for the example graph in Fig. 1.3, assuming unit weights, are given by

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

Note that the in-degree d_i (out-degree d_i^o) is simply the i^{th} row-sum (column-sum) of \mathcal{A} .

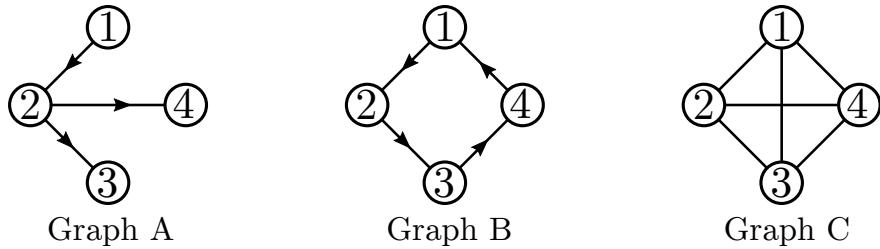
The communication flow is completely and efficiently encoded in the *graph Laplacian matrix*, defined as the difference between in-degree and adjacency matrices:

$$\text{Laplacian Matrix: } L = \mathcal{D} - \mathcal{A}. \quad (1.28)$$

The graph Laplacian will be central in the construction and analysis of cooperative control protocols for multiagent systems. Note that the row sums of L are identically zero, $\sum_{j=1}^N L_{ij} = 0$. From this fact it is clear that $\text{rank } L \leq N - 1$, and L has at least 1 zero-eigenvalue. We now highlight an important theorem which will be central to our stability analyses for cooperative control protocols:

Theorem 1. (Proposition 3.4 of [26]) For communication topologies which contain at least one spanning tree (connected graphs), the Laplacian matrix has $\text{rank } L = N - 1$ and exactly 1 zero-eigenvalue, with all other eigenvalues in the right-half complex plane [4]. Additionally, if the communication graph is undirected, L is symmetric and all eigenvalues of L are real.

To illustrate, we will now construct the Laplacian matrices (and explore the corresponding eigenstructure) for the three example systems shown below:



We will take all nonzero weights to be equal to unity. Graph A is referred to as a *directed tree*. It is connected and has a spanning tree with root node v_1 . This graph is an example of a *leader-follower* communication topology. Graph B is a connected directed graph, and since every node is a root node (starting from any node we can reach all other nodes through a directed path) this graph is strongly connected. Graph C is undirected; furthermore, since every possible edge exists, this graph is complete or fully connected. Note that both Graphs B and C are balanced, since the in-degree of each node is equal to the out-degree ($d_i = d_i^o = 1$ for Graph B, and $d_i = d_i^o = 3$ for Graph C). The Laplacian matrices for these three systems can be constructed directly from the graph by inspection: The diagonal elements represent the in-degrees for each node and the off-diagonal elements in the i^{th} row correspond to the nodes providing information to node v_i :

$$L_A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}, \quad L_B = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \quad L_C = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}. \quad (1.29)$$

Note that all row-sums for each matrix vanish as required. Also note that since Graph C is undirected, the Laplacian matrix is symmetric.

We also give the eigenvalues for these three Laplacians in Table 1.1, which demonstrate the typical eigenstructure of directed tree, cyclic, and complete graphs. Since all graphs contain at least one spanning tree, all Laplacians have exactly one zero-eigenvalue. In directed tree (leader-follower type) graphs, the remaining eigenvalues are degenerate and equal to 1. For cyclic graphs, the nonzero eigenvalues nondegenerate and complex. Finally, for complete graphs, all nonzero eigenvalues are degenerate and equal to N , the number of agents in the system.

	μ_1	μ_2	μ_3	μ_4
Graph A	0	1	1	1
Graph B	0	$1+i$	$1-i$	2
Graph C	0	4	4	4

Table 1.1: Eigenvalues corresponding to the Laplacians in Eq. (1.29).

Throughout most of this work we will typically set all non-zero weights a_{ij} equal to unity as in the examples above; the more general case of $a_{ij} > 0$ corresponds to control systems where the controller gains are agent-dependent, as discussed in the next section.

1.2.2 General Second-Order Consensus Protocols

Let us now imagine that we have a system of N identical second-order agents $x_i = [r_i \ v_i]^T$, $i \in \mathcal{N} = 1, 2, \dots, N$, each of which evolves according to the dynamics

$$\dot{x}_i = Ax_i + Bu_i. \quad (1.30)$$

For now we will assume our agents are constrained to one dimension, so that the number of states per agent is $n = 2$; generalization to three dimensions is straightforward and will be developed subsequently. At this point we take this to be a *general* second-order LTI system, so that

$$A = \begin{bmatrix} 0 & 1 \\ f & g \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

for some general dynamics characterized by f and g .

Let us further imagine that the N agents have perfect knowledge of their own states, and can communicate this information to other agents in the system through the communication topology characterized by the Laplacian matrix L . If our goal is for the N agents to reach *consensus* in position and velocity — so that $r_i - r_j \rightarrow 0$ and $v_i - v_j \rightarrow 0$ for all i, j — then a controller which might achieve this objective is given by

$$u_i = - \sum_{j=1}^N a_{ij} \left(k_P(r_i - r_j) + k_D(v_i - v_j) \right), \quad i \in \mathcal{N}. \quad (1.31)$$

This is known as the *local voting protocol*. As implied by the choice of notation, this is a PD-type controller which feeds back the position and velocity differences to agent i . Note that

the elements of the adjacency matrix a_{ij} take into account the available information to agent i : if agent i does not receive information from agent j , then $a_{ij} = 0$ (the communication graph does not contain the edge (ν_j, ν_i)).

Due to the coupling between agents induced by the above control law, the equations of motion for the entire system is described by N coupled second-order equations of the form Eq. (1.30):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} A & 0 & \cdots & 0 \\ 0 & A & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} Bu_1 \\ Bu_2 \\ \vdots \\ Bu_N \end{bmatrix} = (I_N \otimes A) \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} Bu_1 \\ Bu_2 \\ \vdots \\ Bu_N \end{bmatrix},$$

where the operator \otimes represents the Kronecker matrix product, defined for general matrices P and Q as

$$P \otimes Q = \begin{bmatrix} p_{11}Q & \cdots & p_{1n}Q \\ \vdots & \ddots & \vdots \\ p_{m1}Q & \cdots & p_{mn}Q \end{bmatrix},$$

with p_{ij} the components of the $m \times n$ matrix P . Defining the matrix $K = [k_P \ k_D]$, we can break up the control protocol Eq. (1.31) as

$$\begin{aligned} u_i &= - \sum_{j=1}^N a_{ij} K x_i + \sum_{j=1}^N a_{ij} K x_j \\ &= -K x_i \sum_{j=1}^N a_{ij} + \sum_{j=1}^N a_{ij} K x_j \\ &= -K \left(x_i d_i - \sum_{j=1}^N a_{ij} x_j \right). \end{aligned} \tag{1.32}$$

This form suggests a connection to the Laplacian matrix, $L = D - A$. It is in fact straightforward to show that the system-wide control vector can written in state-feedback form as

$$\begin{bmatrix} Bu_1 \\ Bu_2 \\ \vdots \\ Bu_N \end{bmatrix} = (L \otimes BK) \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}.$$

If we therefore define the state vector of the entire system as $\mathbf{x} = [x_1; x_2; \dots; x_N]$, the closed-loop system-wide equations of motion under the control protocol Eq. (1.31) is given concisely as

$$\dot{\mathbf{x}} = \left((I_N \otimes A) - (L \otimes BK) \right) \mathbf{x}. \quad (1.33)$$

■ Example: consensus control for two-agent system

Let us assume both agents share information with one-another with equal (unity) weights, $a_{12} = a_{21} = 1$. The in-degree for each agent is thus $d_1 = d_2 = 1$. The Laplacian matrix is therefore

$$L = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Denoting the elements of L as ℓ_{ij} , the control vector for the entire system is computed as

$$\begin{aligned} -(L \otimes BK) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= - \begin{bmatrix} \ell_{11}BK & \ell_{12}BK \\ \ell_{21}BK & \ell_{22}BK \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = - \begin{bmatrix} BKx_1 - BKx_2 \\ -BKx_1 + BKx_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ -k_P(r_1 - r_2) - k_D(v_1 - v_2) \\ 0 \\ -k_P(r_2 - r_1) - k_D(v_2 - v_1) \end{bmatrix}, \end{aligned}$$

which is exactly the form expected from the consensus protocol Eq. (1.31) for $a_{12} = a_{21} = 1$. Note that the interplay between the coefficients d_i and a_{ij} in the protocol, as expanded in Eq. (1.32), becomes more apparent in the case of a fully-connected *three*-body system. Furthermore, the true efficiency of this notation is revealed when the number of agents grows.

■

It may be apparent that there are in fact two natural ways in which one can construct the state-vector of the entire system. We previously defined this state vector by stacking the individual states of each agent. This will be referred to as the *local* representation:

$$\mathbf{x}_L = [x_1; x_2; \dots; x_N] = [r_1 \ v_1 \ r_2 \ v_2 \ \dots \ r_N \ v_N]^T.$$

Alternatively, one can define the system-wide state vector by first forming the vector of each agent's position, $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_N]^T$, and each agent's velocity, $\mathbf{v} = [v_1 \ v_2 \ \dots \ v_N]^T$.

A complete system-wide state vector can then be constructed by stacking these two vectors vertically; this will be referred to as the *global* representation:

$$\mathbf{x}_G = [\mathbf{r}; \mathbf{v}] = [r_1 \ r_2 \ \cdots \ r_N \ v_1 \ v_2 \ \cdots v_N]^T.$$

Of course, these two state representations are related by a permutation transformation $\mathbf{x}_G = \Pi \mathbf{x}_L$, $\mathbf{x}_L = \Pi^T \mathbf{x}_G$. The permutation matrix is defined explicitly as follows: first define the projection operator $P_j^{(n)}$, $j = 1, 2, \dots, n$, which projects out the j^{th} substate from agent i . This is simply a $1 \times n$ row matrix with a 1 in the j^{th} position and zeros elsewhere. Note that for second order systems there are only two projections for position and velocity,

$$\begin{aligned} P_1^{(2)} x_i &= [1 \ 0] x_i = r_i, \\ P_2^{(2)} x_i &= [0 \ 1] x_i = v_i. \end{aligned}$$

The permutation matrix transforming \mathbf{x}_L to \mathbf{x}_G is then given by

$$\Pi = \begin{bmatrix} I_N \otimes P_1^{(n)} \\ I_N \otimes P_2^{(n)} \\ \vdots \\ I_N \otimes P_n^{(n)} \end{bmatrix}.$$

With this permutation matrix, we can now deduce the form of the closed-loop equations of motion under the consensus protocol in the global representation:

$$\begin{aligned} \dot{\mathbf{x}}_G &= \Pi \dot{\mathbf{x}}_L = \Pi((I_N \otimes A) - (L \otimes BK)) \mathbf{x}_L \\ &= \Pi((I_N \otimes A) - (L \otimes BK)) \Pi^T \Pi \mathbf{x}_L \\ &= \Pi((I_N \otimes A) - (L \otimes BK)) \Pi^T \mathbf{x}_G. \end{aligned}$$

The last line can be rewritten by careful application of the Kronecker product identities; we find

$$\dot{\mathbf{x}}_G = ((A \otimes I_N) - (BK \otimes L)) \mathbf{x}_G = A_{\text{CL}}^G \mathbf{x}_G. \quad (1.34)$$

Thus, transforming from local to global representations has the effect⁶ of changing the order of the Kronecker products in the closed-loop equations of motion Eq. (1.33). In

⁶This is not necessarily true for systems with spatial dimension $d > 1$. In these cases, this statement depends on our definition of the global representation, as will be discussed in Sec. 1.2.4.

addition, in the case of double-integrator dynamics the closed-loop state matrix in the global representation can be written neatly as a block matrix,

$$A_{\text{CL}}^G \equiv (A \otimes I_N) - (BK \otimes L) = \begin{bmatrix} \mathbf{0} & I_N \\ -k_P L & -k_D L \end{bmatrix}, \quad (1.35)$$

where $\mathbf{0}$ represents the $N \times N$ matrix of zeros⁷ and I_N is the $N \times N$ identity matrix.

Before moving on, a few comments on the results of this section are in order. The consensus protocol we have introduced (the stability of which will be the topic of the next section) involved agent-independent gains k_P and k_D . It therefore may seem that this is a very restricted type of controller, and the closed-loop equations of motion would need to be modified if we varied the gains between agents. Things are not, however, as restricted as they seem; the gains in each term of the control protocol Eq. (1.31) are multiplied by weights a_{ij} , which in general are any real positive numbers. We can therefore choose different gains for each agent's control protocol by associating appropriate weights to the edges in the communication graph. In our two-agent example, for instance, if we instead chose the Laplacian to be

$$L = \begin{bmatrix} 1 & -1 \\ -0.5 & 0.5 \end{bmatrix},$$

the control protocols for each agent would be

$$\begin{aligned} u_1 &= -k_P(r_1 - r_2) - k_D(v_1 - v_2), \\ u_2 &= -0.5k_P(r_2 - r_1) - 0.5k_D(v_2 - v_1) \\ &= -k'_P(r_2 - r_1) - k'_D(v_2 - v_1). \end{aligned}$$

The gains therefore can be different within each agent's control law depending on the weights of the edges of the communication graph, a_{ij} . This in fact allows us to absorb k_P into the weights a_{ij} , so that without loss of generality the consensus control law Eq. (1.31) can be rewritten with $k_P \rightarrow 1$ and $k_D \rightarrow \gamma$. This is the form used, for instance, in [27] and other literature by W. Ren.

Although this is quite general, the ratio of gains k_P/k_D for each agent does not vary. One can introduce even more generality by defining *two* communications graphs \mathcal{G}_r and \mathcal{G}_v ,

⁷As mentioned in Sec. 1.1.1, to avoid notational clutter we will not specify the dimensions on block matrices (*i.e.*, $\mathbf{0}_{N \times N}$) when there is no ambiguity, as in this case.

with \mathcal{G}_r characterizing the position-information communication and \mathcal{G}_v characterizing the velocity-information communication. These graphs necessarily have the same node set \mathcal{V} but in general can have different edge sets, $\mathcal{G}_r = \{\mathcal{V}, \mathcal{E}_r\}$ and $\mathcal{G}_v = \{\mathcal{V}, \mathcal{E}_v\}$, such that each edge set is associated with a set of edge weights,

$$\mathcal{E}_r \leftrightarrow a_{ij}^{(r)}, \quad \mathcal{E}_v \leftrightarrow a_{ij}^{(v)}.$$

The local-voting protocol can then be written as

$$u_i = - \sum_{j=1}^N a_{ij}^{(r)} (r_i - r_j) - \sum_{j=1}^N a_{ij}^{(v)} (v_i - v_j).$$

This is the most general form of the local voting protocol one can write between N second-order agents. We will from now on, however, simply assume that the position and velocity communication topology is identical, $\mathcal{G}_r = \mathcal{G}_v$, so that the consensus control law is given by Eq. (1.31).

To summarize, we have found that the general consensus control protocol for second-order systems (the stability of which we have yet to show) gives rise to two simple forms for the closed-loop equations of motion, depending on whether we use the local or global representation. Since these relations will be central throughout this work, we collect them here for referencing convenience. The consensus control protocol is given by

$$u_i = - \sum_{j=1}^N a_{ij} \left(k_P(r_i - r_j) + k_D(v_i - v_j) \right). \quad (1.31)$$

Depending on how we define our system-wide state vector, the closed-loop equations of motion are given as follows:

Local Representation:

$$\begin{aligned} \mathbf{x}_L &\equiv [r_1 \ r_2 \ \dots \ r_N \ v_1 \ v_2 \ \dots \ v_N]^T \\ \dot{\mathbf{x}}_L &= ((I_N \otimes A) - (L \otimes BK)) \mathbf{x}_L \end{aligned} \quad (1.36)$$

Global Representation:

$$\begin{aligned} \mathbf{x}_G &\equiv [r_1 \ r_2 \ \dots \ r_N \ v_1 \ v_2 \ \dots \ v_N]^T \\ \dot{\mathbf{x}}_G &= ((A \otimes I_N) - (BK \otimes L)) \mathbf{x}_G \end{aligned} \quad (1.37)$$

Furthermore, for double-integrator dynamics, the closed-loop system matrix in the global representation can be written in block-matrix form:

$$\dot{\mathbf{x}}_G = \begin{bmatrix} \mathbf{0} & I_N \\ -k_P L & -k_D L \end{bmatrix} \mathbf{x}_G \quad (1.38)$$

1.2.3 Stability of the Consensus Protocol

In the previous section we introduced, somewhat relying on intuition, the consensus control protocol Eq. (1.31). We will now prove that this control protocol indeed achieves consensus for a general second-order N -agent system, regardless of initial conditions. As already introduced above Eq. (1.31), we define consensus formally as follows:

Definition 1 (Consensus). Given a multiagent system, where agent i is described by the state $\mathbf{x}_i(t) \in \mathbb{R}^n$, the system is said to *reach* or *achieve consensus* if

$$\mathbf{x}_i(t) - \mathbf{x}_j(t) \rightarrow \mathbf{0}, \quad \text{as } t \rightarrow \infty$$

for all $i, j \in \mathcal{N} = 1, 2, \dots, N$. Here N is the number of agents in the system.

For second-order systems in one spatial dimension, $n = 2$ in the above definition, and the components of the state \mathbf{x}_i correspond to position and velocity of the agent. For d spatial dimensions, a second-order system has $n = 2d$ components per state.

If a multiagent system under a consensus control protocol reaches consensus *regardless of initial conditions*, we say that the consensus control protocol *stabilizes the system*. More formally, the control law *stabilizes the consensus subspace*, where the consensus subspace is defined as the system-wide state such that $\mathbf{x}_i = \mathbf{x}_j, \forall i, j \in \mathcal{N}$. We will also sometimes refer to the control protocol itself as stable or unstable, depending on whether it stabilizes the system. Additionally, if a controlled system reaches consensus regardless of initial conditions, we simply say that the *controlled system is stable*.

Illustrative prelude: first-order consensus.

Before discussing the stability of these consensus control protocols, however, it will be instructive to consider consensus of *first*-order multi-agent systems; the dynamics of each agent in such a system is given by

$$\dot{r}_i = u_i, \quad i = 1, 2, \dots, N.$$

The consensus control protocol for this system is now given by

$$u_i = - \sum_j^N a_{ij}(r_i - r_j) \quad (1.39)$$

(where we have absorbed the control gains into the weights a_{ij}), which gives rise to the system-wide closed-loop equations of motion

$$\dot{\mathbf{r}} = -L\mathbf{r}, \quad (1.40)$$

where $\mathbf{r} = [r_1, r_2, \dots, r_N]^T$. The equilibrium of this system is reached when the left-hand side vanishes; we denote this equilibrium state as \mathbf{r}_{ss} :

$$\mathbf{0} = -L\mathbf{r}_{ss}.$$

Now, as discussed in Sec. 1.2.1, the fact that the rows of L individually sum to zero implies that $\text{Rank } L \leq N - 1$, and therefore L has at least one zero-eigenvalue. This also implies that the vector $c\mathbf{1}_N$, with $\mathbf{1}_N$ the N -dimensional column vector of ones and $c \in \mathbb{R}$, is in the null space of $-L$:

$$-L(c\mathbf{1}) = 0.$$

We therefore immediately see that the system reaches equilibrium when $\mathbf{r}_{ss} = c\mathbf{1}$, or

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} = \begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix}$$

for any constant c ; that is, equilibrium is reached when the states of all N agents reach *consensus*. Moreover, when the communication topology contains a spanning tree (so that

there exists a directed communication path from at least one agent to all other agents; see Sec. 1.2.1), then according to Theorem 1 the Laplacian L contains exactly one zero-eigenvalue and therefore $c\mathbf{1}$ is the only vector in the null space.

To determine the conditions under which the first-order consensus protocol (1.39) drives the system to consensus, we express the solution to the equation of motion Eq. (1.40) in terms of the initial state (at time $t_0 = 0$) and the state transition matrix $\Phi(t, 0)$:

$$\mathbf{r}(t) = \Phi(t, 0)\mathbf{r}(0) = e^{-Lt}\mathbf{r}(0).$$

Expressed in modal form (see Sec. 1.1.1), the STM is given by

$$e^{-Lt} = Ve^{-Jt}W,$$

where J is the Jordan matrix associated with L , and V and W are the matrices built from the right- and left-eigenvectors of L , respectively, normalized such that $VW = I_N$. The matrix exponential on the right-hand side is built from matrix exponentials of Jordan blocks along the diagonal,

$$e^{-Jt} = \begin{bmatrix} e^{-J_1 t} & & & \\ & e^{-J_2 t} & & \\ & & \ddots & \\ & & & e^{-J_m t} \end{bmatrix}.$$

Now, since the graph associated with Laplacian L contains a spanning tree, L has exactly one zero-eigenvalue ($\mu_1 = 0$) and all other eigenvalues are positive ($\mu_i > 0$ for $i = 2, 3, \dots, N$). This implies that the Jordan block associated with μ_1 is simply $J_1 = 0$, and all other Jordan blocks are associated with positive eigenvalues. Therefore, asymptotically, we have

$$\left. \begin{array}{l} e^{-J_1 t} \rightarrow 1 \\ e^{-J_i t} \rightarrow \mathbf{0}, \quad i = 2, 3, \dots, m \end{array} \right\} \quad \text{as } t \rightarrow \infty, \quad (1.41)$$

so that $Ve^{-Jt}W \rightarrow \mathbf{v}_1\mathbf{w}_1^T$ as $t \rightarrow \infty$, with \mathbf{v}_1 and \mathbf{w}_1 the right- and left-eigenvectors associated with $\lambda_1 = 0$ and normalized such that $\mathbf{w}_1^T\mathbf{v}_1 = 1$. In other words, the STM approaches $\mathbf{v}_1\mathbf{w}_1^T$ asymptotically, and therefore

$$\mathbf{r}(t) = \mathbf{v}_1\mathbf{w}_1^T\mathbf{r}(0), \quad \text{as } t \rightarrow \infty.$$

Recalling that $\mathbf{v}_1 = c\mathbf{1}$, and \mathbf{w}_1 is the left-eigenvector of L associated with $\mu_1 = 0$, we can without loss of generality set $c = 1$; this choice determines the normalization of the components of \mathbf{w}_1 through the normalization condition imposed by the Jordan decomposition, $\mathbf{w}_1^T \mathbf{v}_1 = 1$. We write this normalized eigenvector as $\mathbf{w}_1^T = [p_1 \ p_2 \ \cdots \ p_N]$, so that setting $c = 1$ forces

$$\mathbf{w}_1^T \mathbf{1} = \sum_i^N p_i = 1. \quad (1.42)$$

The system-wide state thus asymptotically approaches

$$\mathbf{r}(t) \rightarrow \left(\sum_i^N p_i r_i(0) \right) \mathbf{1}$$

or, equivalently,

$$r_j(t) \rightarrow \sum_i^N p_i r_i(0), \quad \text{for all } j.$$

We therefore see that the sum $\sum_i^N p_i r_i(0)$, which is a *weighted average* of the initial positions, represents the consensus value. The weighting factors p_i are determined directly from the communication topology encoded in L and the normalization condition Eq. (1.42). It is interesting to note that the quantity $\mathbf{w}_1^T \mathbf{r}(t)$, for any time t , is in fact a constant of motion:

$$\frac{d}{dt} (\mathbf{w}_1^T \mathbf{r}(t)) = \mathbf{w}_1^T \dot{\mathbf{r}}(t) = \mathbf{w}_1^T (-L\mathbf{r}(t)) = -(\mathbf{w}_1^T L)\mathbf{r}(t) = 0.$$

Second-order consensus

We have just shown that the first-order consensus control protocol Eq. (1.39) brings an N -body system to consensus asymptotically as long as the communication graph contains a spanning tree. We will now return to second-order multivehicle systems under the consensus control protocols introduced in Sec. 1.2.2. The proof that these systems are stable will follow very similarly to the proof for first-order systems. The essentials of the following argument were given in [6]; these results are reproduced here since the stability proof we construct in Chapter 2 for fractional consensus controllers follows a similar logic.

Let us assume that an N -body system has an undirected communication graph which

contains a spanning tree, so that according to Theorem 1

$$\text{eig } L = \begin{cases} \mu_1 &= 0 \\ \mu_i &> 0, \quad i = 2, 3, \dots, N \end{cases} \quad (1.43)$$

We write our closed-loop equations of motion under the consensus control protocol Eq. (1.31),

$$u_i = - \sum_{j=1}^N a_{ij} \left(k_P(r_i - r_j) + k_D(v_i - v_j) \right), \quad i \in \mathcal{N}, \quad (1.31)$$

in the global-representation,

$$\dot{\mathbf{x}}_G = \begin{bmatrix} \mathbf{0} & I_N \\ -k_P L & -k_D L \end{bmatrix} \mathbf{x}_G = \Gamma \mathbf{x}_G, \quad (1.44)$$

where for notational clarity we have relabeled the closed-loop system matrix in the global representation $A_{CL}^G \rightarrow \Gamma$.

Guided by the analysis for first-order systems above, we will first determine the eigenstructure of Γ , which will allow us to find the Jordan decomposition and asymptotic STM. The eigenvalues λ_i of Γ are found as the roots of the characteristic polynomial,

$$\begin{aligned} 0 &= \det(\lambda I_{2N} - \Gamma) = \det \begin{bmatrix} \lambda I_N & -I_N \\ k_P L & \lambda I_N + k_D L \end{bmatrix} \\ &= \det(\lambda^2 I_N + \lambda k_D L + k_P L), \end{aligned} \quad (1.45)$$

where in the second line we have used a general determinant property for block matrices,

$$\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(A D - C B).$$

Now, utilizing the determinant identity,

$$\det(a I_N + B) = \prod_{i=1}^N (a + b_i)$$

(with b_i the eigenvalues of B), we can write Eq. (1.45) as

$$\det \left((\lambda^2) I_N + (k_P + \lambda k_D) L \right) = \prod_i \left(\lambda^2 + (k_P + \lambda k_D) \mu_i \right)$$

where, again, μ_i are the eigenvalues of L . Solutions to the characteristic equation Eq. (1.45), which correspond to the eigenvalues of the closed-loop system matrix Γ , are now determined from the individual terms in the product on the right-hand side,

$$\lambda^2 + (k_P + \lambda k_D)\mu_i = 0, \quad \text{for } i = 1, 2, \dots, N. \quad (1.46)$$

Being second-order in λ , we therefore see that each eigenvalue μ_i corresponds to two eigenvalues of Γ :

$$\lambda_{i\pm} = \frac{-k_D\mu_i \pm \sqrt{k_D^2\mu_i^2 - 4\mu_i k_P}}{2}. \quad (1.47)$$

Since we are assuming that the undirected communication diagram contains a spanning tree, and thus the eigenvalues of L are given by Eq. (1.43), and furthermore since we are assuming that $k_P, k_D > 0$, the eigenvalues of Γ satisfy the following:

$$\text{eig } \Gamma = \begin{cases} \lambda_{1\pm} & = 0 \\ \text{Re } \lambda_{i\pm} & < 0, \quad i = 2, 3, \dots, N. \end{cases} \quad (1.48)$$

That is, Γ has *exactly* two zero-eigenvalues ($\lambda_{1\pm}$), and all other $2N - 2$ eigenvalues are in the negative left-half plane.

The eigenvalues of Γ are associated with Jordan blocks J_1, J_2, \dots, J_m , whose number and dimension depend on the algebraic and geometric multiplicities of the eigenvalues $\lambda_{i\pm}$; note that the indices on the Jordan blocks are in general distinct from the indices on the eigenvalue pairs. We know that the zero-eigenvalue has algebraic multiplicity two ($\lambda_{i+} = 0$ and $\lambda_{i-} = 0$). We can show that these eigenvalues have a geometric multiplicity of one as follows: Let us write an eigenvector associated with a zero-eigenvalue of Γ as $\mathbf{v} = [\mathbf{q}_a; \mathbf{q}_b]$, where $\{\mathbf{q}_a, \mathbf{q}_b\} \in \mathbb{R}^N$, so that

$$\Gamma \mathbf{v} = 0\mathbf{v} = \mathbf{0}_{2N} \quad \Rightarrow \quad \begin{bmatrix} \mathbf{0} & I_N \\ -k_P L & -k_D L \end{bmatrix} \begin{bmatrix} \mathbf{q}_a \\ \mathbf{q}_b \end{bmatrix} = \begin{bmatrix} \mathbf{0}_N \\ \mathbf{0}_N \end{bmatrix}.$$

This implies that $\mathbf{q}_b = \mathbf{0}_N$ and $L\mathbf{q}_a = \mathbf{0}_N$, and therefore that \mathbf{q}_a is in the null space of L . Since L has exactly one zero-eigenvalue and all others in the right-half complex plane, \mathbf{q}_a is the only linearly-independent eigenvector in the null space of L . This further implies that $\mathbf{v} = [\mathbf{q}_a; \mathbf{0}_N]$ is the only linearly-independent eigenvector of Γ associated with a zero-eigenvalue, and thus the zero-eigenvalues $\lambda_{i\pm}$ of Γ have geometric multiplicity of one.

We will associate the eigenvector $\mathbf{v} = [\mathbf{q}_a; \mathbf{0}_N]$ with the first zero-eigenvalue λ_{i+} , and since we have seen that \mathbf{q}_a is in the null-space of L , we can without loss of generality set $\mathbf{q}_a = \mathbf{1}_N$. We therefore have $\mathbf{v}_1 = [\mathbf{1}_N; \mathbf{0}_N]$. Since the zero-eigenvalue of Γ has geometric multiplicity of one, the eigenvector \mathbf{v}_2 associated with $\lambda_{i-} = 0$ must be a generalized eigenvector of rank 2:

$$\Gamma^2 \mathbf{v}_2 = \mathbf{0}_{2N} \quad \Rightarrow \quad \Gamma \mathbf{v}_2 = \mathbf{v}_1.$$

This vector is given by $\mathbf{v}_2 = [\mathbf{0}_N; \mathbf{1}_N]$, since

$$\Gamma \mathbf{v}_2 = \begin{bmatrix} \mathbf{0} & I_N \\ -k_P L & -k_D L \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{1}_N \end{bmatrix} = \begin{bmatrix} \mathbf{1}_N \\ -k_D L \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_N \\ \mathbf{0}_N \end{bmatrix}.$$

Similarly, we can show that the left-eigenvectors ($\mathbf{w}_i^T \Gamma = \lambda_i \mathbf{w}_i^T$) associated with $\lambda_{1\pm} = 0$ are given by $\mathbf{w}_1 = [\mathbf{p}; \mathbf{0}_N]$ and $\mathbf{w}_2 = [\mathbf{0}_N; \mathbf{p}]$, where \mathbf{w}_2 is the generalized left-eigenvector of \mathbf{w}_1 . The remaining right- and left-eigenvectors will be denoted \mathbf{v}_i and \mathbf{w}_j for $i, j = 3, 4, \dots, 2N$.

The Jordan decomposition of Γ now takes the form

$$\Gamma = V J W = V \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_m \end{bmatrix} W = V \begin{bmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_m \end{bmatrix} W, \quad (1.49)$$

with the columns of V containing the right-eigenvectors \mathbf{v}_i ,

$$V = \begin{bmatrix} | & | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \cdots & \mathbf{v}_{2N} \\ | & | & | & & | \end{bmatrix} = \begin{bmatrix} \mathbf{1}_N & \mathbf{0}_N & | & & | \\ & \mathbf{v}_3 & \cdots & \mathbf{v}_{2N} \\ \mathbf{0}_N & \mathbf{1}_N & | & & | \end{bmatrix}, \quad (1.50)$$

and the rows of W containing the left-eigenvectors \mathbf{w}_i^T ,

$$W = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \mathbf{w}_3^T \\ \vdots \\ \mathbf{w}_{2N}^T \end{bmatrix} = \begin{bmatrix} \mathbf{p}^T & \mathbf{0}_N^T \\ \mathbf{0}_N^T & \mathbf{p}^T \\ \mathbf{w}_3^T \\ \vdots \\ \mathbf{w}_{2N}^T \end{bmatrix}. \quad (1.51)$$

From Eq. (1.49) we can construct the STM of the system; setting $t_0 = 0$, we find

$$\Phi(t, 0) = e^{\Gamma t} = V \begin{bmatrix} e^{J_1 t} & & & \\ & e^{J_2 t} & & \\ & & \ddots & \\ & & & e^{J_m t} \end{bmatrix} W.$$

Since the eigenvalues associated with the Jordan blocks J_i , $i = 2, 3, \dots, m$, are negative, all blocks except for $\exp(J_1 t)$ vanish as $t \rightarrow \infty$ (as in Eq. (1.41)). For J_1 , we find and

$$e^{J_1 t} = \exp \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} t \right) = \sum_{k=0}^{\infty} \frac{J_1^k t^k}{k!} = I_2 + J_1 t = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix},$$

where the series terminates after $k = 1$ since J_1 is nilpotent ($J_1^k = 0$ for $k \geq 2$). The STM therefore asymptotically approaches

$$\Phi(t, 0) \rightarrow V \begin{bmatrix} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} W.$$

Utilizing the explicit form of the V and W matrices in Eqs. (1.50) and (1.51), we see that the STM asymptotically approaches

$$\Phi(t, 0) \rightarrow \begin{bmatrix} \mathbf{1}_N \mathbf{p}^T & \mathbf{1}_N \mathbf{p}^T t \\ \mathbf{0}_{N \times N} & \mathbf{1}_N \mathbf{p}^T \end{bmatrix}. \quad (1.52)$$

We therefore find that, for any $\mathbf{x}(0) = [\mathbf{r}(0); \mathbf{v}(0)]$, the system-wide state asymptotically approaches

$$\begin{aligned} \mathbf{r} &\rightarrow \mathbf{1}_N \left((\mathbf{p}^T \mathbf{r}(0)) + (\mathbf{p}^T \mathbf{v}(0))t \right) \\ \mathbf{v} &\rightarrow \mathbf{1}_N (\mathbf{p}^T \mathbf{v}(0)). \end{aligned}$$

Defining $\mathbf{p}^T \mathbf{r}(0) \equiv \bar{r}$ and $\mathbf{p}^T \mathbf{v}(0) \equiv \bar{v}$ as the weighted averages of the initial states $\mathbf{r}(0)$ and $\mathbf{v}(0)$, the state asymptotically approaches

$$\begin{aligned} \mathbf{r} &\rightarrow \mathbf{1}_N (\bar{r} + \bar{v}t) \\ \mathbf{v} &\rightarrow \mathbf{1}_N \bar{v}, \end{aligned} \quad (1.53)$$

or $r_i \rightarrow \bar{r} + \bar{v}t$ and $v_i \rightarrow \bar{v}$ for all N agents ($\forall i = 1, 2, \dots, N$).

The above argument proves that if Γ has exactly two zero-eigenvalues with all other eigenvalues in the left-half plane, the system achieves consensus. This is a sufficient condition; the necessary condition is proved easily by contradiction [6]. We have therefore proved the following:

Theorem 2. (Lemma 4.1 of [6]) A second-order multiagent system under consensus control protocol Eq. (1.31) is stable if and only if the closed-loop system matrix

$$\Gamma = (A \otimes I_N - BK \otimes L) = \begin{bmatrix} \mathbf{0} & I_N \\ -k_P L & -k_D L \end{bmatrix}$$

has exactly two zero-eigenvalues and all other eigenvalues in the left-half plane. Here, the matrices A and B are the second-order system and input matrices defined in Eq. (1.9), $K = [k_P \ k_D]$, and L is the Laplacian matrix.

We have also shown as a corollary that if the graph \mathcal{G} is undirected and connected (contains a spanning tree), then the consensus control protocol stabilizes the system for all $k_P, k_D > 0$. For directed graphs which contain a spanning tree (as is the case in leader-follower communication topologies), the closed-loop system will be stable if

$$k_D > \max_i \sqrt{\frac{k_P}{a_i} \frac{b_i^2}{a_i^2 + b_i^2}}, \quad i = 2, 3, \dots, N,$$

where a_i and b_i are the real and imaginary parts of the eigenvalues of L , respectively. This result can be found by tracing the stability region $\lambda_i = i\rho$ in the characteristic equation Eq. (1.46). A similar result will be derived explicitly in Chapter 2.

1.2.4 Generalization to Three Dimensions

Generalization of the above cooperative control formalism to three dimensions is fairly straightforward, but there are a few important notational and definitional issues which arise which will be mentioned here. Let us represent the state of the i^{th} agent as $\mathbf{x}_i = [\mathbf{r}_i^T \ \mathbf{v}_i^T]^T \equiv [\mathbf{r}_i; \mathbf{v}_i]$, where $\mathbf{r}_i = [r_{xi} \ r_{yi} \ r_{zi}]^T$ and $\mathbf{v}_i = [v_{xi} \ v_{yi} \ v_{zi}]^T$ are the three

dimensional position and velocity vectors, respectively. A general local-voting protocol in 3 dimensions is given by

$$\mathbf{u}_i = - \sum_j^N a_{ij} K(\mathbf{x}_i - \mathbf{x}_j) = \sum_j^N a_{ij} \left(K_P(\mathbf{r}_j - \mathbf{r}_i) + K_D(\mathbf{v}_j - \mathbf{v}_i) \right), \quad (1.54)$$

where $K = [K_P \ K_D]$, and K_P and K_D are, in general, non-diagonal (3×3) matrices. As discussed in Sec. 1.1.5, the off-diagonal elements induce coupling between the control signal in the i^{th} direction with position and velocity differences in the j^{th} direction.

The closed-loop equations can now be written as a matrix equation in terms of the graph Laplacian L as in the 1-dimensional case. Recall, however, that we have formulated the consensus control law using two different N -body system-state representations. In three dimensions, there are now *three* natural ways to form the system state-vector. As in the 1D case, we can form the *local* state-vector by stacking the states of the individual bodies:

$$\mathbf{x}_L = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_N].$$

The closed-loop equations of motion under the protocol (1.54) can be written as

$$\dot{\mathbf{x}}_L = \left((I_N \otimes A_{3D}) - (L \otimes B_{3D}K) \right) \mathbf{x}_L, \quad (1.55)$$

where A_{3D} and B_{3D} are the three dimensional generalizations of the second-order state and input matrices, as introduced in Sec. 1.1.5,

$$A_{3D} = \begin{bmatrix} \mathbf{0} & I_3 \\ \mathbf{f} & \mathbf{g} \end{bmatrix}, \quad B_{3D} = \begin{bmatrix} \mathbf{0} \\ I_3 \end{bmatrix}.$$

In the local representation, then, the closed-loop dynamics have the same form in one and three dimensions.

In the 3-dimensional case, we can define *two* different global representations, the first of which keeps the 3-dimensional components of each agent intact:

$$\mathbf{x}_{G_1} = [\mathbf{r}_1; \mathbf{r}_2; \dots; \mathbf{r}_N; \mathbf{v}_1; \mathbf{v}_2; \dots; \mathbf{v}_N]^T.$$

In this representation the closed-loop equations for the 3D double integrator are given by

$$\dot{\mathbf{x}}_{G_1} = \begin{bmatrix} \mathbf{0} & I_N \\ -L \otimes K_P & -L \otimes K_D \end{bmatrix} \mathbf{x}_{G_1}.$$

A second global representation can be built by treating all components of each agent's state independently; we can form the system-wide component vectors

$$\begin{aligned}\mathbf{r}_x &= [r_{1x}, r_{2x}, \dots, r_{Nx}]^T, \\ \mathbf{v}_x &= [v_{1x}, v_{2x}, \dots, v_{Nx}]^T,\end{aligned}$$

and similarly for the y - and z - components, then build the second global system-state vector as

$$\mathbf{x}_{G_2} = [\mathbf{r}_x; \mathbf{r}_y; \mathbf{r}_z; \mathbf{v}_x; \mathbf{v}_y; \mathbf{v}_z].$$

In the case of 3D double-integrator dynamics (so that the natural dynamics in each spatial component decouple), the closed-loop equations of motion are given by

$$\dot{\mathbf{x}}_{G_2} = ((A \otimes I_N) - (BK \otimes L))\mathbf{x}_{G_2},$$

where A and B are the (1D) double-integrator system and input matrices. Comparing to Eq. (1.37) and (1.38), we see that the first and second global representations become identical when the spatial dimension is taken to $d = 1$.

1.3 FRACTIONAL CALCULUS AND CONTROL

One of the central themes of the research presented in this thesis is the generalization of standard control protocols through the use of fractional calculus. In one sense, then, the generalization of certain controllers discussed herein essentially emerges from the generalization of differential and integral calculus itself, the latter of which can be conceptually expressed as

$$\left(\frac{d}{dx} \right)^{\text{integer}} \rightarrow \left(\frac{d}{dx} \right)^{\text{real or complex}}$$

The construct at right represents a derivative or integral of non-integer order. The fundamental objective of fractional calculus is therefore to assign meaning to, and determine the formal structure of, these so-called *fractional derivatives*.

Given the central role these objects play in this work, it behooves me to provide within this section a reasonably self-sufficient introduction to the topic of fractional calculus.

This discussion, however, will remain grounded to the ultimate goal of utilizing fractional derivatives in control protocols for second order systems. We will therefore restrict our attention to real-ordered derivatives and integrals. For more complete introductions to this rich subject, we point the reader to [9, 11, 28], the latter two of which are often considered canon.

1.3.1 Motivation

One of the most fundamental things a young physicist or engineer begins to learn is how to understand and manipulate constructs of the form

$$\int_0^t f(\tau) d\tau \quad \text{and} \quad \frac{df(t)}{dt},$$

and furthermore to understand the connections between chains of these operators,

$$\underbrace{\frac{d}{dt} \cdots \frac{d}{dt} \frac{d}{dt}}_{n \text{ times}} f(t) = \frac{d^n f(t)}{dt^n} \equiv D^n f(t), \quad \underbrace{\int d\tau'' \cdots \int d\tau' \int d\tau}_{n \text{ times}} f(\tau) \equiv D^{-n} f(\tau).$$

The notation on the left-hand sides of the above expressions suggest that n must be a member of the exclusive club \mathbb{Z} , the integers. In fact from this notation the expression becomes nonsensical for $n \notin \mathbb{Z}$. We will now show, however, that this notation is too restrictive; indeed, objects such as $D^{\pm\alpha} f(t)$, $\alpha \notin \mathbb{Z}$ have perfectly reasonable definitions.

As an instructive example, let us consider a power function of the form $f(x) = x^m$, with m a positive integer. In this case, acting on $f(x)$ with a chain of derivative operators gives

$$\begin{aligned} f(x) &= x^m \\ Df(x) &= mx^{m-1} \\ D^2 f(x) &= m(m-1)x^{m-2} \\ &\vdots \\ D^n f(x) &= m(m-1) \cdots (m-n+1)x^{m-n} \\ &= \frac{m!}{(m-n)!} x^{m-n}, \end{aligned}$$

where $n \leq m$. We now realize that this can be written in terms of the generalized factorial function (Euler's *gamma function*, $\Gamma(k) = (k - 1)!$) as

$$D^n x^m = \frac{\Gamma(m + 1)}{\Gamma(m - n + 1)} x^{m-n}. \quad (1.56)$$

As the gamma function⁸ can be defined for all complex numbers with positive real part as

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx,$$

and further extended to all complex numbers excluding the non-positive integers through analytic continuation, we realize that the derivative of the power function can now formally be defined for non-integer order, both real and complex:

$$D^\alpha x^m = \frac{\Gamma(m + 1)}{\Gamma(m - \alpha + 1)} x^{m-\alpha}, \quad \alpha \in \mathbb{C}', \quad m \in \mathbb{Z}^+. \quad (1.57)$$

Here \mathbb{C}' corresponds to the subset of \mathbb{C} such that the argument of the gamma function in Eq. (1.57) exclude non-positive integers (negative integers and zero); these cases, however, correspond to the vanishing standard integer-order derivatives $D^n x^m$, $n > m$, which is also suggested by the poles in the Gamma function at these points. This definition was first published by Lacroix in 1819, though it was only mentioned in passing as a formal curiosity [28].

Note that when α is a negative integer, Lacroix's definition corresponds to standard integration from 0 to x :

$$\begin{aligned} D^{-1} x^m &= \frac{1}{m + 1} x^{m+1} &= \int_0^x \xi^m d\xi \\ D^{-2} x^m &= \frac{1}{(m + 1)(p + 2)} x^{m+2} &= \int_0^x d\xi \int_0^\xi \xi'^m d\xi' \\ &\vdots && \vdots \end{aligned}$$

so that Eq. (1.57) generalizes both differentiation and integration. Thus, through simple substitution of Euler's clever invention $\Gamma(z)$, we have now assigned formal meaning to expressions such as

$$\frac{d^{1/3}}{dx^{1/3}} x^2 \quad \text{or} \quad \int_0^x \xi^3 d\xi^{1/137}.$$

⁸The gamma function will always be written with its argument (*i.e.*, $\Gamma(z)$), so no confusion between this function and the closed-loop consensus system matrix introduced in Eq. (1.44) should arise

We have therefore defined through Eq. (1.57) the *fractional derivative* of a power function. We mention here that this term is somewhat of a misnomer, as the derivative order does not necessarily have to be “fractional” (within the set \mathbb{Q} , the rational numbers).

For illustration we plot in Fig. 1.4 the Lacroix derivative $D^\alpha x$ for various choices of α . Also shown (right panel) is the gamma function $\Gamma(x)$ for $x \in \mathbb{R}$. We see from the left panel that the fractional derivative D^α , $\alpha \in [0, 1]$, essentially “interpolates” between the integer-order derivatives D^0 (the identity) and D^1 .

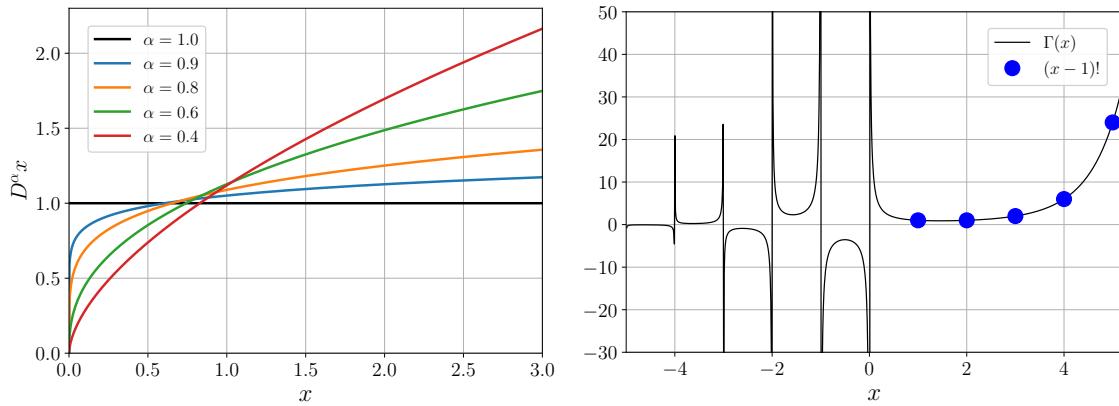


Figure 1.4: Fractional derivative $D^\alpha x$ (left), as defined in Eq. (1.57), for various fractional orders α . Included for reference is the standard first-order derivative, $D^1 x = 1$. This fractional derivative is defined using the Euler gamma function (right).

Definition (1.57) of course only applies to functions of the form $f(x) = x^m$, including functions defined through a power series. This definition was only introduced here as an illustrative example to motivate the more formal definitions introduced in what follows. Before moving on to more general definitions of the fractional derivative operator, however, we point out that definition (1.57) satisfies certain familiar properties from integer-order calculus, such as $D^\alpha D^\beta f(x) = D^{\alpha+\beta} f(x)$. For instance,

$$D^{0.3} D^{1.7} x^3 = D^{0.3} (5.1427 x^{1.3}) = 5.1427(1.1667 x) = 6x = D^2 x^3. \quad (1.58)$$

We also see that $D^\alpha D^\beta f(x) = D^\beta D^\alpha f(x)$. Though these properties are satisfied here, we must be very careful to not naively apply the intuition one has developed from standard calculus; these types of composition and commutation properties may not hold in general. I

will make an effort throughout this work to warn the reader when these issues may appear and how these identities need to be modified.

1.3.2 Formal Definitions

To introduce more general versions of fractional operators, we will first introduce a common definition of a fractional integral $D^{-\nu} \equiv {}_0I_t^\nu$, $\nu > 0$. Through careful use of the law of exponents $D^n D^{-\nu} \rightarrow D^{n-\nu}$ where the integer $n > \nu$, we can arrive at more general definitions of fractional derivatives.

The Riemann-Liouville fractional integral

We define the Riemann-Liouville (RL) fractional integral as

$${}_0I_t^\nu f(t) = \frac{1}{\Gamma(\nu)} \int_0^t (t-\tau)^{\nu-1} f(\tau) d\tau, \quad \text{Re } \nu > 0. \quad (1.59)$$

Note that, unlike Lacroix's construct, this integral is defined for a general function $f(t)$ as long as the integral converges. A sufficient condition for this integral to converge is that

$$f\left(\frac{1}{t}\right) = \mathcal{O}(t^{1-\epsilon}), \quad \epsilon > 0.$$

We say that functions satisfying this condition belong to the Riemann class. Included in this class are constants and power functions t^p , $p > -1$.

Let us consider the specific case of a power function. The RL integral diverges for $p \leq -1$; for $p > -1$ we find

$${}_0I_t^\nu t^p = \frac{1}{\Gamma(\nu)} \int_0^t (t-\tau)^{\nu-1} \tau^p d\tau = \frac{\Gamma(p+1)}{\Gamma(p+\nu+1)} t^{p+\nu}. \quad (1.60)$$

Testing this result with $\nu = 1$ and $p = 2$, we find

$${}_0I_t^1 t^2 = \frac{\Gamma(3)}{\Gamma(4)} t^3 = \frac{\Gamma(3)}{3\Gamma(3)} t^3 = \frac{t^3}{3}, \quad (1.61)$$

agreeing with the expected result from integer-order calculus. Note that we have used a central property of the Gamma function, $\Gamma(z+1) = z\Gamma(z)$. This property is of course

satisfied by the factorial — $k! = k(k-1)!$ — which the Gamma function formally generalizes as $\Gamma(k+1) = k!$. Clearly, Eq. (1.61) will hold if we take the exponent to be any real number $p \geq 0$,

$${}_0I_t^1 t^p = \frac{t^{p+1}}{p+1} = \int_0^t \tau^p d\tau.$$

We therefore see from Eq. (1.60) that the RL integral reproduces the Lacroix definition for $D^{-\nu} t^m$ in Eq. (1.57), but generalizes this definition from integer-powers to any non-integer powers, $m \in \mathbb{Z}^\mathbb{P} \rightarrow p \in \mathbb{R}^+$.

I have chosen not to derive the RL integral or to stipulate in detail the requirements on the function f here, as this can be found in various sources such as [11, 28, 29]. Moreover, it most likely would take us too far off our desired course. We have shown, however, that this object provides valid and expected results when applied to the power function. This is perhaps an appropriate amount of justification for Eq. (1.59): our main goal will be to apply fractional differentiation to trajectories which in most cases can be expanded as a power series in t . (In other words, physical trajectories are inherently “smooth” and can be described by continuous, well-behaved functions.) To quote my undergraduate thesis adviser, Nick Wheeler, from a 1997 seminar on fractional calculus,⁹

“This is work which I am happy to leave to the mathematicians; as a physicist, I know myself to be protected from major faux pas by the well-constructedness of Nature.”

Moreover, as will be discussed in Sec. 1.3.4, the specific numerical propagation method we will be using for fractionally controlled systems expands the trajectory as a polynomial in t . Thus as long as the reader is convinced of the validity of our definitions in the case of power functions, a reasonable amount of confidence in subsequent results should follow.

⁹Taken from [9]. Within this quote, Wheeler cites our reference [11] for details on the formal validity of the fractional derivative operators.

An approach to fractional derivatives

Since Eq. (1.59) generalizes integration to any order ν such that $\text{Re } \nu > 0$, one may be tempted by the operator notation and the familiar chain of relationships

$$\cdots \iint d\tau' d\tau \rightarrow \int d\tau \rightarrow I \rightarrow \frac{d}{dt} \rightarrow \frac{d^2}{dt^2} \rightarrow \cdots$$

to treat the order α in our new construct ${}_0I_t^\alpha$ as an unconstrained free variable, and therefore simply identify the fractional derivative as the object

$$\frac{d^\alpha}{dt^\alpha} \leftrightarrow {}_0I_t^{-\alpha}.$$

Unfortunately this line of reasoning fails, since Eq. 1.59 is typically divergent for $\nu < 0$. A similar line of reasoning, however, provides us with a practical result: We recognize that the chain of relationships illustrated above is essentially a statement of the fundamental theorem of calculus,

$$F(t) = \int_a^x f(\tau) d\tau \Rightarrow \frac{d}{dt} F(t) = f(t), \quad \forall t \in \mathcal{D}(f),$$

for all t in the domain of f ; in less formal terms, the action of the derivative operator D^1 serves to “peal off” integral operators from a function. We can therefore attempt to peal off a whole integral from a fractional integral to arrive at a fractional derivative, *e.g.*,

$$D^1[{}_0I_t^{0.3}f(t)] \stackrel{?}{=} D^{0.7}f(t).$$

If this operation is valid, it *generally* defines a fractional derivative in terms of the RL integral of a function: $D^\alpha f(t) \equiv D^n[{}_0I_t^{n-\alpha}f(t)]$, where n is the smallest integer greater than α . Here, n represents the number of “steps” we must take in the chain illustrated above to arrive at a particular fractional derivative from a fractional integral of order $0 < n - \alpha < 1$.

We can test the validity of such a definition by returning to the simple power function. Returning to Eq. (1.60) and taking the derivative gives

$$D^1[{}_0I_t^\nu t^p] = D^1\left[\frac{\Gamma(p+1)}{\Gamma(p+\nu+1)}t^{p+\nu}\right] = \frac{\Gamma(p+1)}{\Gamma(p+\nu+1)}(p+\nu)t^{p+\nu-1} = \frac{\Gamma(p+1)}{\Gamma(p+\nu)}t^{p+\nu-1}.$$

Making the substitution $\nu = 1 - \alpha$ (which is valid for $0 < \alpha \leq 1$ since the RL integral converges for $\nu \geq 0$) we find

$$D^1[{}_0I_t^{1-\alpha}t^p] = \frac{\Gamma(p+1)}{\Gamma(p-\alpha+1)}t^{p-\alpha}.$$

comparing this to Eq. (1.57), we see that this exactly matches the fractional derivative of t^p found by Lacroix by simple substitution of Euler's generalization of the factorial, $\Gamma(z)$, but *is now valid for all $\alpha \in \mathbb{R}$* . For higher order fractional derivatives, we can simply take higher (integer) order derivatives and repeat the above manipulations. For instance, to express the derivative $D^\alpha t^p$ for $1 < \alpha \leq 2$ we take two derivatives:

$$D^2 [{}_0 I_t^{2-\alpha} t^p] = \frac{\Gamma(p+1)}{\Gamma(p-\alpha+1)} t^{p-\alpha}.$$

This again is identical to Lacroix's result, which is expected since this latter result is formally valid for all α excluding the negative integers.

Therefore, for general $\alpha > 0$, we define the *Riemann-Liouville* fractional derivative of t^p as

$${}^{\text{RL}} D^\alpha t^p = D^n ({}_0 I_t^{n-\alpha} t^p) = \frac{\Gamma(p+1)}{\Gamma(p-\alpha+1)} t^{p-\alpha}, \quad t > 0, \alpha \in \mathbb{R}^+. \quad (1.62)$$

We notice that the domain of integration in Eq. (1.62) has been hidden by the notation ${}^{\text{RL}} D^\alpha$. This does not introduce ambiguity, since the pre-superscript "RL" indicates we are using the RL integral, defined over the integration window $[0, x]$. This notation, however, is perhaps dangerous because it hides the crucial fact that RL fractional derivatives are, unlike standard integer-order derivatives, *non-local* operators. That is, the entire "history" of $f(\tau)$ within the domain $[0, t]$ is used to compute the fractional derivative ${}^{\text{RL}} D^\alpha f(t)$. Compare this to the standard integer-order derivatives $D^n f(t)$, $n \in \mathbb{N}$, which can be computed using only the *local* region around t in the domain. Therefore, fractional differentiation is in a sense more "integration-like" than "differentiation-like". In broader terms, *differintegration* operators of order α are nonlocal *except in the specific cases $\alpha = 0, +1, +2, \dots$* . This fact leads to many subtle yet important differences between fractional- and integer-order calculus; it is therefore crucial to keep in mind that fractional derivatives are non-local and depend on the history of the function being operated upon.

The general Riemann-Liouville fractional derivative

Motivated by the above example, we define the *Riemann-Liouville fractional derivative* of a general function as

$${}^{\text{RL}}D^\alpha f(t) = D^n \left({}_0I_t^{n-\alpha} f(t) \right) = D^n \left[\frac{1}{\Gamma(n-\alpha)} \int_0^t (t-\tau)^{n-\alpha-1} f(\tau) d\tau \right], \quad (1.63)$$

where n is the smallest integer greater than $\text{Re } \alpha$ (or, for real α , $n = \lceil \alpha \rceil$ where $\lceil \bullet \rceil$ is the ceiling function). For this to be well defined, f must be piecewise continuous on $J' = (0, \infty)$ and integrable on any finite subinterval of $J = [0, \infty)$. Among the functions satisfying this property are $\ln t$ and t^p for $p > -1$.

We notice an interesting property of the RL fractional derivative if we take our function to be a constant. In standard calculus, the derivative of a constant function is zero, since this function is not *changing* with its domain argument. Using the RL definition for fractional derivative, however, we find for $0 < \alpha \leq 1$

$$\begin{aligned} {}^{\text{RL}}D^\alpha c &= D^1 \left({}_0I_t^{1-\alpha} c \right) = D^1 \left(\frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} c d\tau \right) \\ &= D^1 \left(-\frac{c}{\Gamma(1-\alpha)} \frac{t^{1-\alpha}}{\alpha-1} \right) \\ &= \frac{ct^{-\alpha}}{\Gamma(1-\alpha)}. \end{aligned}$$

We therefore see that RL derivatives of constants are not zero; in particular, we find the intriguing result,

$$D^{1/2}1 = D^{1/2}t^0 = \sqrt{\frac{1}{\pi t}}.$$

Due to the poles in $\Gamma(1-\alpha)$ at $\alpha = 1, 2, \dots$, however, this expression provides the familiar and expected result $D^m 1 = 0$ for integer m .

The general Caputo fractional derivative

An alternative definition, which turns out to be more practical for physical application, is the *Caputo fractional derivative*,

$${}^C D^\alpha f(t) \equiv {}_0I_t^{n-\alpha} [D^n f(t)] = \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-\tau)^{n-\alpha-1} [D^n f(\tau)] d\tau, \quad n = \lceil \alpha \rceil. \quad (1.64)$$

Comparing to Eq. (1.63), we see that this definition simply reverses the order of the fractional integral and integer-order derivative. From this definition we immediately see that if the function on which the Caputo derivative is acting is a constant, $f(t) = c$, the result vanishes. This is a key difference between the Caputo and RL definitions of the fractional derivative and, as we shall see, affords us a better physical grasp of systems which obey fractional differential equations of motion.

Let us compute the Caputo derivative of the power function for $0 < \alpha \leq 1$:

$$\begin{aligned} {}^C D^\alpha t^p &= \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} [D^1 \tau^p] d\tau \\ &= \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} [p\tau^{p-1}] d\tau. \end{aligned}$$

For $p < 0$ this integral does not converge, while for $p = 0$ the inner derivative vanishes; for $p > 0$, we have

$${}^C D^\alpha t^p = \frac{1}{\Gamma(1-\alpha)} \left(\frac{p\Gamma(p)\Gamma(1-\alpha)t^{p-\alpha}}{\Gamma(p-\alpha+1)} \right).$$

Again using the identity $\Gamma(z+1) = z\Gamma(z)$ and simplifying, we find

$${}^C D^\alpha t^p = \frac{\Gamma(p+1)}{\Gamma(p-\alpha+1)} t^{p-\alpha},$$

again matching the result found by Lacroix when $p \in \mathbb{Z}^+$ and the RL definition when $p > 0$. The RL derivative of the power function is therefore slightly more general since it is valid for $p > -1$. As we will see, however, the Caputo definition is more convenient for physical systems obeying fractional differential equations since to solve these systems, only the integer-order derivative initial conditions, $f(0), f'(0), \dots$, need to be specified. In what follows, then, we will always assume the Caputo derivative is being used, and simplify notation as

$${}^C D^\alpha \quad \longleftrightarrow \quad D^\alpha$$

unless otherwise specified. The notational brevity achieved by this relabeling, however, comes with the increased risk of forgetting the non-local nature of fractional derivatives. The words of warning below Eq. (1.62) should therefore be strongly heeded.

Eigenfunctions of fractional derivative operators

In what follows, we will utilize a function that in fractional calculus serves as a generalization of the exponential function. The *Mittag-Leffler* function is defined as

$$E_\alpha(z) \equiv \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(1+\alpha k)}. \quad (1.65)$$

The Mittag-Leffler (ML) function is the eigenfunction of the fractional derivative operator; that is, it solves the fractional differential equation $D^\alpha x(t) = ax(t)$ given an initial condition $x(t_0) = x_0$ as

$$x(t) = x_0 E_\alpha(at^\alpha). \quad (1.66)$$

Notice from the series definition Eq. (1.65), $E_1(x) = \exp(x)$. We show in Fig. 1.5 the solutions Eq. (1.66) for $a = -1$, $x_0 = 1$, and for various choices of α . When $\alpha = 1$ we have the typical exponential solution to the differential equation $Dx(t) = -x(t)$. As α is reduced towards zero, we notice that the solution decays at a slower rate. This “creeping” behavior is a common feature of asymptotically decaying solutions of fractional differential equations. Indeed, in the discussion of stability of fractional systems, it is common to generalize the concept of exponential stability to *Mittag-Leffler (ML) stability* (see, e.g., [30], Definition 4.1).

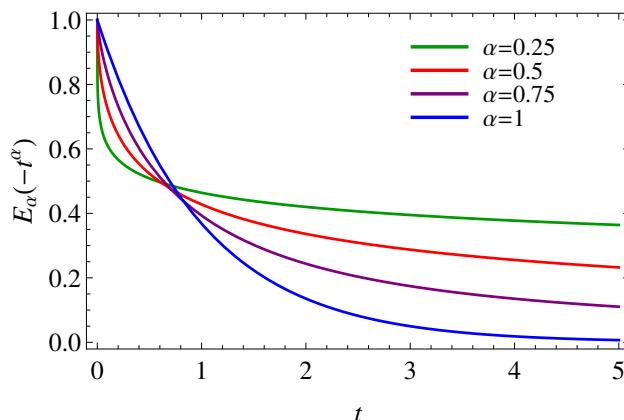


Figure 1.5: The behavior of the solutions to $D^\alpha x(t) = -x(t)$, given by $E_\alpha(-t^\alpha)$, for various choices of α . The creeping behavior of the solutions is seen as $\alpha \rightarrow 0$.

In the design of consensus controllers as discussed herein, we desire that the our multiagent systems generally reach consensus quickly; This creeping behavior is therefore an undesirable side-effect of using fractional controllers. As we will show numerically, this creeping behavior can be largely mitigated by including integral control in addition to fractional derivative control.

1.3.3 Fractional Controllers

We will now very briefly introduce how one constructs a certain class of fractional controllers, noting that the following chapters more fully develop these concepts. For dedicated introductions to fractional control systems, see [31, 32, 33].

Let us consider a double-integrator system with PD-type control of the form

$$\begin{aligned} u &= -K\mathbf{x} = -k_P r - k_D \dot{r} \\ &= -k_P r - k_D D^1 r. \end{aligned}$$

In the second line we have represented the derivative term using operator notation. In this form the generalization to fractional control becomes obvious:

$$u = -k_P r - k_D D^1 r \quad \rightarrow \quad u = -k_P r - k_D D^\alpha r.$$

This will be referred to as a *fractional* PD $^\alpha$ controller. In this case the controller cannot be written in a full-state feedback form:

$$u = -k_P r - k_D D^\alpha r = -[k_P \ k_D] \begin{bmatrix} r \\ D^\alpha r \end{bmatrix} \neq -K\mathbf{x}.$$

We therefore are unable to write the state equation in the closed-loop form $\dot{\mathbf{x}} = A_{CL}\mathbf{x}$ as before,

$$\begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (-k_P r - k_D D^\alpha r) = A\mathbf{x} - BK \begin{bmatrix} r \\ D^\alpha r \end{bmatrix} \neq (A - BK)\mathbf{x}.$$

Solving this system for the trajectory using the standard STM is therefore not an option. Instead we can write the closed-loop dynamics as a second-order *fractional differential equation* (FDE),

$$\ddot{r} + k_D D^\alpha r + k_P r = 0, \tag{1.67}$$

and, given an initial condition, solve this system numerically.¹⁰ The above equation describes what we refer to as the *fractionally damped harmonic oscillator*. There are several numerical avenues one can take; in this work we specifically use the fractional Chebyshev collocation method as will be discussed in Sec. 1.3.4. While these methods can be efficient and produce highly accurate approximations to the solutions of FDEs, they often rely on unwieldy series approximations or similar tricks which tend to hide important quantitative results (and illuminating qualitative details) of the true solutions. A more concrete analytical handle on the situation is therefore highly desirable.

Let us assume that the fractional order is rational, $\alpha \in Q$, so that it can be written as a *true* fraction $\alpha = \nu/\delta$. In this case, one can re-write the FDE in Eq. (1.67) as

$$\begin{aligned} D^2r &= -K_D D^{\nu/\delta}r - k_P r \\ \Rightarrow D^\beta D^{2-\beta}r &= -K_D D^{\nu\beta}r - k_P r, \end{aligned} \quad (1.68)$$

where in the second line we have defined $\beta = 1/\delta$.

Let us now define the *pseudostate*

$$\xi = [r; D^\beta r; D^{2\beta} r; D^{3\beta} r; \dots; D^{2-\beta} r].$$

In terms of this pseudostate, the double-integrator dynamics $\ddot{r} = 0$ can be written as a system of β^{th} -order equations:

$$D^\beta \begin{bmatrix} r \\ D^\beta r \\ D^{2\beta} r \\ \vdots \\ D^{2-\beta} r \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}}_{\tilde{A}} \begin{bmatrix} r \\ D^\beta r \\ D^{2\beta} r \\ \vdots \\ D^{2-\beta} r \end{bmatrix}. \quad (1.69)$$

Notice that the last row of this matrix equation gives the double integrator dynamics,

¹⁰Note that in special cases, FDEs can be solved analytically utilizing ML functions. See, *e.g.*, reviews [11, 29].

$D^\beta D^{2-\beta}r = D^2r = 0$. The remaining rows simply give the relational statements

$$\begin{aligned} D^\beta(r) &= D^\beta r \\ D^\beta(D^\beta r) &= D^{2\beta}r \\ D^\beta(D^{2\beta}r) &= D^{3\beta}r \\ &\vdots \end{aligned}$$

The control can also be written in terms of the pseudostate as

$$u = -k_P r - k_D D^{\nu\beta} r = -[k_P \ \cdots \ k_D \ \cdots] \begin{bmatrix} r \\ D^\beta r \\ \vdots \\ D^{\nu\beta} r \\ \vdots \end{bmatrix} = -\tilde{K}\xi, \quad (1.70)$$

where the notation “ $\cdots k_D \cdots$ ” means that k_D will occupy the ν^{th} position¹¹, and all other elements are zero. That is, in terms of the pseudostate, we can write the controller in full-*pseudostate* feedback form. According to Newtons second law, this control signal will affect only the last row of the matrix equation Eq. (1.69). In other words, in terms of the pseudostate ξ , the dynamical equation for a double integrator is given by

$$D^\beta \xi = \tilde{A}\xi + \tilde{B}u,$$

where $\tilde{B} = [0 \ 0 \ \cdots \ 1]^T$. This is referred to as the *pseudostate equation*, and is an obvious generalization of the state equation for integer-order LTI systems. We formally call β the *commensurate fractional order* of the system, but often (and when no confusion can arise) it will simply be referred to as the *fractional order* of the system. A more detailed discussion of the pseudostate equation for general fractional controllers will be given in Chapter 2, Sec. 2.2.

Utilizing Eq. (1.70), the fractionally-controlled double integrator Eq. (1.68) can therefore be written as

$$D^\beta \xi = \tilde{A}\xi - \tilde{B}\tilde{K}\xi = (\tilde{A} - \tilde{B}\tilde{K})\xi = \tilde{A}_{CL}\xi. \quad (1.71)$$

¹¹Here and elsewhere, unless stated otherwise, we use a “0-indexing” convention so that the first position in an array is referred to as the “0th” element.

Explicitly, for the fractionally-damped harmonic oscillator Eq. (1.68), the closed-loop pseudostate equation is given by

$$D^\beta \begin{bmatrix} r \\ D^\beta r \\ D^{2\beta} r \\ \vdots \\ D^{2-\beta} r \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -k_P & \cdots & -k_D & \cdots & \cdots \end{bmatrix} \begin{bmatrix} r \\ D^\beta r \\ D^{2\beta} r \\ \vdots \\ D^{2-\beta} r \end{bmatrix},$$

where in the last row of the above matrix, $-k_D$ occupies the ν^{th} position and all other omitted elements are zero.

Given that the closed-loop equations of motion are now given simply by $D^\beta \xi = \tilde{A}_{CL} \xi$, we can utilize the Mittag-Leffler function to form the *pseudostate transition matrix* (pSTM) just as we have done in Sec. 1.1.2. Utilizing the fact that the ML function solves the scalar FDE (see Eq. (1.66)), we can solve this matrix FDE using the *matrix Mittag-Leffler function*:

$$D^\beta \xi = \tilde{A}_{CL} \xi, \quad \xi(0) = \xi_0 \quad \Rightarrow \quad \xi(t) = E_\beta(t \tilde{A}_{CL}) \xi_0,$$

with pSTM given by

$$\Phi(t, 0) = E_\beta(t \tilde{A}_{CL}) = \sum_{k=0}^{\infty} \frac{t^k (\tilde{A}_{CL})^k}{\Gamma(1 + \alpha k)}. \quad (1.72)$$

All formalism of Sec. 1.1.2 can similarly be translated to the fractional-order case, such as finding the explicit form of this pSTM through Jordan decomposition. We will illustrate this explicitly in Chapter 2 when we discuss fractional consensus controllers for multiagent systems.

One important result deserves mention here, however. For integer-order systems, we are well aware that an LTI system of the form $\dot{\mathbf{x}} = M\mathbf{x}$ is stable if the matrix M is Hurwitz, or equivalently if all eigenvalues of M are in the left-half complex plane. Similarly, the fractional pseudo-state equation $D^\beta \xi = \tilde{M} \xi$ of dimension n is stable if eigenvalues λ_i of \tilde{M} satisfy

$$|\operatorname{Arg} \lambda_i| > \beta \frac{\pi}{2}, \quad \text{for all } i = 1, 2, \dots, n. \quad (1.73)$$

This is known as *Matignon's theorem* [16]. Note that this condition reduces to the Hurwitz condition for integer-order systems when we take $\beta \rightarrow 1$.

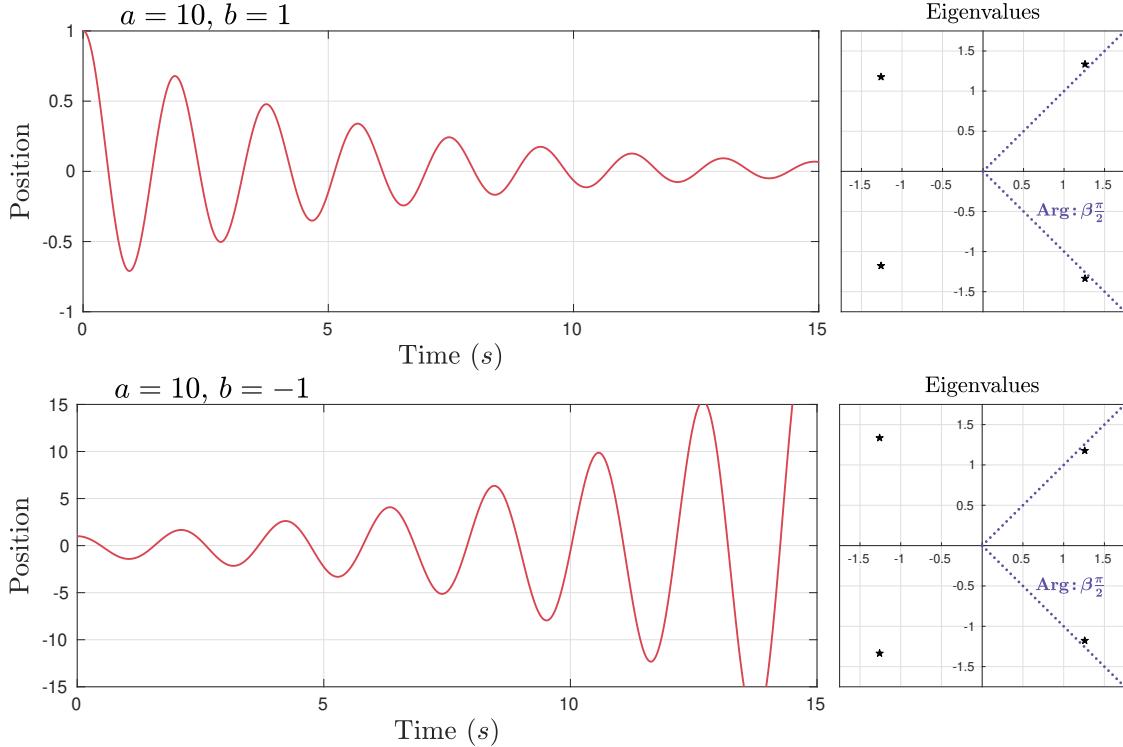


Figure 1.6: Illustration of Matignon’s theorem, Eq. (1.73).

We illustrate Matignon’s theorem in Fig. 1.6. In this figure we simulate the fractional order system

$$D^{0.5} \mathbf{X} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -a & -b & 0 & 0 \end{bmatrix} \mathbf{X}$$

for two different choices of $\{a, b\}$. To the right of the trajectories we show the distribution of the eigenvalues along with the critical stability boundary as given by Matignon’s theorem in Eq. (1.73). When the argument of one conjugate pair falls below $0.5(\pi/2)$, the system becomes unstable. We will show explicitly how Matignon’s theorem arises for fractional controlled systems in Chapter 2.

1.3.4 Numerical Methods: The Fractional Chebyshev Collocation Method

We will now outline the solution method which we will use throughout this work to propagate fractional differential equations. The method we will use is based on [34, 35, 36]; these references can be consulted for a more detailed and general exposition of the method. For notational clarity, throughout this section (and this section alone) we will use standard bold-faced letters to represent column vectors (*e.g.*, \mathbf{a} , \mathbf{B}) and “blackboard-bold” letters to represent matrices (*e.g.*, \mathbb{A} , \mathbb{B}).

Given an ordinary differential equation with initial conditions, a solution can be found through collocation methods using polynomial interpolation. The main strategy of these methods can be illustrated as follows: For a general m^{th} -order differential equation

$$F\{x, f(x), f'(x), f''(x), \dots, f^{(m)}(x)\} = 0,$$

along with m initial conditions,

$$f(0) = f_0, \quad f'(0) = f'_0, \quad \dots, \quad f^{(m-1)}(0) = f_0^{(m-1)},$$

we assume that we can approximate our solution by a n^{th} -order polynomial

$$f(x) \approx g(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n,$$

where a_i , $i = 0, 1, \dots, n$, are $N = n + 1$ yet to be determined coefficients. The m initial conditions determine m coefficients for our approximate solution $g(x)$. The remaining coefficients can be found by evaluating the differential equation at $(N - m)$ points in the domain, the *collocation points*. This provides $(N - m)$ equations which can be solved for the remaining unknown coefficients. The resulting approximating function $g(x)$ matches the actual solution $f(x)$ at the collocation points.

As a brief illustration, let us consider the first-order differential equation

$$f'(x) - f(x) = 0,$$

with initial condition $f(0) = f_0 = 1$. Imagine that we are unaware that the actual solution if given by

$$f(x) = e^x.$$

To approximately solve this differential equation in the domain $x = [0, 1]$, let us choose $N = 5$ collocation points within this domain, $x_i = \{0, 0.1, 0.2, 0.4, 0.6\}$, and choose our approximate solution to be

$$g(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4.$$

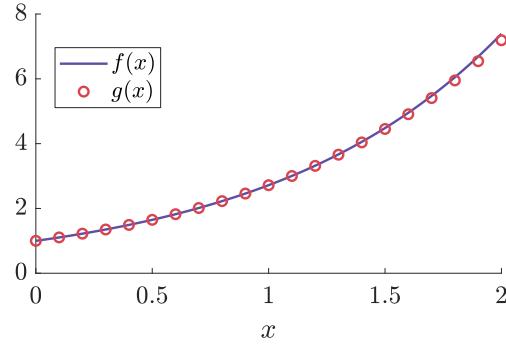
The initial condition at x_0 determines a_0 as $g(0) = a_0 = 1$. The remaining coefficients can be solved by evaluating the result of the differential equation $g'(x) - g(x) = 0$ at the remaining $N - 1$ collocation points x_1, \dots, x_4 . This provides

$$\begin{aligned} 1 - 0.9a_1 - 0.19a_2 - 0.029a_3 - 0.0039a_4 &= 0, \\ 1 - 0.8a_1 - 0.36a_2 - 0.112a_3 - 0.0304a_4 &= 0, \\ 1 - 0.6a_1 - 0.64a_2 - 0.416a_3 - 0.2304a_4 &= 0, \\ 1 - 0.4a_1 - 0.84a_2 - 0.864a_3 - 0.7344a_4 &= 0. \end{aligned}$$

Solving these four equations for the four unknown coefficients, we find our approximate polynomial solution to be

$$g(x) = 1 + 0.999721x + 0.50253x^2 + 0.156678x^3 + 0.0580289x^4,$$

which we plot with the actual solution, $f(x) = e^x$, below:



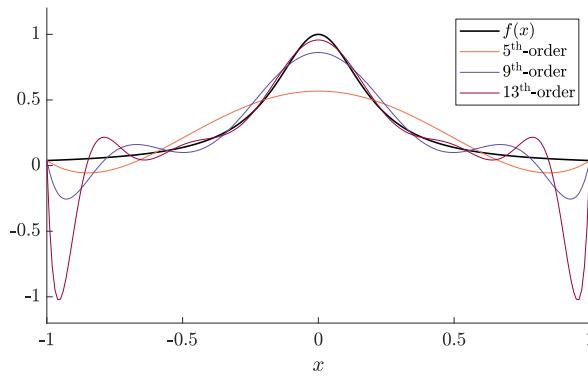
We see that, in this case, $n = 4$ collocation points accurately solves the differential equation inside the domain (and indeed gives a good approximation to the solution slightly beyond the domain).

Of course, as we increase the number of collocation points throughout the domain of interest, we can increase the order of our polynomial approximating function. As long as the

true solution and it's first m derivatives are continuous, this will provide a more accurate fit to the true solution. The way in which we choose the collocation points, however, can have a large effect on the accuracy of the polynomial approximation at different points in the domain. As a classic illustration, we consider the polynomial interpolation of the Runge function,

$$f(x) = \frac{1}{1 + 25x^2}.$$

We can interpolate this function as a polynomial, using equally spaced points between $x = [-1, 1]$; the 5th, 9th, and 13th-order interpolation is plotted below:



Though the polynomial approximation converges to the actual function in the inner region of the domain window as the interpolation order increases, the deviation at the endpoints of the domain window actually increases. This is known as *Runge's phenomenon*.

Intuitively, we expect that we can combat this phenomenon by increasing the density of collocation points near the endpoints of the domain window. An algorithmic way of doing this is to take as our collocation points the horizontal projection of equally spaced points along a circle (see Fig. 1.7). For $N = n + 1$ points distributed in this manner between $x \in [-1, 1]$, these projections are given by

$$x_k = \cos k\theta_n = \cos \left(\frac{k\pi}{n} \right), \quad k = 0, 1, 2, \dots, n.$$

These are known as the *Chebyshev-Gauss-Lobatto points*,¹² since they are the extrema of the Chebyshev polynomials of the first kind, defined recursively as

¹²Note that the Chebyshev-Gauss-Lobatto points are distinct from the Chebyshev *nodes*, $\bar{x}_k = \cos((2k - 1)\pi/2n)$, for $k = 1, 2, \dots, n$, which are the *roots* of the Chebyshev polynomials.

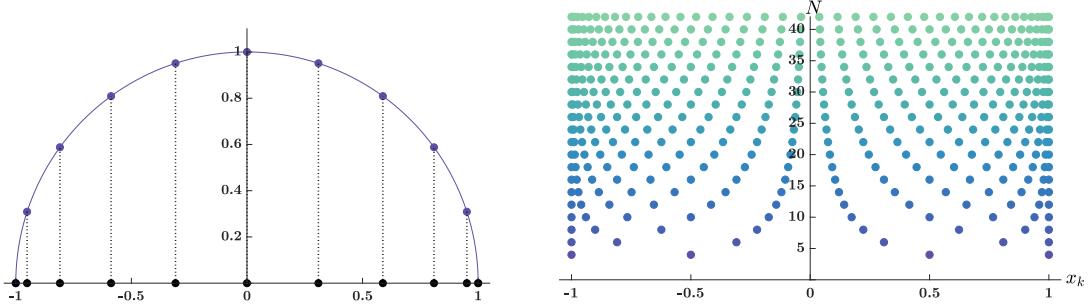


Figure 1.7: Collocation points as horizontal projections of equally spaced points around the unit circle (left). At right we show the distribution of these collocation points as we increase N .

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{k+1} = 2xT_k(x) - T_{k-1}(x),$$

or, equivalently for $|x| \leq 1$,

$$T_k(x) = \cos(k \arccos x).$$

The first five Chebyshev polynomials are given explicitly as

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1, \end{aligned}$$

and the Chebyshev polynomials $T_4(x)$, $T_6(x)$, and $T_{12}(x)$ are plotted in Fig. 1.8.

We notice that the Chebyshev polynomials generally oscillate faster at the endpoints of the domain window $x \in [-1, 1]$. These polynomials can thus be used as a basis set to construct solutions using the collocation method; roughly speaking, the faster oscillation of the polynomials at the endpoints increases the “resolution” of the approximate solution at these endpoints, and serves to mitigate Runge’s phenomenon. Indeed, it can be shown ([37]) that polynomial interpolation using this basis — with the Chebyshev-Gauss-Lobatto (CGL) points as collocation points — minimizes the interpolation error.

We can approximate a continuous function $y(t)$ as a series expansion in Chebyshev

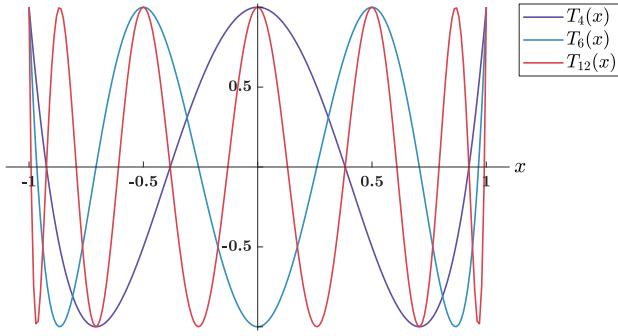


Figure 1.8: A few select Chebyshev polynomials $T_n(x)$. The density of the extremal points (CGL points) increases near the domain endpoints $\{-1, 1\}$.

polynomials as

$$y(t) \approx y_n(t) = \sum_{k=0}^n a_k T_k(t) = \mathbf{T}^T(t) \mathbf{a},$$

where $\mathbf{T}(t)$ represents the column vector built from Chebyshev polynomials $T_k(t)$ and \mathbf{a} is the column vector of coefficients a_k , from $k = 0$ to $k = n$. To solve differential equations using the collocation method as above, we will evaluate this polynomial at the CGL collocation points. Note that the value of the Chebyshev polynomials $T_i(t)$ at the extrema points of $T_n(t)$, $t_k = \cos(k\pi/n)$, are given by

$$T_i(t_k) = \cos \frac{ik\pi}{n}, \quad i < n. \quad (1.74)$$

The Chebyshev polynomials evaluated at the CGL points satisfy the orthogonality relationship

$$\sum_{k=0}^n {}' T_i(t_k) T_j(t_k) = \begin{cases} 0 & i \neq j \\ n/2 & i = j \neq 0 \\ n & i = j = 0 \end{cases}$$

where the prime on the summation symbol indicates that the first and last terms of the summation must be halved. These orthogonality conditions allow us to solve for the coefficients a_k in terms of the solution function $y(t)$. Let us require that the series expansion

(up to order n) at a collocation point t_k matches the true solution:

$$\sum_{i=0}^n a_i T_i(t_k) = y(t_k).$$

Both sides are now multiplied by $T_j(t_k)$ and summed over k according to the orthogonality condition:

$$\sum_{k=0}^n \left(\sum_{i=0}^n a_i T_i(t_k) \right) T_j(t_k) = \sum_{k=0}^n y(t_k) T_j(t_k).$$

For $j = 0$, this gives

$$\begin{aligned} a_0 n &= \sum_{k=0}^n y(t_k) T_0(t_k) \\ &= \frac{1}{2} T_0(t_0) y(t_0) + T_0(t_1) y(t_1) + T_0(t_2) y(t_2) + \cdots + \frac{1}{2} T_0(t_n) y(t_n) \\ &= \frac{1}{2} y(t_0) + y(t_1) + y(t_2) + \cdots + \frac{1}{2} y(t_n), \end{aligned}$$

where in the last line we have used property Eq. (1.74). For $1 \leq j \leq n$, we have

$$\begin{aligned} a_j \frac{n}{2} &= \sum_{k=0}^n y(t_k) T_j(t_k) \\ &= \frac{1}{2} T_j(t_0) y(t_0) + T_j(t_1) y(t_1) + T_j(t_2) y(t_2) + \cdots + \frac{1}{2} T_j(t_n) y(t_n). \end{aligned}$$

We can therefore collect all coefficients into the matrix equation

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \underbrace{\frac{1}{n} \begin{bmatrix} \frac{1}{2} & 1 & \cdots & 1 & \frac{1}{2} \\ -1 & 2T_1(t_1) & \cdots & 2T_1(t_{n-1}) & 1 \\ 1 & 2T_2(t_1) & \cdots & 2T_2(t_{n-1}) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (-1)^{n-1} & T_{n-1}(t_1) & \cdots & T_{n-1}(t_{n-1}) & 1 \\ \frac{1}{2}(-1)^n & T_n(t_1) & \cdots & T_n(t_{n-1}) & \frac{1}{2} \end{bmatrix}}_{=\mathbb{H}} \begin{bmatrix} y(t_0) \\ y(t_1) \\ \vdots \\ y(t_n) \end{bmatrix}.$$

Defining the coefficient vector $\mathbf{a} = [a_0; a_1; \cdots; a_n]$, the matrix \mathbb{H} as above, and the discretized solution vector $\mathbf{y}_d = [y(t_0); y(t_1); \cdots; y(t_n)]$, this is more concisely written as

$$\mathbf{a} = \mathbb{H} \mathbf{y}_d. \quad (1.75)$$

We therefore see that we can solve for the coefficients of the interpolating polynomial by constructing the \mathbb{H} matrix and multiplying it against the vector built from the function evaluated at the N collocation points.

Since we will be using the collocation method to solve differential equations, let us determine the action of the derivative operator on our polynomial expansion,

$$\frac{d}{dt} \mathbf{y}_n(t) = \frac{d}{dt} (\mathbf{T}^T(t) \mathbf{a}) = \left(\frac{d}{dt} \mathbf{T}^T(t) \right) \mathbf{a}.$$

Since the derivatives of the Chebyshev polynomials are given by

$$\frac{d}{dt} T_n(t) = \begin{cases} 2n(T_{n-1} + T_{n-3} + \cdots + T_1), & n \text{ even} \\ 2n(T_{n-1} + T_{n-3} + \cdots + T_2) + nT_0, & n \text{ odd} \end{cases} \quad (1.76)$$

we can express the derivative of the Chebyshev polynomial vector \mathbf{T} as a matrix equation

$$\frac{d}{dt} \mathbf{T}(t) = \frac{d}{dt} \begin{bmatrix} T_0(t) \\ T_1(t) \\ \vdots \\ T_n(t) \end{bmatrix} = \mathbb{D}^1 \mathbf{T}(t), \quad (1.77)$$

where \mathbb{D}^1 is a constant lower-triangular matrix of dimension $N \times N$. It is straightforward to show from Eq. (1.76) that the elements of \mathbb{D}^1 are given by

$$\begin{aligned} [\mathbb{D}^1]_{2j-1,0} &= 2j - 1, & j \geq 1, \\ [\mathbb{D}^1]_{2j-1,i,i} &= 2(2j - 1 + i), & j \geq 1, i \geq 1, \end{aligned}$$

so that, *e.g.*, for a Chebyshev expansion of order $n = 4$ ($N = 5$ collocation points), we have

$$\mathbb{D}^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 3 & 0 & 6 & 0 & 0 \\ 0 & 8 & 0 & 8 & 0 \end{bmatrix}.$$

An analogous matrix exists for a fractional derivative acting upon \mathbf{T} , as we will discuss below.

We mention here that the above relationships apply to Chebyshev polynomials $T_n(x)$ over the domain $x \in [-1, 1]$. It is straightforward to generalize the above results to the domain $z \in [a, b]$. This generalization is accomplished through a scale transformation,

$$x \rightarrow \beta(z - a) - 1, \quad \text{with } \beta = \frac{2}{b - a}.$$

The scale transformation takes $T_k(x) \rightarrow T_k^*(z)$, with these scaled Chebyshev polynomials defined recursively as

$$T_0^*(z) = 1, \quad T_1^*(z) = \beta(z - a) - 1, \quad T_k^*(z) = 2(\beta(z - a) - 1)T_{k-1}^*(z) - T_{k-2}^*(z).$$

All other results above are generalized accordingly through the scale transformation. Details can be found in [35]. For the purposes of this section, however, we will continue to use the unscaled Chebyshev polynomials $T_k(x)$ in the domain $x \in [-1, 1]$.

We now illustrate how one can utilize this Chebyshev expansion to solve differential equations using the collocation method. For this illustration, we will consider a simple 1-dimensional first-order ODE,

$$\dot{y}(t) + by(t) = 0, \quad y(t_a) = y_a, \quad (1.78)$$

and we will assume we are interested in the solution of this function over the domain $t \in [t_a, t_b] = [-1, 1]$. Generalization to higher-order or multi-dimensional systems is also straightforward, but we will refer the reader to [36] for details. Once the Chebyshev collocation method is described for this integer-order differential equation, the method will be generalized to include fractional-order derivative terms in the differential equation.

Replacing the function $y(t)$ in Eq. (1.78) with its Chebyshev expansion (of order n), we have

$$\begin{aligned} & \frac{d}{dt} \mathbf{T}^T(t) \mathbf{a} + b \mathbf{T}^T(t) \mathbf{a} = 0 \\ \Rightarrow & \mathbf{T}^T(t) \mathbb{D}^{1T} \mathbf{a} + b \mathbf{T}^T(t) \mathbf{a} = 0 \\ \Rightarrow & \mathbf{T}^T(t) (\mathbb{D}^{1T} \mathbf{a} + b \mathbf{a}) = 0 \\ \Rightarrow & \mathbf{T}^T(t) (\mathbb{D}^{1T} \mathbb{H} + b \mathbb{H}) \mathbf{y}_d = 0, \end{aligned} \quad (1.79)$$

where in the last line we have replaced the unknown expansion coefficients via Eq. (1.75). Note that \mathbf{y}_d is the $N \times 1$ column vector describing the discretized solution (the value of the solution at the $N = n + 1$ collocation points), and $\mathbf{T}^T(\mathbb{D}^{1T}\mathbb{H} + b\mathbb{H})$ is a $1 \times N$ row vector. Of course the solution $y(t)$ is at this point still unknown; we therefore require N equations to solve for the N components of \mathbf{y}_d . To this end, we can evaluate Eq. (1.79) at the N collocation points t_k , $k = 0, 1, \dots, n$,

$$\left. \begin{aligned} \mathbf{T}^T(t_0) (\mathbb{D}^{1T}\mathbb{H} + b\mathbb{H}) \mathbf{y}_d &= 0 \\ \mathbf{T}^T(t_1) (\mathbb{D}^{1T}\mathbb{H} + b\mathbb{H}) \mathbf{y}_d &= 0 \\ &\vdots \\ \mathbf{T}^T(t_n) (\mathbb{D}^{1T}\mathbb{H} + b\mathbb{H}) \mathbf{y}_d &= 0 \end{aligned} \right\} \Rightarrow \mathbb{T}_n (\mathbb{D}^{1T}\mathbb{H} + b\mathbb{H}) \mathbf{y}_d = \mathbf{0}, \quad (1.80)$$

where \mathbb{T}_n is now the $N \times N$ matrix given by

$$\mathbb{T}_n = \begin{bmatrix} \mathbf{T}^T(t_0) \\ \mathbf{T}^T(t_1) \\ \vdots \\ \mathbf{T}^T(t_n) \end{bmatrix} = \begin{bmatrix} T_0(t_0) & T_1(t_0) & \cdots & T_n(t_0) \\ T_0(t_1) & T_1(t_1) & \cdots & T_n(t_1) \\ \vdots & \vdots & \ddots & \vdots \\ T_0(t_n) & T_1(t_n) & \cdots & T_n(t_n) \end{bmatrix}.$$

Inverting the matrix equation Eq. (1.80) simply provides the trivial solution $\mathbf{y}_d = \mathbf{0}$. To find the non-trivial solution, we must incorporate the initial condition $y(t_a) = y(-1) = y_a$. Recall, however, that the collocation points

$$t_k = \cos\left(\frac{k\pi}{n}\right), \quad k = 0, 1, 2, \dots, n$$

are reverse-chronologically ordered, so that $t_0 = 1$ and $t_n = -1$. The discretized solution vector \mathbf{y}_d is therefore also reverse-chronologically ordered, and therefore the initial condition should be assigned to the *last* element of \mathbf{y}_d . We will find it convenient, however, to instead define the *chronological-ordered* discretized solution,

$$\mathbf{y}_{\tilde{d}} = \mathbb{P}_N \mathbf{y}_d = [\underbrace{y(t_n); y(t_{n-1}); \cdots; y(t_1)}_{y(-1)}; y(t_0)],$$

where \mathbb{P}_N is the $N \times N$ permutation matrix which reverses the rows of an array from top to

bottom,

$$\mathbb{P}_N = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix}.$$

Since $\mathbb{P}_N = \mathbb{P}_N^{-1}$, the chronologically-ordered equivalent of Eq. (1.80) is given by

$$\mathbb{P}_0 \mathbb{T}_N (\mathbb{D}^{1T} \mathbb{H} + b \mathbb{H}) \mathbb{P}_0 \mathbf{y}_{\tilde{d}} = \mathbb{M} \mathbf{y}_d = \mathbf{0}. \quad (1.81)$$

The first row of this equation now contains the differential equation evaluated at the initial time.

To incorporate the initial condition, we can thus replace the first row in Eq. (1.81) by $y(t_a) = y_a$. We write this as

$$\tilde{\mathbb{M}} \mathbf{y}_{\tilde{d}} = y_a \tilde{\mathbf{1}},$$

where $\tilde{\mathbb{M}}$ represents the matrix found by replacing the first row of \mathbb{M} with $[1 \ 0 \ \cdots \ 0]$, and where $\tilde{\mathbf{1}} = [1; 0; \cdots; 0]$. The matrix \mathbb{M} is of dimension $N \times N$, and can be shown to be of full rank [35]. This equation can therefore be inverted to give \mathbf{y}_d , the solution of the differential equation at the collocation points:

$$\mathbf{y}_{\tilde{d}} = \tilde{\mathbb{M}}^{-1} y_a \tilde{\mathbf{1}}.$$

It is now hopefully apparent that, if we seek solutions to *fractional* differential equations, we need the fractional derivative of the Chebyshev polynomial vector, $D^\alpha \mathbf{T}$. That is, we need the fractional analog of the \mathbb{D}^1 matrix,

$$\frac{d}{dt} \mathbf{T}(t) = \mathbb{D}^1 \mathbf{T}(t) \quad \rightarrow \quad \frac{d^\alpha}{dt^\alpha} \mathbf{T}(t) = \mathbb{D}^\alpha \mathbf{T}(t).$$

Since the Chebyshev polynomials within \mathbf{T} are built from power functions t^p (for integer p), and furthermore since from Sec. 1.3.2 we know how to compute fractional derivatives of power functions, determining the form of \mathbb{D}^α is relatively straightforward. With \mathbb{D}^α obtained, the collocation method of solving a fractional differential equation proceeds identically to the

integer-order example above. We will call this method of solution the *fractional Chebyshev collocation* (FCC) method.

For instance, given a fractional differential equation and initial condition

$$D^1 y(t) - D^{0.6} y(t) - y(t) = 0, \quad \text{with } y(t_a) = y_a,$$

one first replaces the solution with the Chebyshev expansion,

$$D^1(\mathbf{T}^T(t)\mathbf{a}) - D^{0.6}(\mathbf{T}^T(t)\mathbf{a}) - (\mathbf{T}^T(t)\mathbf{a}) = 0,$$

then determines the action of the derivatives (both integer- and fractional-order) on \mathbf{T} ,

$$\begin{aligned} D^1\mathbf{T}(t) &= \mathbb{D}^1\mathbf{T}(t), & D^\alpha\mathbf{T}(t) &= \mathbb{D}^\alpha(t)\mathbf{T}(t) \\ \Rightarrow \quad \mathbf{T}^T(t)(\mathbb{D}^{1T}\mathbf{a} - \mathbb{D}^{\alpha T}(t)\mathbf{a} - \mathbf{a}) &= 0, \end{aligned}$$

then replaces the coefficients with the discretized solution utilizing Eq. (1.75),

$$\mathbf{T}^T(t)(\mathbb{D}^{1T}\mathbb{H} - \mathbb{D}^{\alpha T}(t)\mathbb{H} - \mathbb{H})\mathbf{y}_d = 0.$$

Note that due to the fact that fractional derivatives of power functions involve fractional powers of t (see, *e.g.*, Eq. (1.57)), the fractional-differential matrix \mathbb{D}^α becomes time-dependent. This equation is then evaluated at the collocation points as before,

$$\begin{bmatrix} \mathbf{T}^T(t_0)(\mathbb{D}^{1T}\mathbb{H} - \mathbb{D}^{\alpha T}(t_0)\mathbb{H} - \mathbb{H}) \\ \vdots \\ \mathbf{T}^T(t_n)(\mathbb{D}^{1T}\mathbb{H} - \mathbb{D}^{\alpha T}(t_n)\mathbb{H} - \mathbb{H}) \end{bmatrix} \mathbf{y}_d = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The ordering of the arrays is then flipped to chronological order, the first row on both sides is replaced with the initial condition, and the system is inverted to provide the discretized solution \mathbf{y}_d . The main significant difference between the FCC method and the integer-order collocation method discussed above is the form and time-dependence of the \mathbb{D}^α matrix. As mentioned, the derivation of this matrix is fairly straightforward using the formalism introduced in Sec. 1.3.2. Care must be taken, however, to ensure that the fractional derivatives are well defined over the domain.

For the purposes of this work, it is more appropriate to omit these details and point the interested reader to the references. The above discussion, however, accurately summarizes

how one computes solutions to fractional differential equations (that is, how one *propagates* the trajectory of a system obeying fractional differential equations) using the fractional Chebyshev collocation method. In this work, we have utilized the **FCC toolbox** MATLAB package, developed by A. Dabiri et. al. [35, 36, 38], to generate our fractional Chebyshev differentiation matrices \mathbb{D}^α . The derivation and analytical form of this matrix, along with in-depth discussion and benchmarking of the FCC method, can be found in these references as well as in [34]. In addition, the utility of this method is demonstrated in [39] and [40], where the FCC method was used to design feedback control in time-delay systems and systems with periodic coefficients.

1.4 REVIEW OF ORBITAL MECHANICS

A satellite orbiting a celestial body such as the Earth obeys Newtonian dynamics, and thus its complete trajectory can be computed as long as one has knowledge of all forces acting on the satellite along with the full *state* of the satellite at any given time. Assuming that the celestial body is a uniform sphere with total mass M , the primary force acting upon an uncontrolled satellite is given by Newton's law of universal gravitation,

$$\mathbf{F} = m\ddot{\mathbf{r}} = -\frac{GMm}{r^3}\mathbf{r}, \quad (1.82)$$

where m is the mass of the satellite, \mathbf{r} is the vector from the center of the gravitating body to the satellite, $r = |\mathbf{r}|$, and $G = 6.673 \times 10^{-20} \text{ km}^3\text{kg}^{-1}\text{s}^{-2}$ is the gravitational constant. It is common to define the *standard gravitational parameter* of a central body by $\mu = GM$, as this parameter is known to greater precision for bodies in the solar-system; for the Earth this parameter is approximately $\mu_E = 3.986 \times 10^5 \text{ km s}^{-2}$. Other *perturbing* forces can contribute to the natural (uncontrolled) motion, such as those from atmospheric drag, solar radiation pressure, and perturbations to the gravity potential from third bodies and non-spherical mass distributions of the central body. For the remainder of this work we will neglect these perturbing forces. In most cases this will be well-justified since these perturbing forces are typically negligible compared to the forces applied by the control systems considered herein.

1.4.1 State Vectors for Orbiting Satellites

The state of a single satellite can be characterized in several ways, but in all cases must comprise at least six independent pieces of parametric information. The most familiar way to do this from a Newtonian perspective is to specify, in some inertial frame, the position and velocity coordinates of the satellite at a specified time $t = t_0$,

$$\mathbf{X}_0 = \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{bmatrix}, \quad \mathbf{r}_0 = [x_0, y_0, z_0]^T, \quad \mathbf{v}_0 = [\dot{x}_0, \dot{y}_0, \dot{z}_0]^T.$$

To take advantage of the spherical symmetry inherent in Eq. (1.82), the origin of the inertial¹³ reference frame is typically taken to be the center of the gravitating body.

For the Earth, the conventional reference frame is taken to be Cartesian, with a right-handed triad of unit vectors $\hat{\mathbf{I}}\hat{\mathbf{J}}\hat{\mathbf{K}}$. These unit vectors are oriented in such a way that $\hat{\mathbf{K}}$ points in the direction of the North Pole, and $\hat{\mathbf{I}}$ and $\hat{\mathbf{J}}$ lie on the equatorial plane with $\hat{\mathbf{I}}$ pointing towards the *vernal equinox*. The vernal equinox direction (often denoted Υ) points towards the Sun during the vernal equinox (approximately March 20), and is sometimes referred to as the first point of Aries since this direction corresponded with the constellation of Aries during the Hellenistic period.¹⁴ The $\hat{\mathbf{J}}$ direction necessarily points in a direction 90° eastward from $\hat{\mathbf{I}}$. This coordinate system is referred to as the Earth-Centered Inertial (ECI) frame.

A second way to characterize the complete state of an orbiting satellite is by specifying its Keplerian *orbit elements*. If all forces except for the gravitational force of central body are neglected, the instantaneous position and velocity of the satellite as defined above would correspond to a particular Keplerian orbit, the trajectory of which is described by a fixed ellipse positioned such that one focus lies at center of the gravitating body. A particular elliptical trajectory can be parameterized by five quantities: Two quantities are used to specify the scale and shape of the ellipse, and three additional quantities are required to

¹³In most cases this is in fact a *pseudo*inertial reference frame due to forces acting on the central body; in the case of Earth-orbiting satellites, the reference frame with origin at the center of the Earth itself orbits around the Sun. For all intents and purposes discussed herein, this motion can be neglected.

¹⁴Due to the $\sim 26,000$ -year axial precession of the Earth, this direction now points towards the direction of the constellation Pisces and will next point towards Aquarius.

parameterize its orientation in 3D space. A sixth quantity then determines the position of the satellite along the ellipse at any given time. Thus, just as in the ECI state description, a total of six independent quantities is used to completely specify the state at a given time.

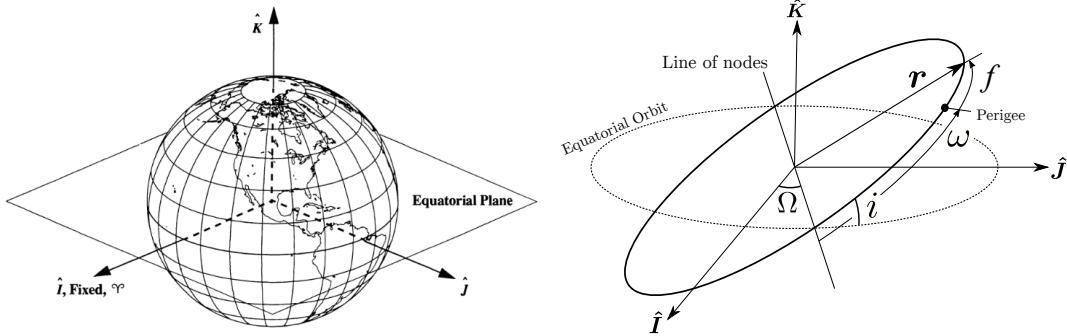


Figure 1.9: Two methods of characterizing the state of an orbiting satellite. At left we show the axes of the ECI cartesian coordinate system, and at right we show the Keplerian orbital elements. Note that orbital elements a and e are not shown.

The scale and shape of the ellipse is parameterized by the semi-major axis a and eccentricity e . This defines the elliptical trajectory as

$$r(f) = \frac{a(1 - e^2)}{1 + e \cos f}, \quad (1.83)$$

where f , the *true anomaly*, is the angular distance along the ellipse as measured from the point of closest approach (the periapsis). The orientation of the ellipse is then specified by a set of three Euler angles in the ‘ z - x - z ’ rotation-order convention; attaching to the ellipse a Cartesian coordinate frame such that the ellipse is oriented in the xy plane with focus at the origin and periapsis along the x -axis, a general orientation is found by rotating the ellipse first about the z axis by angle Ω , then about the x -axis by angle i , then about the z -axis by angle ω (see Fig. 1.9).

The angle Ω , called the *longitude of the ascending node*, describes the angle away from the original x -axis (before the Euler rotations) to the point where the orbit crosses from the lower hemisphere to the upper hemisphere; this point is called the *ascending node*. Longitude of the ascending node is restricted to the domain $\Omega \in [0^\circ, 360^\circ]$. The angle i is called the *inclination* and describes the angle between the original xy -plane and the orbital plane; the line where

these two planes intersect is called the *line of nodes*. The inclination is restricted to values $i \in [0^\circ, 180^\circ]$, with $i = \{0^\circ, 180^\circ\}$ corresponding to *equatorial* orbits, $i = 90^\circ$ corresponding to *polar* orbits, $i < 90^\circ$ corresponding to *prograde* orbits, and $i > 90^\circ$ corresponding to *retrograde* orbits. Finally, the angle ω which we call the *argument of periapsis* describes the angle around the orbital plane from the ascending node to the periapsis of the orbit. This angle is restricted to the domain $\omega \in [0^\circ, 360^\circ)$. We also note that for an orbiting satellite, a can be any positive length and $e = [0, 1)$. Orbit elements can still be used to characterize *unbound* trajectories. In these cases $e \geq 1$ and the trajectory is either parabolic ($e = 1$) or hyperbolic ($e > 1$). The orientation and instantaneous position is defined through $\{i, \Omega, \omega, f\}$ in the same way as above. Additionally, in both bound and unbound cases, rectilinear motion is possible and corresponds to motion in the purely radial direction (with vanishing orbital angular momentum).

The full state of the satellite is therefore characterized by *either* the position and velocity $\{\mathbf{r}, \mathbf{v}\}$ in the ECI frame, or the set of six orbital elements $\{a, e, i, \Omega, \omega, f\}$. These two descriptions are *almost* in one-to-one correspondence. In the exceptional case of perfectly circular orbits, orbital radius is constant so that there is no unique way to define argument of periapsis ω , and consequently no unique way of defining the true anomaly. In these cases we can simply set $\omega = 0^\circ$ so that true anomaly f is the angular distance along the ellipse from the ascending node to the instantaneous position. This description, however, is not unique, since replacing argument of perigee with $\omega' \neq 0$ and $f' = f - \omega'$ describes the same state. An additional ambiguity arises for equatorial orbits ($i = 0^\circ$). In this case the pair $\{\Omega, \omega\}$ is not uniquely defined. In these cases we can simply set Ω to zero so that angular distance between the vernal equinox (ECI $\hat{\mathbf{I}}$ direction) and the periapsis defined ω . Using these convention choices for circular and/or equatorial orbits, however, the orbit elements (on their prescribed domains) and the ECI vectors $\{\mathbf{r}, \mathbf{v}\}$ stand in one-to-one correspondence. We therefore can find an invertible (nonlinear) transformation between ECI coordinates and orbital elements:

$$\{\mathbf{r}, \mathbf{v}\} \quad \xleftarrow{\text{Nonlinear Trans.}} \quad \{a, e, i, \Omega, \omega, f\}$$

Derivation of these transformations (corresponding to either “left-ward” or “right-ward” as sketched above) is straightforward and outlined in most introductory texts on celestial

mechanics, *e.g.*, [41]. As such, we will not explicitly give them here.

1.4.2 Dynamics of Orbital Motion

The ECI state of the satellite evolves according to the Newtonian dynamics in Eq. (1.82). Since this equation is nonlinear, it is typically integrated numerically to propagate the satellite's trajectory. Furthermore, in general all components of the ECI state are time-dependent as the satellite orbits the body. On the other hand, in the absence of perturbations to the motion, all orbital elements except f are constant; this of course is the main advantage of using the orbital elements to define the state.

The time evolution of f obeys Kepler's second law, stating that the position vector of the satellite sweeps out equal areas of the ellipse in equal time intervals. For elliptical orbits ($0 < e < 1$) this cannot be expressed in closed-form as a function of time. We can propagate f indirectly, however, by introducing the *mean anomaly*, which evolves linearly in time: We know from Kepler's third law that orbital period is given by $T = \sqrt{\mu/a^3}$ (with the proportionality constant μ found from Newton's law of universal gravitation). We can therefore define the mean anomaly at any time t as the angle

$$M(t) = n(t - t_0) + M_0, \quad (1.84)$$

where M_0 is the mean anomaly at initial time t_0 and n is the *mean motion* defined through the period as $n = 2\pi/T = 2\pi\sqrt{a^3/\mu}$. The relationship between mean anomaly and true anomaly is then found by introducing an second auxiliary angle E , called the *eccentric anomaly* which is related to the true anomaly through the relations

$$\cos E = \frac{e + \cos f}{1 + e \cos f}, \quad \sin E = \frac{\sqrt{1 - e^2} \sin f}{1 + e \cos f}. \quad (1.85)$$

The eccentric and mean anomaly are then related through *Kepler's equation*,

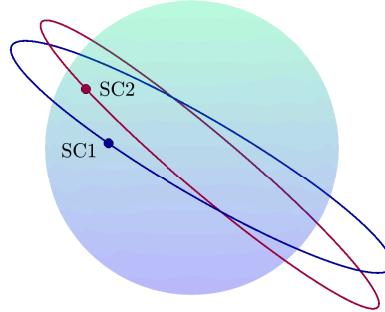
$$M = E - e \sin E. \quad (1.86)$$

A more detailed description and explanation of the relationships between these three anomalies can be found in most introductory texts (see, *e.g.*, [41]).

Therefore, to determine f at any time t given f_0 at t_0 , one computes M_0 through Eqs. (1.85) and (1.86), propagates M_0 to $M(t)$, then computes $f(t)$ from $M(t)$. This last step requires inverting Kepler's equation, which cannot be done in closed-form since it is transcendental; it is typically inverted numerically through various means. Although the mean anomaly M does not correspond to any physical angle in Keplerian motion, it is convenient in that it evolves linearly in time. It is therefore common to represent the set of orbit elements at any given time as $\{a, e, i, \Omega, \omega, M\}$, with M in place of f , since there is a one-to-one correspondence between these two anomalies.

1.4.3 Relative Orbits

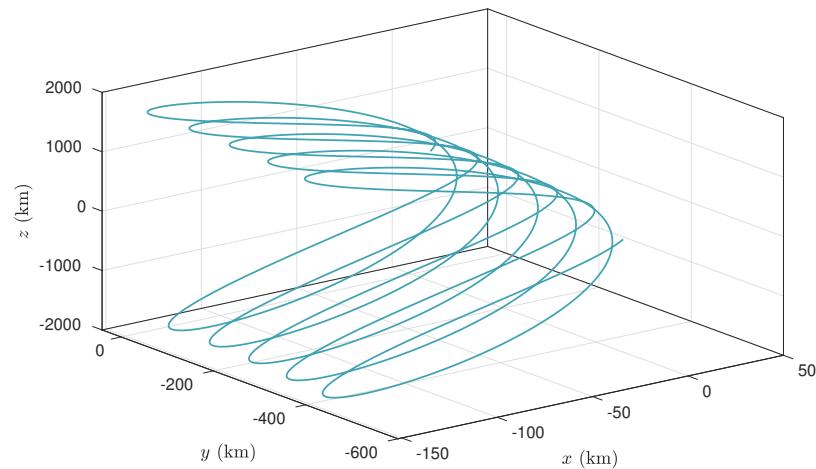
In many practical applications, one is concerned with the *relative orbit* between two or more bodies orbiting a common center. We illustrate such a scenario, in the case of two Earth-orbiting spacecraft, in the figure below.



At some initial time $t = t_0$, each spacecraft will be described by its own set of orbit elements. We denote these orbit element sets as ϵ_1 and ϵ_2 , corresponding, respectively, to the spacecraft labeled “SC1” and “SC2”. We can also describe each spacecraft by its corresponding ECI position-velocity states at time $t = t_0$, denoted respectively by $\mathbf{X}_1 = [\mathbf{r}_1^T \ \mathbf{v}_1^T]^T$ and $\mathbf{X}_2 = [\mathbf{r}_2^T \ \mathbf{v}_2^T]^T$. Either choice of these states (either ECI states or orbital element sets) can be used to independently propagate each orbital trajectory, as described above. In this way we can determine the full evolution of the two-spacecraft system.

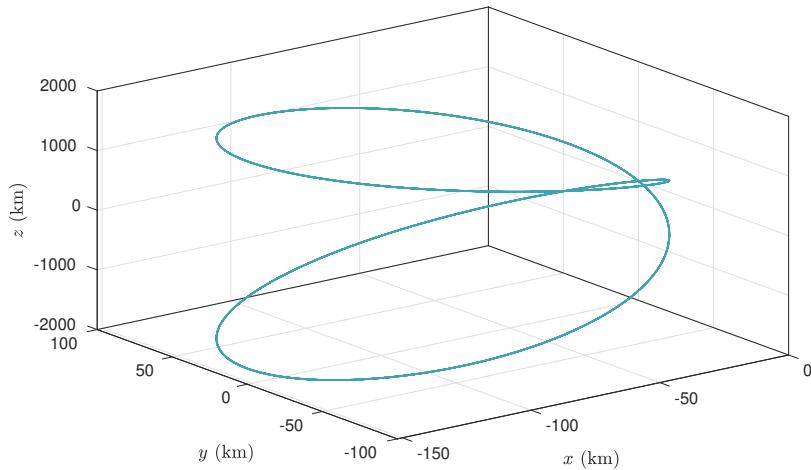
In many cases, however, one is interested in the relative motion of one spacecraft with

respect to another. In the above case, let us imagine we are sitting in the pilot's seat of SC1 and observing the motion of SC2. This motion will be referred to as the *relative orbit trajectory*. Although each individual spacecraft simply traverses an elliptical path, the motion of SC2 *as viewed by* SC1 is in general quite complicated. This is especially true if the spacecraft have different semi-major axes, $a_1 \neq a_2$, for in this case one spacecraft will traverse a complete orbit faster than another. From the SC1 viewpoint, then, SC2 will continuously drift away along some curvilinear trajectory. We illustrate this below for the case $a_1 = 10^4$ km, $a_2 = a_1 + 10$ km, $i_1 = 20^\circ$, and $i_2 = 30^\circ$ (all other orbit elements zero):



In this figure we have plotted the trajectory of SC2 as viewed in a reference frame¹⁵ attached to SC1. In cases where $a_1 = a_2$, however, both spacecraft return to the same relative positions after one complete orbit; the relative orbit trajectories in these cases are always closed curves. Equating the semi-major axes in the above example, $a_1 = a_2 = 10^4$ km, the trajectory becomes

¹⁵This specific reference frame is known as the local-vertical-local-horizontal frame, and will be introduced in Chapter 3.



The above examples were plotted by individually propagating each Keplerian orbit trajectory, then transforming coordinate systems to a reference frame attached to SC1. We will see in Chapter 3 that in cases where the difference in orbit elements is small, one can in fact linearize the dynamics so that the relative orbit trajectory is approximately described by

$$\dot{\mathbf{x}} = A(t)\mathbf{x}, \quad (1.87)$$

where \mathbf{x} represents the position-velocity state of SC2 as measured from SC1. Furthermore, when the orbit of SC1 is approximately circular, the system matrix A becomes time-independent. In other words, in these cases one needs only to propagate the motion of SC2 according to the linearized dynamics in Eq. (1.87). As we will see, this is an extremely convenient description of relative orbital motion and allows us to design controllers using all of the formal tools discussed in this chapter.

1.5 CONCLUDING REMARKS

This chapter has introduced all of the necessary formalism that will be used in Chapters 2 and 3, which contain the main results of this thesis. There were several reasons why I chose to include such a detailed and lengthy introductory exposition. First, considering the context of this work, I decided that one of the primary functions of a masters degree thesis is to demonstrate the knowledge that has been acquired over the course of one's

academic program. Second, on a more personal level, I wholeheartedly subscribe to the notion that one of the most thorough ways to learn a subject is by teaching it. As a result, the stylistic structure of this introduction is candidly pedagogical. Considering the length of this introduction, however, it behooves me to at this point remind the reader of the central topics which will be explored in this thesis, and which are novel products of my research.

In the control of multiagent second-order systems, by far the most common strategy considered in the literature is to use proportional-derivative feedback protocols to bring the system to consensus. Since these controllers can only be tuned by adjusting the proportional and derivative gains, one is limited in how one can adjust the trajectory of the controlled system. For instance, one can increase the derivative gains to reduce the amount of “overshoot” in the trajectories, but this will inherently increase the amount of time it takes to reach consensus. Motivated by these limitations, the main purpose of Chapter 2 is to increase the tunability of consensus controllers. We do so by first introducing integral terms into the controller, essentially forming a PID consensus controller. This adds an additional tunable parameter through the integral control gain. This PID controller is then further generalized by replacing the derivative term in the control law with a fractional derivative (forming the PID^ν consensus controller) whose fractional order can be smoothly tuned between zero and one. This introduces even more tunability into the controller, and thus gives us a much larger amount of freedom in how we can shape the trajectory. The PID and PID^ν controller will be introduced and studied in depth, and the stability of these controllers will be proved.

These results were shown primarily for double-integrator systems, describing inertial vehicles in absense of external forces. Chapter 3 applies both fractional control and consensus control strategies to the more complicated (and interesting) dynamics describing the relative motion of orbiting spacecraft in close proximity. Controllers for these systems typically take the form of PD controllers, so introducing fractional derivative terms similarly introduces more tunability. It will be shown that more optimal trajectories can be achieved by utilizing fractional control; specifically, these controllers can reduce both settling time and integrated control effort (fuel cost). In the latter half of Chapter 3, I will show how consensus controllers

can be used to efficiently construct spacecraft formations. This topic therefore serves as somewhat of a “keystone”, bridging together the topics of Chapter 2 (consensus control) and Chapter 3 (relative orbit control).

CHAPTER 2

FRACTIONAL COOPERATIVE CONTROL

This chapter generalizes the standard consensus control protocols introduced in Chapter 1. We will first introduce proportional-integral-derivative (PID) control for consensus systems, as we will find that integral control is useful in reducing settling time for fractionally controlled systems. We find that these controllers are asymptotically stable as long as certain conditions on the controller gains are satisfied. We will then introduce a fractional-order PID^α consensus controller for multiagent systems, and similarly analyze the stability; to this end we utilize the pseudostate space formalism and the pseudostate transition matrix. These results will be validated by numerically simulating the trajectories for several fractionally-controlled systems communicating through various communication topologies. These results will be compared to trajectories under standard integer-order control, illustrating some of the benefits of introducing fractional control.

2.1 PID CONSENSUS PROTOCOLS

We introduced in Chapter 1 the local voting protocol for second-order agents within an N -agent cooperative system,

$$u_i = - \sum_{j=1}^N a_{ij} \left(k_P(r_i - r_j) + k_D(v_i - v_j) \right), \quad i = 1, 2, \dots, N,$$

where r_i and v_i is the position and velocity of the i^{th} agent, where a_{ij} are the edge-weights corresponding to the communication topology of the N -agent system (with $a_{ij} > 0$ if agent j communicates its state to agent i), and where $k_P, k_D > 0$ are the proportional and derivative gains of the controller. For undirected communication graphs, consensus of the system,

$$(r_i - r_j) \rightarrow 0, \quad (v_i - v_j) \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty, \quad \forall i, j \in \mathcal{N} = \{1, 2, \dots, N\}$$

is reached if and only if the communication topology contains a spanning tree (which is equivalent to the condition that graph Laplacian matrix L has exactly one zero-eigenvalue). See the corollary below Theorem 2 for additional conditions on k_P and k_D for directed graphs.

As indicated by the subscripts on the controller gains, the local voting protocol is in the form of a proportional-derivative (PD) controller: the position differences ($r_i - r_j$) and their derivatives ($v_i - v_j = (\dot{r}_i - \dot{r}_j)$) are fed back to the system in order to drive these quantities to zero. It therefore seems natural to generalize the above control protocol by introducing integral feedback of the form

$$-k_I(\xi_i - \xi_j),$$

where ξ_k is an auxiliary state-variable for agent k , defined as $\xi_k = \int_0^t r_k(\tau) d\tau$. Introducing these additional terms to the local voting protocol gives a *proportional-integral-derivative* (PID) *consensus controller*,

$$\begin{aligned} u_i &= \sum_{j=1}^N a_{ij} (k_I(\xi_j - \xi_i) + k_P(r_j - r_i) + k_D(\dot{r}_j - \dot{r}_i)) \\ &= \sum_{j=1}^N a_{ij} \left(k_I(\xi_j - \xi_i) + k_P \frac{d}{dt}(\xi_j - \xi_i) + k_D \frac{d^2}{dt^2}(\xi_j - \xi_i) \right) \end{aligned} \quad i = 1, 2, \dots, N, \quad (2.1)$$

where in the second line we have written the protocol entirely in terms of the position integral $\xi_i = \int_0^t r_i(\tau) d\tau$, and where k_P , k_I , and k_D are the proportional, integral, and derivative gains, respectively.

We express this system in a state-space form as follows¹: Let us define \mathbf{z}_i as the augmented state for agent i including the integral state, *i.e.*, $\mathbf{z}_i = [\xi_i; r_i; v_i]$, and define the *local* state-vector for the entire system to be $\mathbf{z}_L = [\mathbf{z}_1; \mathbf{z}_2; \dots; \mathbf{z}_N]$. The closed-loop dynamics of the entire system under protocol Eq. (2.1) can be written concisely as

$$\dot{\mathbf{z}}_L = A_{CL}^L \mathbf{z}_L = [I_N \otimes A_3 - L \otimes B_3 K_3] \mathbf{z}_L, \quad (2.2)$$

where A_{CL}^L is the closed-loop system matrix (in the local representation), I_N is the N

¹As in Chapter 1, we will continue to use the short-hand column-vector notation $[\mathbf{a}; \mathbf{b}; \mathbf{c}] \equiv [\mathbf{a}^T \ \mathbf{b}^T \ \mathbf{c}^T]^T$, for general column vectors \mathbf{a} , \mathbf{b} , and \mathbf{c} .

dimensional identity matrix, and

$$A_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad K_3 = [k_I \ k_P \ k_D].$$

We can also represent the closed-loop system in terms of the *global* state vector, defined as $\mathbf{z} = [\boldsymbol{\xi}; \mathbf{r}; \dot{\mathbf{r}}]$, with $\boldsymbol{\xi} = [\xi_1; \xi_2; \dots; \xi_N]$, and \mathbf{r} and $\dot{\mathbf{r}}$ defined similarly. In terms of the global state vector, the closed-loop state-equations can be written as

$$\dot{\mathbf{z}}_G = A_{CL}^G \mathbf{z}_G = [A_3 \otimes I_N - B_3 K_3 \otimes L] \mathbf{z}_G, \quad (2.3)$$

with the closed-loop system matrix taking the explicit form

$$A_{CL}^G = \begin{bmatrix} \mathbf{0} & I_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_N \\ -k_I L & -k_P L & -k_D L \end{bmatrix}. \quad (2.4)$$

In either representation, by setting $k_I = 0$ we recover the dynamics of the PD consensus controller discussed in Chapter 1.

2.1.1 Eigenanalysis of the Closed-loop System and Stability

Consider the state equation corresponding to the closed-loop PID consensus system in the local representation, Eq. (2.2). The conditions for asymptotic stability of the consensus subspace $r_i = r_j, v_i = v_j, \forall i, j \in \mathcal{N}$, are obtained by analysis of the spectrum of the closed-loop linear matrix in Eq. (2.2), which is equivalent [4] to the combined spectrums of

$$A_3 - \mu_i B_3 K_3, \quad i = 1, 2, \dots, N, \quad (2.5)$$

where μ_i are the eigenvalues of L . That is,

$$\text{spec}\{I_N \otimes A_3 - L \otimes B_3 K_3\} = \text{spec}\{\text{diag}\{A_3 - \mu_1 B_3 K_3, A_3 - \mu_2 B_3 K_3, \dots, A_3 - \mu_N B_3 K_3\}\}. \quad (2.6)$$

The eigenvalues $\lambda_i, i = 1, \dots, 3N$, of the closed-loop system are thus easily found from the characteristic equation

$$0 = \prod_{i=1}^N \det(\lambda I_3 - (A_3 - \mu_i B_3 K_3)) = \lambda^3 \prod_{i=2}^N \det(\lambda I_3 - (A_3 - \mu_i B_3 K_3)). \quad (2.7)$$

(The second equality in Eq. (2.7) follows from the fact that the Laplacian matrix always has at least one zero eigenvalue, $\mu_1 = 0$.) We therefore find $3N$ eigenvalues for the closed-loop system matrix A_{CL}^L : The first three eigenvalues correspond to the zero-eigenvalue μ_1 of L ,

$$\lambda_1 = \lambda_2 = \lambda_3 = 0.$$

The remaining $3(N - 1)$ eigenvalues are determined from the $N - 1$ third-order polynomial factors arising from the determinant in Eq. (2.7). Explicitly, these remaining eigenvalues are the solutions to

$$\lambda^3 + k_D\mu_i\lambda^2 + k_P\mu_i\lambda + k_I\mu_i = 0, \quad i = 2, 3, \dots, N. \quad (2.8)$$

In Chapter 1, Sec. 1.2.3, we proved that the PD consensus controller was asymptotically stable if and only if $k_P, k_D > 0$ and the undirected communication graph for the system has a spanning tree (a condition on gains was also found for *directed* graphs). From Theorem 1, this latter condition is equivalent to the statement that the graph Laplacian has exactly 1 zero-eigenvalue; from the Geršgorin's circle theorem, the remaining eigenvalues μ_i are in the right-half complex plane. We found that the closed-loop system matrix had eigenvalues

$$\lambda^2 + k_D\mu_i\lambda + k_P\mu_i = 0, \quad i = 1, 2, 3, \dots, N,$$

providing exactly two zero-eigenvalues and the remaining $2(N - 1)$ eigenvalues in the left-half complex plane. The fact that the real part of these non-zero eigenvalues are negative led to the asymptotic form of the STM given in Eq. (1.52). Therefore, an equivalent condition for the asymptotic stability of PD consensus controller is that the closed-loop system matrix (A_{CL}^G or A_{CL}^L) has exactly two zero-eigenvalues and all other eigenvalues in the left-half complex plane.

Analogously, we find that the PID consensus controller is stable if and only if the closed-loop system matrix has exactly *three* zero-eigenvalues, with all other eigenvalues in the left-half plane. The proof of this statement is essentially identical to the proof in Chapter 1 for the case of PD consensus controllers. We will omit this proof here since, in the next section, we will prove stability for fractional PID^α consensus controllers; these reduce to integer-order PID consensus controllers as a special case.

For PID consensus controllers, we therefore require that the $3(N - 1)$ eigenvalues $\lambda_4, \dots, \lambda_{3N}$ determined from Eq. (2.8) have negative real part for the consensus controller to be stable. In contrast to the PD case where the characteristic equation for A_{CL} factored into quadratic equations of λ , the solution to this cubic equation does not provide us with a simple form for the general conditions on the gains $\{k_P, k_I, k_D\}$. Rather, the conditions on the gains which lead to stability of the consensus space take the form of 3D hypersurfaces; for any given choice of k_P and k_D there is some maximum value of k_I , above which the controlled system becomes unstable. These hypersurfaces also depend on the communication topology through the L -eigenvalues μ_i . For a given communication topology each non-zero eigenvalue μ_i will be associated with a hypersurface bounding the stability region; the stability region for the entire system will then be the intersection of the regions corresponding to each μ_i individually.

We can determine, for a given μ_i , the hypersurface bounding the stability region using the Routh test on the characteristic equation Eq. (2.8). Equivalently, we can simply “trace” the border of the stability region by placing λ on the imaginary axis, $\lambda = i\rho$, for $\rho \in \mathbb{R}$. Substituting this into the above characteristic equation provides two conditions corresponding to the real and imaginary parts of the left-hand side. These two conditions can be solved to find the border of the stability region. For real μ_i , we find

$$\left. \begin{array}{l} i(\rho\mu_i k_P - \rho^3) = 0 \\ \mu_i k_I - \rho^2 \mu_i k_D = 0 \end{array} \right\} \Rightarrow k_I = \mu_i k_P k_D. \quad (2.9)$$

It can easily be shown that this is an upper limit on k_I for stability. We therefore find that for each L -eigenvalue μ_i , $i = 2, 3, \dots, N$, the stability condition on the gains is $k_I < \mu_i k_P k_D$. The hypersurface corresponding to this region of stability is shown in the left panel of Fig. 2.1.

For complex $\mu_i = a + ib$, the stability region is similarly found. Substituting $\lambda = i\rho$ into Eq. (2.8) provides the following two constraints from the resulting real and imaginary parts:

$$\begin{aligned} ak_D\rho^2 + bk_P\rho - ak_I &= 0 \\ \rho^3 + bk_D\rho^2 - ak_P\rho - bk_I &= 0. \end{aligned}$$

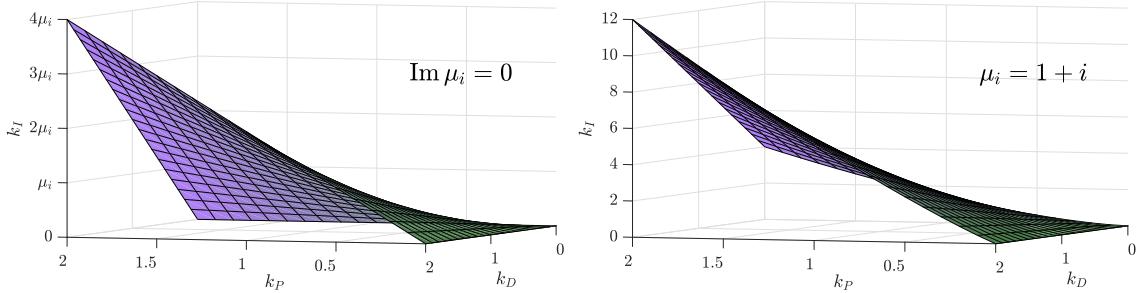


Figure 2.1: Hypersurfaces bounding the stability regions in $\{k_P, k_I, k_D\}$ parameter space for a N -agent system of double-integrators under PID consensus control. Regions below the surfaces are stable. These surfaces correspond to the non-zero Laplacian matrix eigenvalue μ_i , with $\mu_i \in \mathbb{R}$ (left) and $\mu_i = 1 + i$ (right). Note that the actual stability region for the system is the intersection of the regions corresponding to $\mu_2, \mu_3, \dots, \mu_N$.

Noting that $a > 0$ for all communication topologies from Geršgorin's circle theorem, and $|b| > 0$ from our assumption that μ_i is complex, these can be rewritten as

$$k_D\rho^2 + \frac{b}{a}k_P\rho - k_I = 0 \quad (2.10a)$$

$$\frac{1}{b}\rho^3 + k_D\rho^2 - \frac{a}{b}k_P\rho - k_I = 0. \quad (2.10b)$$

Equating the left-hand sides of these two equations provides

$$\rho = \sqrt{k_P \left(a + \frac{b^2}{a} \right)},$$

which is re-substituted into Eq. (2.10a) to provide the condition

$$k_I < \left(a + \frac{b^2}{a} \right) k_D k_P + \left(\frac{b}{a} \sqrt{a + \frac{b^2}{a}} \right) k_P^{3/2}. \quad (2.11)$$

Note that this reproduces the condition in Eq. (2.9) when $b \rightarrow 0$. The hypersurface bounding this stability region is shown in the right panel of Fig. 2.1.

The region of $\{k_P, k_I, k_D\}$ parameter space giving rise to stable closed-loop dynamics is now given by the intersection of the regions bounded by the hypersurfaces corresponding to $\mu_2, \mu_3, \dots, \mu_N$. We therefore have the following result:

Theorem 3. Given an N -agent double-integrator system, whose communication topology contains a spanning tree, the PID consensus control law Eq. (2.1) brings the system to consensus asymptotically if the gains $\{k_P, k_I, k_D\}$ satisfy the stability condition

$$k_I < \min_i \left\{ \left(a_i + \frac{b_i^2}{a_i} \right) k_D k_P + \left(\frac{b_i}{a_i} \sqrt{a_i + \frac{b_i^2}{a_i}} \right) k_P^{3/2} \right\}, \quad (2.12)$$

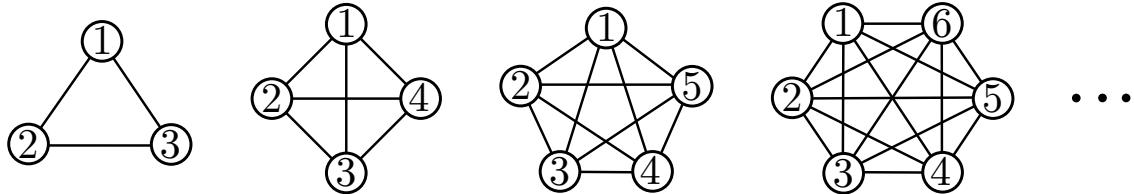
where $a_i = \operatorname{Re} \mu_i$, $b_i = \operatorname{Im} \mu_i$, and where μ_i , $i = 2, 3, \dots, N$, are the non-zero eigenvalues of the Laplacian matrix L .

In cases where all non-zero eigenvalues of L are real, this stability region takes a simple form. Let us order the eigenvalues of L in increasing order, such that

$$\mu_1 = 0 < \mu_2 \leq \mu_3 \leq \dots \leq \mu_N,$$

so that the smallest non-zero eigenvalue, which we call the *Fieldler* eigenvalue,² is denoted μ_2 . Since this gives the tightest constraint on the integral gain k_I , the intersection of the stability regions bounded by the hypersurfaces corresponding to μ_i , $i = 2, 3, \dots, N$, is simply the region bounded by the μ_2 -hypersurface.

An important example of such a case is when the N -agent system is fully connected, so that the communication topology is described by complete graphs $\mathcal{G} = K_N$:



For these systems, the eigenvalues of the graph Laplacian L are given by

$$\mu_i = \begin{cases} 0 & i = 1, \\ N & i = 2, \dots, N. \end{cases} \quad (2.13)$$

²Named in reference to Miroslav Fiedler (1926–2015), a Czech mathematician who pioneered, among other things, the study of algebraic graph theory.

This can be seen from the identity $\det(aI_N + B) = \prod_i(a + b_i)$, with b_i the eigenvalues of B , and noting that complete graphs have Laplacian $L(K_N) = NI_N - \mathbf{1}_{N \times N}$ with $\mathbf{1}_{N \times N}$ the array of ones. In this case we see that the Fiedler eigenvalue $\mu_2 = N$ is $(N - 1)$ -degenerate. The eigenvalues of the closed-loop system are then the roots of the characteristic equation

$$\begin{aligned} 0 &= \lambda^3 (\det(\lambda I_3 - A_3 + NB_3K_3))^{N-1} \\ &= \lambda^3 (\lambda^3 + Nk_D\lambda^2 + Nk_P\lambda + Nk_I)^{N-1}. \end{aligned} \quad (2.14)$$

That is, the closed-loop system has three zero-eigenvalues in addition to a triplet of eigenvalues, each of which has degeneracy $N - 1$. All of these eigenvalues will have negative real-part if

$$k_I < Nk_Pk_D. \quad (2.15)$$

That is, a fully-connected N -agent system of double-integrators, the PID consensus control protocol Eq. (2.1) will asymptotically bring the system to consensus if $k_I < Nk_Pk_D$.

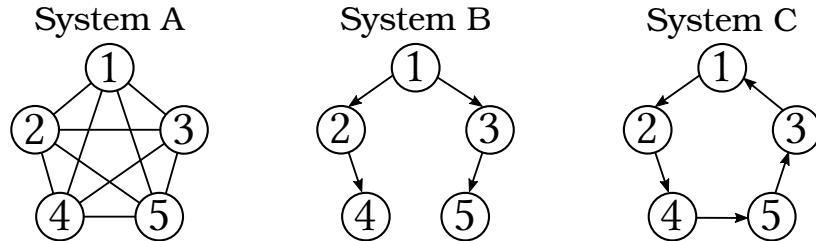


Figure 2.2: Communication graphs for three benchmark 5-agent systems. System A is an undirected complete graph, describing a fully connected topology. System B is a directed graph (digraph) representing a leader-follower communication topology. System C is a 5-cycle graph representing a cyclic communication topology.

To illustrate the above results, we will numerically simulate the trajectories three benchmark systems, with communication graphs illustrated in Fig. 2.2. We will take the weights of all edges in these graphs equal to unity. The Laplacian matrices corresponding to

	λ_1	λ_2	λ_3	λ_4	λ_5
System A	0	5	5	5	5
System B	0	1	1	1	1
System C	0	$0.691 + 0.951i$	$0.691 - 0.951i$	$1.809 + 0.588i$	$1.809 - 0.588i$

Table 2.1: Laplacian eigenvalues for the three benchmark systems illustrated in Fig. 2.2

these systems are given by

$$L_A = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{bmatrix}, \quad L_B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}, \quad L_C = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad (2.16)$$

and their corresponding eigenvalues listed in Table 2.1. For the following simulations we will set $k_P = 1$, $k_D = 2$, and we will initialize each of the five agents at rest and at the positions $\mathbf{r}_0 = [1; 2; 4; 7; 9]$. According to Eq. (2.12), we find the following stability conditions on the integral gains:

$$\text{System A: } k_I < 10$$

$$\text{System B: } k_I < 2$$

$$\text{System C: } k_I < 2.054$$

To illustrate these stability conditions we simulate the systems using values for k_I above, below, and exactly at the critical values for stability. We plot these responses for System A in Fig. 2.3. As illustrated by the central plot, the critical value of $k_I = 10$ gives rise to a marginally stable system; below and above this value of gain we see stable and unstable behavior, respectively. In addition, since this system is fully connected and all weights are set to unity, we expect the consensus value to be the simple average of initial conditions. (This was shown in the case of PD consensus control and given in Eq. (1.53); the result for PID consensus control is identical as will be shown in the following section.) We therefore

expect the consensus value to be $\bar{r} = \sum_i^N r_i(0)/N = 4.6$. This consensus value is confirmed through simulation as shown in the left panel.

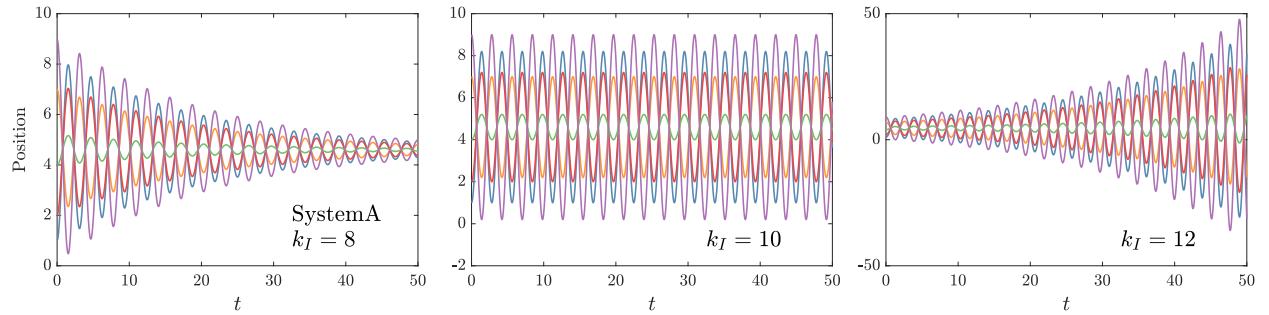


Figure 2.3: Simulation of System A under PID consensus control. From left to right, we show values of k_I corresponding to stable, marginally stable, and unstable systems as determined from Eq. (2.12).

We also simulate System B at, above, and below the critical value of $k_I = 2$. In this case, the response of the system is rather slow, and a much longer simulation time window is needed to illustrate the criticality point for k_I . With this longer time window we will be able to see stable and unstable behavior even in cases when k_I is only minutely below and above k_I . We therefore simulate the system for $k_I = \{1.8, 2, 2.2\}$. These responses are shown in Fig. 2.4. We again see that the critical value $k_I = 2$ as found in Eqs. (2.17) gives marginal stability, since the growth in oscillations of agents 4 and 5 is linear. Above this gain value the oscillations grow exponentially as expected for unstable systems. Below this threshold, the oscillations for agents 4 and 5 initially increase, but subsequently become smaller and straddle the consensus value. In this case, since agent 1 is the leader (it does not receive information from other agents), there is no control signal affecting its trajectory. The consensus value in this case is just the value of the leader, and since initial velocity of the leader is zero this consensus value is $\bar{r} = r_i(0) = 1$.

Finally, we show the response for the cyclical topology of System C in Fig. 2.5. These results again confirm the stability thresholds of Eq. (2.17). In this case we notice interesting transient behavior as the system is first set into motion.

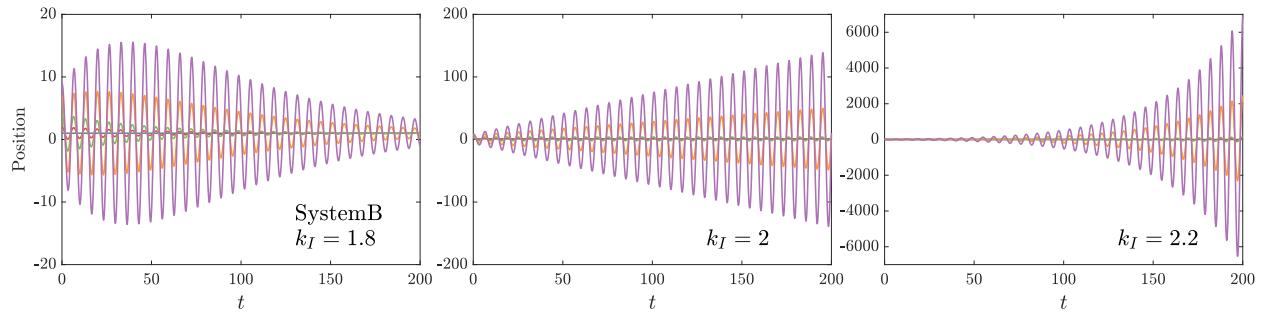


Figure 2.4: Same as Fig. 2.3, but simulating response of System B. We simulate for a much longer time window due to the slower response of this system. These plots also confirm the stability conditions on the PID gains.

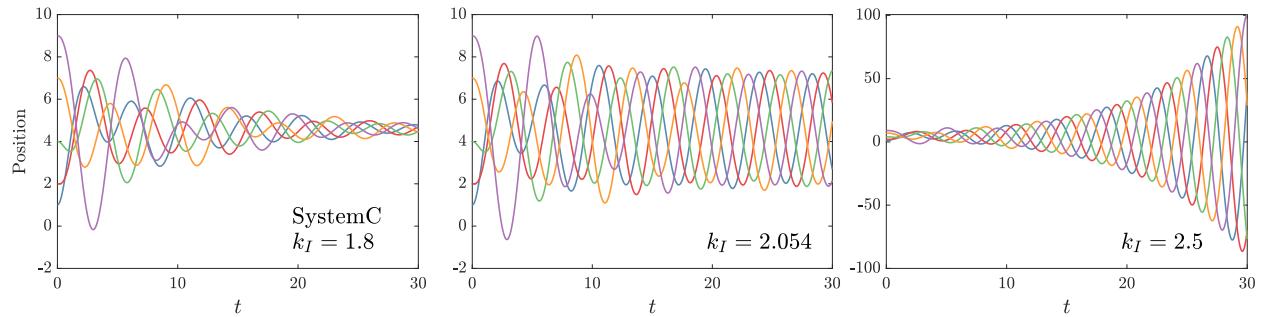


Figure 2.5: Same as Figs. 2.3 and 2.4, but simulating response of System C.

2.2 FRACTIONAL PID CONSENSUS CONTROL

Our present goal is to introduce fractional control into the consensus control protocols discussed in the previous section and in Chapter 1. To this end, let us rewrite the integer-order PID consensus protocol in Eq. (2.1) as

$$u_i = \sum_{j=1}^N a_{ij} \left(k_I D^0(\xi_i - \xi_j) + k_P D^1(\xi_i - \xi_j) + k_D D^2(\xi_i - \xi_j) \right), \quad i = 1, 2, \dots, N.$$

Recall that $\xi = \int_0^t r(\tau) d\tau$, so that the first, second, and third terms within the sum describe the time-integral differences, the position differences, and velocity differences, respectively, between agent i and the other agents j . In this operator-notation form, the generalization from integer-order to fractional-order PID control becomes obvious: the general fractional-

order PID-type consensus protocol for the i^{th} agent is given by

$$u_i = \sum_{j=1}^N a_{ij} \left(\tilde{k}_0 D^{\alpha_0}(\xi_j - \xi_i) + \tilde{k}_1 D^{\alpha_1}(\xi_j - \xi_i) + \cdots + \tilde{k}_{\tilde{m}} D^{\alpha_{\tilde{m}}}(\xi_j - \xi_i) \right) \quad i = 1, 2, \dots, N, \quad (2.18)$$

where $\tilde{k}_i > 0$ and $0 \leq \alpha_i < 3$ for all i . Terms within the sum for which $0 \leq \alpha_k < 1$ correspond to integral-type terms (since these are less-than-unity-order fractional derivatives of the integral of position differences) and terms with $1 < \alpha_k < 3$ are referred to as derivative-type. Proportional-type terms simply refer to the terms where $\alpha_k = 1$.

Note that we recover the integer-order PID consensus controller, Eq. (2.1), when $\tilde{m} = 2$, $\alpha_0 = 0$, $\alpha_1 = 1$, and $\alpha_2 = 2$. This is the reason that we refer to this controller as “PID-type,” though in general this controller need not contain a P-, I-, or D-type term. In what follows, however, we will always assume a proportional term exists, so that $1 \in \{\alpha_0, \alpha_1, \dots, \alpha_{\tilde{m}}\}$. In these cases we will refer to controllers containing both fractional integral *and* fractional derivative terms as $\text{PI}^\mu\text{D}^\nu$ consensus controllers, and those containing integer-order integral and fractional-order derivative terms as PID^ν consensus controllers.

2.2.1 Equations of Motion in Terms of the Fractional Pseudostate

In what follows we will assume that the fractional derivative orders in Eq. (2.18) are rational, $\alpha_i \in \mathbb{Q}$, $\forall i$. In this case, these fractional orders can be written $\alpha_i = \nu_i/\delta_i$ with $\nu_i, \delta_i \in \mathbb{N}$. Defining δ to be the least common multiple (LCM) of the set $\{\delta_1, \delta_2, \dots, \delta_{\tilde{m}}\}$, we can write $\alpha_i = n_i/\delta = n_i\alpha$. It is therefore always possible to rewrite Eq. (2.18) in the form

$$u_i = \sum_{j=1}^N a_{ij} \left(k_0(\xi_j - \xi_i) + k_1 D^\alpha(\xi_j - \xi_i) + k_2 D^{2\alpha}(\xi_j - \xi_i) + \cdots + k_m D^{m\alpha}(\xi_j - \xi_i) \right), \quad (2.19)$$

where $\alpha = 1/\delta$ is the *fractional order* of the controller (note the slight change in notation from when we introduced fractional state-space in Sec. 1.3.3). Together with the dynamical equation $D^3\xi_i = u_i$, Eq. (2.19) describes a *commensurate order* fractional system. We note that the number of non-zero terms does not change when rewriting Eq. (2.18) as Eq. (2.19); each term “ D^{α_i} ” in the former is written as a “ $D^{n_i\alpha}$ ” in the latter. Although in general $m \neq \tilde{m}$, the coefficients k_i will only be non-zero for terms which correspond to the α_i -order terms in Eq. (2.18).

It is now possible to write the equations of motion for a multiagent system in a form analogous to the state equation Eq. (2.2). We first consider each agent individually. Let us define the *pseudostate vector* for agent i to be

$$\begin{aligned}\mathbf{x}_i &= [\xi_i; D^\alpha \xi_i; D^{2\alpha} \xi_i; \dots; D^{(n-1)\alpha} \xi_i] \\ &= [\xi_i \ D^\alpha \xi_i \ D^{2\alpha} \xi_i \ \dots \ D^{1-\alpha} \xi_i \ ; \ r_i; D^\alpha r_i; \dots; D^{1-\alpha} r_i \ ; \ v_i; D^\alpha v_i; \dots; D^{1-\alpha} v_i].\end{aligned}\quad (2.20)$$

Note that in the second line we have rewritten the pseudostate in terms of $\dot{\xi}_i$, $r_i = \dot{\xi}_i$, and $v_i = \dot{r}_i$ so that all derivatives have fractional order between 0 and 1. The number of pseudostate components n is defined such that the acceleration of the point-mass agent is $\ddot{r}_i = D^3 \xi_i = D^{n\alpha} \xi_i$, which implies that $n = 3/\alpha$. Note that since $\alpha = 1/\delta$ and $\delta \in \mathbb{N}$, the number of pseudostate components is guaranteed to be an integer. With this definition, the equations of motion for the i^{th} point-mass agent in terms of the pseudostate \mathbf{x}_i is given by the controllable canonical form

$$D^\alpha \mathbf{x}_i = A_n \mathbf{x}_i + B_n u_i, \quad \text{with} \quad A_n = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \\ \vdots & \ddots & \ddots & \ddots & 0 \\ & & & 0 & 1 \\ 0 & \cdots & & & 0 \end{bmatrix}}_{n \times n}, \quad B_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad (2.21)$$

analogous to the form of the state-equation for integer-order PID-controlled systems.³ The bottom row of this matrix equation gives

$$D^\alpha [D^{(n-1)\alpha} \xi_i] = D^{n\alpha} \xi_i = D^3 \xi_i = u_i \quad \Rightarrow \quad \ddot{r}_i = u_i,$$

reproducing Newton's second law as desired. (Note that this result relies on the fact that the Caputo definition for the derivative is being used, and that $0 < \alpha < 1$. [42])

Before proceeding, we note that in practical applications one is only able to access or measure the physical states of the system such as position and velocity. In state-space formalism this is typically accounted for by introducing *output equations* of the form

³The validity of pseudostate differential equations of this form — for $0 < \alpha < 1$ and utilizing the Caputo derivative — is laid out in [42].

$\mathbf{y}_i = C\mathbf{x}_i + Du_i$; vanishing elements of the output matrix C reflect the fact that certain state variables are not observed and therefore cannot be used in the control law. This seemingly inhibits the use of full-pseudostate feedback. A common solution in standard integer-order systems is to use an *observer* to reconstruct the full state of the system so that this information can be used in the control law. Methods of determining the fractional-order pseudostates using fractional observers have been developed and remain an active area of research [43, 44]. By combining fractional-order observers with the proposed consensus control, the full pseudostate can be used for feedback. Although this is an important practical consideration, we will not discuss this further; for the remainder of this work we will assume that the full fractional state is reconstructed (either through the use of such observers or through numerical computation of these fractional states by some “onboard” system) so that these controllers can be implemented.

We note here that defining $1/\alpha = \delta = \text{LCM}[\delta_1, \delta_2, \dots, \delta_{\tilde{m}}]$ does not always lead to a *minimal* representation. To illustrate, let us consider a single point-mass agent under fractional feedback control of the form

$$u = -D^{3/4}\xi - D^{3/2}\xi$$

(so that this corresponds to a fractional $P^0 I^{\frac{1}{4}} D^{\frac{1}{2}}$ controller). Taking the LCM of the denominators of these fractional orders gives $1/\alpha = \delta = 4$, so that our pseudostate comprises $n = 3\delta = 12$ states and our state-equation is of fractional order $\alpha = 1/4$. However, we can equivalently characterize our system by a state equation of fractional order $\bar{\alpha} = 3/4$ using the pseudostate $\mathbf{x} = [\xi \ D^{\bar{\alpha}}\xi \ D^{2\bar{\alpha}}\xi \ D^{3\bar{\alpha}}\xi]$. The state-equation is now of dimension 4, with bottom row given by

$$D^{\bar{\alpha}}[D^{3\bar{\alpha}}\xi] = D^{4\bar{\alpha}}\xi = u = -D^{\bar{\alpha}}\xi - D^{2\bar{\alpha}}\xi \quad \Rightarrow \quad D^3\xi = \ddot{r} = -D^{3/4}\xi - D^{3/2}\xi.$$

In general, the minimal representation corresponds to a fractional order

$$\bar{\alpha} = \frac{\text{GCD}[3, n_1, n_2, \dots, n_{\tilde{m}}]}{\text{LCM}[\delta_1, \delta_2, \dots, \delta_{\tilde{m}}]},$$

where $\text{GCD}[\cdot]$ represents the greatest common divisor. Throughout this work, however, we will continue to define the fractional order of the system as $1/\alpha = \text{LCM}[\delta_1, \dots, \delta_{\tilde{m}}]$, since

this gives the minimal representation whenever the controller involves a proportional (*i.e.*.. $D^1\xi$) term.

We now define the state vector of the entire N -agent system; as in Section 1.2.2, this can be done in two different ways, defining the *local* and *global* representations. In the local representation, the system state-vector is defined by concatenating the pseudostate vectors for each of the N agents,

$$\mathbf{X}_L = [\mathbf{x}_1^T \ \mathbf{x}_2^T \ \cdots \ \mathbf{x}_N^T]^T, \quad (2.22)$$

with \mathbf{x}_i defined in Eq. (2.20). In the global representation, the state-vector of the system is found by concatenating the individual pseudostate components of each agent: defining $\mathbf{z} = [\xi_1 \ \xi_2 \ \cdots \ \xi_N]^T$, the system state-vector is constructed as

$$\mathbf{X}_G = [\mathbf{z}^T \ D^\alpha \mathbf{z}^T \ \cdots \ D^{(n-1)\alpha} \mathbf{z}^T]^T. \quad (2.23)$$

The dimension of the total state vector in either representation is $nN \times 1$.

2.2.2 Closed-loop Equations of Motion

With the dynamics of each agent described by Eq. (2.21) and the control given by Eq. (2.19), it is straightforward to show that the closed-loop equations of motion for the system are given by

$$D^\alpha \mathbf{X}_G = A_{CL}^G \mathbf{X}_G = (A_n \otimes I_N - B_n K_n \otimes L) \mathbf{X}_G \quad (2.24)$$

in the global representation, and

$$D^\alpha \mathbf{X}_L = A_{CL}^L \mathbf{X}_L = (I_N \otimes A_n - L \otimes B_n K_n) \mathbf{X}_L \quad (2.25)$$

in the local representation (*c.f.*, Eqs. (2.2) and (2.3)). Here, K_n is a row vector containing the gains of each sequential derivative term, as will be illustrated below by example. Since \mathbf{X}_G is related to \mathbf{X}_L through permutation, we can easily transform between representations; that is,

$$\mathbf{X}_G = \Lambda \mathbf{X}_L \quad \Leftrightarrow \quad A_{CL}^L = \Lambda^T A_{CL}^G \Lambda,$$

where Λ is a (orthogonal) permutation matrix (*i.e.*, $\det\Lambda = \pm 1$ and $\Lambda\Lambda^T = I$). From the left equality, it is easy to see that the explicit form of the permutation matrix is

$$\Lambda = \begin{bmatrix} I_N \otimes P_0 \\ I_N \otimes P_1 \\ \vdots \\ I_N \otimes P_{n-1} \end{bmatrix}, \quad (2.26)$$

where P_j is the $1 \times n$ projection operator which projects out the j^{th} pseudostate, $P_j \mathbf{x}_i = D^{j\alpha} \xi_i$.

Illustrative Example: 3-agent system under $\mathbf{PI}^{1/2}\mathbf{D}^{1/2}$ control

For this system, the integral control of order $1/2$ corresponds to a $D^{1-\alpha_I}(\xi_j - \xi_i) = D^{1/2}(\xi_j - \xi_i)$ term and derivative control of order $1/2$ corresponds to a $D^{1+\alpha_D}(\xi_j - \xi_i) = D^{3/2}(\xi_j - \xi_i)$ in the control law. We therefore have $1 - \alpha_I = \alpha_1 = n_1/\delta_1 = 1/2$, $\alpha_P = \alpha_2 = n_2/\delta_2 = 1/1$, and $1 + \alpha_D = \alpha_3 = n_3/\delta_3 = 3/2$, and the commensurate order of this system is $\alpha = 1/\text{LCM}[\delta_1, \delta_2, \delta_3] = 1/2$. The control law for the i^{th} agent in commensurate form is thus

$$u_i = \sum_{j=1}^3 a_{ij} \left(k_I D^{1/2}(\xi_j - \xi_i) + k_P D^1(\xi_j - \xi_i) + k_D D^{3/2}(\xi_j - \xi_i) \right) \quad (2.27)$$

$$= \sum_{j=1}^3 a_{ij} \left(k_I D^{p\alpha}(\xi_j - \xi_i) + k_P D^{\delta\alpha}(\xi_j - \xi_i) + k_D D^{q\alpha}(\xi_j - \xi_i) \right), \quad (2.28)$$

with $p = \alpha_1/\alpha = 1$ and $q = \alpha_3/\alpha = 3$. Since $\delta = 1/\alpha = 2$, there are $n = 3\delta = 6$ pseudostate components per agent. The 1×6 gain matrix K_n has k_I in the $(p+1)$ position, k_P in the $(\delta+1)$ position, k_D in the $(q+1)$ position, and zeros elsewhere:

$$K_n = [0 \ k_I \ k_P \ k_D \ 0 \ 0].$$

The closed-loop pseudostate equation in the global (Eq. (2.24)) and local (Eq. (2.25))

representations are given by

$$\begin{aligned}
 \text{(Global Rep.)}: \quad D^{1/3} \mathbf{X}_G &= \begin{bmatrix} \mathbf{0} & I_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I_3 \\ \mathbf{0} & -k_I L & -k_P L & -k_D L & \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{X}_G \\
 \text{(Local Rep.)}: \quad D^{1/3} \mathbf{X}_L &= \begin{bmatrix} L_{11}(A_n - B_n K_n) & -L_{12} B_n K_n & -L_{13} B_n K_n \\ -L_{21} B_n K_n & L_{22}(A_n - B_n K_n) & -L_{23} B_n K_n \\ -L_{31} B_n K_n & -L_{32} B_n K_n & L_{33}(A_n - B_n K_n) \end{bmatrix} \mathbf{X}_L
 \end{aligned}$$

The dimension of both of these representations is $nN = 18$. ■

It is important to realize that the dimension nN of the pseudostate equation (Eqs. (2.24) or (2.25)) is highly sensitive to the fractional order α , and can become quite large. To illustrate, let us take a specific case of an integer-order PI controller with fractional derivative control of order α_D . In the incommensurate form of Eq. (2.18), this corresponds to

$$u_i = \sum_{j=1}^N a_{ij} (k_I D^0(\xi_j - \xi_i) + k_P D^1(\xi_j - \xi_i) + k_D D^{1+\alpha_D}(\xi_j - \xi_i)). \quad (2.29)$$

If we take $\alpha_D = 0.5$, the (commensurate) fractional order of the system is $\alpha = \frac{1}{2}$ and the number of pseudostates (the dimension of \mathbf{X}) is $3N/\alpha = 6N$. If we instead take $\alpha_D = 0.51$, then the fractional order of the system is $\alpha = 1/100$ (so that $\alpha_D = 51\alpha$) and the dimension of the pseudostate equation is $300N$. In general, if there are p digits in the mantissa of α_D , then $\dim \mathbf{X} \leq 3 \times 10^p \times N$.

It therefore becomes apparent that the pseudostate-space formalism introduced above is ill-suited for certain practical applications such as numerical simulation of general fractional systems or fractional PID tuning. Nevertheless, the pseudostate-space form, which in principle is valid for any rational fractional orders α_i , allows one to efficiently study the stability properties of a (rationally-ordered) fractionally controlled system.

2.3 STABILITY OF PID $^\alpha$ CONSENSUS CONTROL

In the remainder of this work, we will restrict our attention to PID $^\alpha$ consensus control protocols of the form given in Eq. (2.29), and take $0 < \alpha_D < 1$. Using fractional derivative control gives an extra degree of freedom (fractional order α_D) which can be tuned to achieve more optimal system response characteristics. In this section we will prove that, under certain conditions, a multiagent system will asymptotically (in the Mittag-Leffler sense; see discussion after Eq. (1.66)) reach consensus in position and velocity for any initial conditions.

In what follows we will assume that the communication topology of the system has a spanning tree. In this case the graph Laplacian matrix L has a simple zero-eigenvalue μ_1 and, according to Geršgorin's circle theorem, the remaining $N - 1$ eigenvalues are in the right-half plane. We label these eigenvalues according to $|\mu_1| < |\mu_2| \leq \dots \leq |\mu_N|$.

2.3.1 Eigenanalysis of the PID $^\alpha$ Controlled System

Consider an N -body multiagent system under control protocol Eq. (2.29). Assuming $\alpha_D \in \mathbb{Q}$, we can represent this control protocol in commensurate form

$$\begin{aligned} u_i &= \sum_{j=1}^N a_{ij} (k_I(\xi_j - \xi_i) + k_P(r_j - r_i) + k_D D^{\alpha_D}(r_j - r_i)) \\ &= \sum_{j=1}^N a_{ij} (k_I(\xi_j - \xi_i) + k_P D^{\delta\alpha}(\xi_j - \xi_i) + k_D D^{q\alpha}(\xi_j - \xi_i)), \end{aligned} \tag{2.30}$$

where α is the (commensurate) fractional order of the system, $\delta = 1/\alpha$, $n = 3/\alpha$, and $\{k_I, k_P, k_D\} \geq 0$. In this case, since the derivative orders of the first and second terms are 0 and 1, the fractional order α is determined purely from α_D . Since this derivative order is rational we can write $\alpha_D = \nu_D/\delta_D$, so that $\alpha = 1/\delta_D$. Also, since we are assuming the fractional order $0 \leq \alpha_D \leq 1$, we have $\delta < q < 2\delta$.

The closed-loop dynamics are given in the local representation by Eq. (2.25). As in the integer-order case, since

$$\text{spec}\left\{I_N \otimes A_n - L \otimes B_n K_n\right\} = \text{spec}\left\{\text{diag}(A_n, A_n - \mu_2 B_n K_n, \dots, A_n - \mu_N B_n K_n)\right\},$$

the eigenvalues of the closed-loop system can be found by solving the characteristic equation for the block-diagonal matrix on the right-hand side. This is given by

$$0 = \det(\lambda I_n - A_n) \prod_{i=2}^N \det(\lambda I_n - A_n + \mu_i B_n K_n) \quad (2.31)$$

$$= \lambda^n \prod_{i=2}^N \det(\lambda I_n - A_n + \mu_i B_n K_n). \quad (2.32)$$

We therefore immediately see that the simple zero-eigenvalue of the graph Laplacian corresponds to n zero-eigenvalues of the closed-loop system. The remaining eigenvalues can be found by solving for the roots of the $N - 1$ separate n^{th} -order polynomials,

$$0 = \det(\lambda I_n - A_n + \mu_i B_n K_n) = k_I \mu_i + k_P \mu_i \lambda^\delta + k_D \mu_i \lambda^q + \lambda^n \quad i = 2, \dots, N,$$

each of which provides n additional eigenvalues; for $k_I > 0$, these additional eigenvalues are non-zero. Note that, since A_{CL}^G and A_{CL}^L are related through a similarity transform, they will share the same eigenvalues.

2.3.2 Proof of Stability for PID^α Controlled Systems

We will now prove that, given a PID^α -controlled multiagent system whose communication topology contains a spanning tree, consensus is reached asymptotically if all non-zero eigenvalues of the closed-loop system matrix A_{CL}^G satisfy $|\arg \lambda_i| > \alpha\pi/2$. This condition on the non-zero eigenvalues is an instance of Matignon's theorem [16] introduced in Sec. 1.3.3.

Consider the pseudostate equation for this system in the global representation, Eq. (2.24),

$$D^\alpha \mathbf{X}(t) = A_{CL}^G \mathbf{X}(t), \quad (2.33)$$

with $0 < \alpha \leq 1$ and with initial conditions $\mathbf{X}(0) \equiv \mathbf{X}_0$. Substituting into this equation the power series expansion

$$\mathbf{X}(t) = \mathbf{X}_0 + \mathbf{X}_1 t^\alpha + \mathbf{X}_2 t^{2\alpha} + \dots$$

and equating like powers of t , one finds ([13]) the *pseudostate transition matrix* (pSTM)

$\Phi_\alpha(t, 0)$ for this system as

$$\mathbf{X}(t) = \Phi_\alpha(t, 0)\mathbf{X}_0 = \left(I + \frac{A_{CL}^G t^\alpha}{\Gamma(1+\alpha)} + \frac{(A_{CL}^G)^2 t^{2\alpha}}{\Gamma(1+2\alpha)} + \dots \right) \mathbf{X}_0 \quad (2.34)$$

$$\equiv E_\alpha[A_{CL}^G t^\alpha] \mathbf{X}_0, \quad (2.35)$$

where $E_\alpha[M]$ is the *matrix* Mittag-Leffler function introduced in Eq. (1.72). Note that, since the ML function reduces to the exponential function as $\alpha \rightarrow 1$, the pSTM stands in direct analogy to the STM for integer order systems, $\Phi(t, 0) = \exp[A_{CL}^G t]$. Unlike the integer-order STM, however, $\Phi_\alpha(t, t')\Phi_\alpha(t', 0) \neq \Phi_\alpha(t, 0)$ since the ML function does not satisfy the semigroup property $f(x)f(y) = f(x+y)$. This is intimately connected to the fact that fractional derivatives are *non-local*, in that they depend on their initial conditions through the integration window within the definition of the Caputo derivative, Eq. (1.64).

As shown in the previous section, since we have integer-order integral control, A_{CL}^G has exactly n zero-eigenvalues. That is,

$$\text{eig } A_{CL}^G : \quad \begin{cases} \lambda_j = 0, & j = 1, \dots, n, \\ \lambda_j \neq 0, & j = n+1, \dots, nN. \end{cases}$$

We first show that the zero-eigenvalues of A_{CL}^G have geometric multiplicity of 1. Let us define

$$\mathbf{q} = [q_1^T \ q_2^T \ \cdots \ q_n^T]^T$$

to be the eigenvector associated with $\lambda = 0$, with $q_i \in \mathbb{R}^N$. From the explicit form of A_{CL}^G in Eq. (2.24), we have

$$A_{CL}^G \mathbf{q} = \begin{bmatrix} \mathbf{0} & I_N & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & & \vdots \\ \vdots & & \cdots & & I_N \\ -k_1 L & -k_2 L & \cdots & & -k_n L \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}.$$

From the upper rows of this block-matrix equation, we find that $q_i = 0$ for $i = 2, \dots, n$; from the last row we find that $-Lq_1 = 0$. Since L has a spanning tree, L has exactly one zero-eigenvalue $\mu = 0$. This in turn implies that the geometric multiplicity of $\mu = 0$ is 1,

and thus that q_1 is the only eigenvector of L associated with $\mu = 0$. Therefore, there is only a single linearly-independent eigenvector which satisfies $A_{CL}^G \mathbf{q} = \lambda \mathbf{q} = 0$, namely

$$\mathbf{q} = [q_1^T \quad \mathbf{0}_{1 \times N} \quad \cdots \quad \mathbf{0}_{1 \times N}]^T.$$

Thus the eigenvalue $\lambda = 0$ of A_{CL}^G has algebraic multiplicity of n and geometric multiplicity of 1.

We can now find the Jordan decomposition of A_{CL}^G as

$$A_{CL}^G = VJW = V \begin{bmatrix} J_0 & \mathbf{0}_{n \times \tilde{n}} \\ \mathbf{0}_{\tilde{n} \times n} & J' \end{bmatrix} W,$$

where $\tilde{n} = n(N - 1)$, and where J_0 is the n -dimensional Jordan block associated with $\lambda = 0$. The $\tilde{n} \times \tilde{n}$ sub-matrix J' comprises all Jordan blocks associated with the eigenvalues λ_i for $i = n + 1, \dots, nN$. In general, these eigenvalues can all be unique (*i.e.*, simple), in which case $J' = \text{diag}(\lambda_{n+1}, \lambda_{n+2}, \dots, \lambda_{nN})$. The matrices V and W facilitating the similarity transformation are

$$V = [v_1 \quad v_2 \quad \cdots \quad v_{nN}], \quad W = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_{nN}^T \end{bmatrix},$$

where $v_i \in \mathbb{R}^{nN \times 1}$ are the right (generalized) eigenvectors of A_{CL}^G and $w_i^T \in \mathbb{R}^{1 \times nN}$ are the left (generalized) eigenvectors of A_{CL}^G , chosen and normalized such that $WV = I_{nN}$. This implies that $w_i^T v_j = \delta_{ij}$ for all $i, j \in \{1, 2, \dots, nN\}$.

The matrix ML function applied to $A_{CL}^G t^\alpha$ is thus

$$E_\alpha[A_{CL}^G t^\alpha] = V \begin{bmatrix} E_\alpha[J_0 t^\alpha] & 0 \\ 0 & E_\alpha[J' t^\alpha] \end{bmatrix} W. \quad (2.36)$$

The series definition for $E_\alpha[J_0 t^\alpha]$ is easily computed since the matrix J_0 is nilpotent, *i.e.*, $J_0^m = 0$ for $m \geq n$ (recall that n is the dimension of J_0). Explicitly,

$$E_\alpha[J_0 t^\alpha] = \sum_{k=0}^{n-1} \frac{J_0^k t^{k\alpha}}{\Gamma[1 + k\alpha]} = \begin{bmatrix} \Theta_0 & \Theta_1 & \cdots & \Theta_{n-1} \\ 0 & \Theta_0 & \cdots & \Theta_{n-2} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \Theta_0 \end{bmatrix}, \quad (2.37)$$

where $\Theta_k = t^{k\alpha}/\Gamma[1 + k\alpha]$. Let us for now assume that the lower-right block-matrix of Eq. (2.36) vanishes asymptotically,

$$E_\alpha[J't^\alpha] \rightarrow \mathbf{0}_{\tilde{n} \times \tilde{n}} \quad \text{as } t \rightarrow \infty. \quad (2.38)$$

We will first show that, under this condition, positions and velocities for this multiagent system reach consensus asymptotically. Once this is shown, we will investigate the conditions under which Eq. (2.38) is true.

Given the assumption Eq. (2.38), the transition matrix for our system approaches

$$\Phi_\alpha(t, 0) = E_\alpha[A_{CL}^G t^\alpha] \rightarrow V \begin{bmatrix} E_\alpha[J_0 t^\alpha] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} W \quad \text{as } t \rightarrow \infty, \quad (2.39)$$

By inspection, the only eigenvectors which contribute to the pSTM asymptotically are those associated with the zero eigenvalues. Without loss of generality, we can choose these as

$$\begin{aligned} v_1 &= [\mathbf{1}_N^T \quad \mathbf{0}_N^T \quad \mathbf{0}_N^T \quad \cdots \quad \mathbf{0}_N^T]^T & w_1 &= [\mathbf{p}^T \quad \mathbf{0}_N^T \quad \mathbf{0}_N^T \quad \cdots \quad \mathbf{0}_N^T]^T \\ v_2 &= [\mathbf{0}_N^T \quad \mathbf{1}_N^T \quad \mathbf{0}_N^T \quad \cdots \quad \mathbf{0}_N^T]^T & \text{and} & \quad w_2 = [\mathbf{0}_N^T \quad \mathbf{p}^T \quad \mathbf{0}_N^T \quad \cdots \quad \mathbf{0}_N^T]^T \\ &\vdots & &\vdots \\ v_n &= [\mathbf{0}_N^T \quad \mathbf{0}_N^T \quad \mathbf{0}_N^T \quad \cdots \quad \mathbf{1}_N^T]^T & w_n &= [\mathbf{0}_N^T \quad \mathbf{0}_N^T \quad \mathbf{0}_N^T \quad \cdots \quad \mathbf{p}^T]^T \end{aligned}$$

where $\mathbf{1}_N$ and $\mathbf{0}_N$ represent the $N \times 1$ vector of ones and zeros, respectively, and where \mathbf{p} is the left-eigenvector of the graph Laplacian matrix L with eigenvalue 0, $\mathbf{p}^T L = 0$, normalized such that $\mathbf{p}^T \mathbf{1}_N = 1$.

Defining the submatrices $V_0 = [v_1 \quad \cdots \quad v_n]$ and $W_0 = [w_1 \quad \cdots \quad w_n]$ which contain respectively the right- and left-eigenvectors associated with $\lambda = 0$, we write

$$V = [V_0 \quad V_i] \quad \text{and} \quad W = \begin{bmatrix} W_0^T \\ W_i^T \end{bmatrix}. \quad (2.40)$$

Note that V_0 and W_0 have dimension $(nN \times n)$, while V_i and W_i (which contain the eigenvectors associated with the nonzero eigenvalues) have dimension $(nN \times n(N - 1))$.

In terms of these submatrices, Eq. (2.39) becomes

$$\Phi_\alpha(t, 0) \rightarrow V_0 E_\alpha(J_0 t^\alpha) W_0 = \begin{bmatrix} \Theta_0 \mathbf{P} & \Theta_1 \mathbf{P} & \cdots & \Theta_{n-1} \mathbf{P} \\ \mathbf{0} & \Theta_0 \mathbf{P} & \cdots & \Theta_{n-2} \mathbf{P} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \Theta_0 \mathbf{P} \end{bmatrix} = E_\alpha(J_0 t^\alpha) \otimes \mathbf{P} \quad \text{as } t \rightarrow \infty, \quad (2.41)$$

where we have defined the $N \times N$ matrix $\mathbf{P} = \mathbf{1}_N \mathbf{p}^T$.

This transition matrix is used to find the asymptotic behavior of the N -agent system starting from some initial condition vector \mathbf{X}_0 . Recall that in the global representation, the pseudostate system vector is defined as

$$\begin{aligned} \mathbf{X} &= [\mathbf{z}^T \ D^\alpha \mathbf{z}^T \ D^{2\alpha} \mathbf{z}^T \ \cdots \ D^{(n-1)\alpha} \mathbf{z}^T]^T \\ &= [\mathbf{z}^T \ D^\alpha \mathbf{z}^T \ D^{2\alpha} \mathbf{z}^T \ \cdots \ \mathbf{x}^T \ D^\alpha \mathbf{x}^T \ \cdots \ \mathbf{v}^T \ D^\alpha \mathbf{v}^T \ \cdots \ D^{1-\alpha} \mathbf{v}^T]^T. \end{aligned}$$

From the definition of the Caputo derivative Eq. (1.64), if $\beta \notin \mathbb{N}$ then $D^\beta f(0) = 0$ for any function $f(x) \in C^k[0, T]$ given $T > 0$ and $k > \beta$ (see, e.g., [42]). Additionally, since \mathbf{z} represents the integrals of the positions of the agents from time 0 to t , $\mathbf{z}_0 = \mathbf{0}$. Thus the only nonzero initial conditions are the initial positions and initial velocities of the N agents. These occupy the $\delta + 1$ and $2\delta + 1$ blocks of \mathbf{X} . Thus

$$\mathbf{X}_0 = [\underbrace{\mathbf{0} \ \cdots \ \mathbf{0}}_{\delta N \text{ zeros}} \ \mathbf{r}_0^T \ \underbrace{\mathbf{0} \ \cdots \ \mathbf{0}}_{(\delta-1)N \text{ zeros}} \ \mathbf{v}_0^T \ \underbrace{\mathbf{0} \ \cdots \ \mathbf{0}}_{(\delta-1)N \text{ zeros}}]^T.$$

Using the asymptotic pSTM of Eq. (2.41) we find that, globally, the N -agent system asymptotically approaches

$$\mathbf{X}(t) = \Phi_\alpha(t, 0) \mathbf{X}_0 \rightarrow \begin{bmatrix} \mathbf{P}\tilde{\Theta}_\delta \mathbf{r}_0 + \mathbf{P}\tilde{\Theta}_{2\delta} \mathbf{v}_0 \\ \mathbf{P}\tilde{\Theta}_{\delta-1} \mathbf{r}_0 + \mathbf{P}\tilde{\Theta}_{2\delta-1} \mathbf{v}_0 \\ \vdots \\ \mathbf{0}_N \end{bmatrix} \quad \text{as } t \rightarrow \infty, \quad (2.42)$$

where for brevity we have introduced the notation $\tilde{\Theta}_m$ which equals Θ_m for $m \geq 0$ and 0 for $m < 0$. The components of $\mathbf{X}(t)$ corresponding to position and velocity are given by

$$\begin{aligned} \mathbf{r}(t) &= \mathbf{P}\mathbf{r}_0 + \mathbf{P}\Theta_\delta \mathbf{v}_0 \quad \text{as } t \rightarrow \infty. \\ \mathbf{v}(t) &= \mathbf{P}\mathbf{v}_0 \end{aligned} \quad (2.43)$$

Since $\mathbf{P} = \mathbf{1}_N \mathbf{p}^T$ and $\Theta_\delta = t$, these asymptotic positions and velocities are

$$\begin{aligned}\mathbf{r}(t) &= \bar{r}\mathbf{1}_N + \bar{v}\mathbf{1}_N t && \text{as } t \rightarrow \infty, \\ \mathbf{v}(t) &= \bar{v}\mathbf{1}_N\end{aligned}\tag{2.44}$$

where $\bar{r} = \mathbf{p}^T \mathbf{r}_0$ represents a weighted average of the initial positions and $\bar{v} = \mathbf{p}^T \mathbf{v}_0$ represents a weighted average of initial velocities. The weighting is given by the components of \mathbf{p} , the left-eigenvector of L associated with $\mu_1 = 0$; this weighting therefore depends on the communication topology. For complete graphs, $\mathbf{p} = (1/N)\mathbf{1}_N$, so that \bar{r} and \bar{v} are simply the mean values of the initial positions and velocities, respectively. We therefore see that, as long as $E_\alpha[J't^\alpha] \rightarrow \mathbf{0}$ as $t \rightarrow 0$ (as assumed in Eq. (2.39)), the multiagent system under PID $^\alpha$ consensus control comes to exact consensus in position and velocity asymptotically.

Returning now to the assumption of Eq. (2.38), let us first consider the case of simple non-zero eigenvalues, so that $J' = \text{diag}(\lambda_{n+1}, \lambda_{n+2}, \dots, \lambda_{nN})$. In this case,

$$E_\alpha[J't^\alpha] = \text{diag}(E_\alpha(\lambda_{n+1}t^\alpha), E_\alpha(\lambda_{n+2}t^\alpha), \dots, E_\alpha(\lambda_{nN}t^\alpha)).\tag{2.45}$$

Since the ML function can be expanded asymptotically to give

$$E_\alpha(z) \rightarrow \frac{1}{\alpha} \exp(z^{1/\alpha}), \quad \text{as } |z| \rightarrow \infty$$

(see, *e.g.*, Theorem 3.1 of [31]), the diagonal elements of Eq. (2.45) approach

$$E_\alpha(\lambda_i t^\alpha) \rightarrow \frac{1}{\alpha} \exp(\lambda_i^{1/\alpha} t), \quad \text{as } t \rightarrow \infty.\tag{2.46}$$

This vanishes asymptotically when the real part of $\lambda_i^{1/\alpha}$ is negative,

$$\begin{aligned}0 > \text{Re}\left\{\lambda_i^{1/\alpha}\right\} &= \text{Re}\left\{\left[|\lambda_i| \exp\left(i \arg \lambda_i\right)\right]^{1/\alpha}\right\} \\ &= \text{Re}\left\{|\lambda_i|^{1/\alpha} \exp\left(i \frac{\arg \lambda_i}{\alpha}\right)\right\} \\ &= |\lambda_i|^{1/\alpha} \cos\left(\frac{\arg \lambda_i}{\alpha}\right),\end{aligned}$$

giving the argument condition on the non-zero eigenvalues as

$$|\arg \lambda_i| > \alpha \frac{\pi}{2}, \quad i = n+1, n+2, \dots, nN.\tag{2.47}$$

This condition is a particular instance of Matignon's theorem [16].

If one or more of the non-zero eigenvalues are degenerate, then J' will then contain Jordan blocks corresponding to these degenerate eigenvalues. The matrix ML function of a Jordan block corresponding to a degenerate eigenvalue λ_d has elements built from $E_\alpha(\lambda_d t^\alpha)$ and their integer order derivatives $\frac{d^m}{dt^m} E_\alpha(\lambda_d t^\alpha)$. From Eq. (2.46), if $E_\alpha(\lambda_d t^\alpha)$ vanishes asymptotically, so does its derivative. Therefore, $E_\alpha[J' t^\alpha]$ will vanish asymptotically if the non-zero eigenvalues satisfy Eq. (2.47), regardless of their geometric multiplicity.

Let us review what we have just shown in the preceding argument. We had assumed that the communication topology possessed a spanning tree, which implied that the graph Laplacian had a simple zero eigenvalue. We then showed that the closed-loop system of double integrators under fractional PID $^\nu$ consensus control — which was expressed in a pseudostate-space form having fractional order $\alpha = 1/\delta$ and dimension $nN = 3\delta N$ — then possessed exactly n zero-eigenvalues ($\lambda_i = 0$, $i = 1, \dots, n$), and $n(N-1)$ potentially non-zero eigenvalues (λ_i , $i = n+1, \dots, nN$). Under the condition that the latter set of eigenvalues were in fact non-zero and satisfied $|\arg \lambda_i| > \alpha\pi/2$, we then showed that the system asymptotically reached consensus, with positions and velocities given by Eqs. (2.43). Moreover, the speed at which consensus is reached is dependent on the vanishing of $E_\alpha[J' t^\alpha]$ in the lower-right block of the modal form of the pSTM, Eq.(2.36), and this block vanishes as

$$E_\alpha(-\rho t^\alpha). \quad (2.48)$$

We therefore say that the system is asymptotically stable in the Mittag-Leffler sense [30]. We have therefore proven the following:

Theorem 4. A multiagent system under the fractional PID $^\nu$ consensus protocol Eq. (2.30) will reach consensus asymptotically if and only if the closed-loop system matrix as defined in Eqs. (2.24) or (2.25) has exactly n zero-eigenvalues and all other $n(N-1)$ eigenvalues satisfying $|\arg \lambda| > \alpha\pi/2$. A necessary condition for asymptotic stability is that the communication graph must have a spanning tree.

2.3.3 Stability Conditions on PID^ν Controller Gains

The PID^ν consensus control protocol in commensurate form is given in Eq. (2.30), and repeated here for convenience:

$$u_i = \sum_{j=1}^N a_{ij} (k_I(\xi_j - \xi_i) + k_P D^{\delta\alpha}(\xi_j - \xi_i) + k_D D^{q\alpha}(\xi_j - \xi_i)).$$

Recall that we have the condition $\delta < q < 2\delta$, with $\delta, q \in \mathbb{N}$ in the above expression. The pseudostate equations Eqs. (2.24) and (2.25) are therefore of dimension $n = 3\delta$, and K_n is an n -dimensional row matrix with k_I in the first position, k_P in the $(\delta + 1)$ position, k_D in the $(q + 1)$ position, and zeros elsewhere.

Let us now assume that the communication topology is fully connected, so that the eigenvalues of the graph Laplacian are given by Eq. (2.13). In this case, the characteristic equation of the closed-loop system is given by

$$0 = \lambda^n \det(\lambda I_n - A_n + NB_n K_n)^{N-1} \quad (2.49)$$

$$= \lambda^n (\lambda^n + Nk_D \lambda^q + Nk_P \lambda^\delta + Nk_I)^{N-1}, \quad (2.50)$$

so that the closed loop system has n zero-eigenvalues, and n non-zero eigenvalues each with $(N - 1)$ -fold degeneracy. We can now find the regions in the $\{k_I, k_D, k_P\}$ -space which correspond to systems which reach consensus asymptotically.

As we have shown, the closed-loop system will be asymptotically stable if $|\arg(\lambda_i)| > \frac{\alpha\pi}{2}$ for $i = n + 1, \dots, nN$. We can therefore trace the stability region in $\{k_I, k_D, k_P\}$ -space by setting $\lambda = \rho e^{i\alpha\pi/2}$. The eigenvalues which must satisfy the stability requirement are found from

$$0 = \lambda^n + Nk_D \lambda^q + Nk_P \lambda^\delta + Nk_I \quad (2.51)$$

$$= \rho^n \exp(i \frac{n\alpha\pi}{2}) + Nk_D \rho^q \exp(i \frac{q\alpha\pi}{2}) + Nk_P \rho^\delta \exp(i \frac{\delta\alpha\pi}{2}) + Nk_I. \quad (2.52)$$

Since $n = 3\delta$ and $\delta = 1/\alpha$, this becomes

$$0 = \rho^n \exp(i \frac{3\pi}{2}) + Nk_D \rho^q \exp(i \frac{q\alpha\pi}{2}) + Nk_P \rho^\delta \exp(i \frac{\pi}{2}) + Nk_I \quad (2.53)$$

$$= -i \rho^n + Nk_D \rho^q \exp(i \frac{q\alpha\pi}{2}) + i Nk_P \rho^\delta + Nk_I. \quad (2.54)$$

Demanding that both real and imaginary contributions must separately satisfy this equation gives

$$k_D \rho^q \cos\left(\frac{q\alpha\pi}{2}\right) + k_I = 0, \quad (2.55a)$$

$$Nk_P \rho^\delta - \rho^n + Nk_D \rho^q \sin\left(\frac{q\alpha\pi}{2}\right) = 0. \quad (2.55b)$$

Eliminating ρ using Eq. (2.55a) provides the following condition for an eigenvalue on the border of the stability region:

$$Nk_P \left(\frac{-k_I}{k_D \cos\left(\frac{q\alpha\pi}{2}\right)} \right)^{\frac{\delta}{q}} - \left(\frac{-k_I}{k_D \cos\left(\frac{q\alpha\pi}{2}\right)} \right)^{\frac{3\delta}{q}} - Nk_I \tan\left(\frac{q\alpha\pi}{2}\right) = 0. \quad (2.56)$$

We therefore find that consensus will be reached asymptotically if

$$k_P > k_I \tan\left(\frac{q\alpha\pi}{2}\right) \left(\frac{-k_I}{k_D \cos\left(\frac{q\alpha\pi}{2}\right)} \right)^{-\frac{1}{q\alpha}} + \frac{1}{N} \left(\frac{-k_I}{k_D \cos\left(\frac{q\alpha\pi}{2}\right)} \right)^{\frac{2}{q\alpha}}. \quad (2.57)$$

We note that for standard integer-order PID control in this pseudostate form, $q = 2\delta = 2/\alpha$, and Eq. (2.57) reduces to

$$k_P > \frac{k_I}{Nk_D}, \quad (2.58)$$

which is exactly what we found in Eq. (2.15).

To illustrate these stability conditions on the system gains, we will simulate a 3-agent fully-connected system with $\alpha_D = 0.5$. This corresponds to a system with commensurate fractional order $\alpha = 0.5$ and thus $n = 3/\alpha = 6$ pseudostates per agent. As the system is fully connected, the closed-loop system matrix A_{CL} has 6 zero-eigenvalues and 6 non-zero eigenvalues each with degeneracy 2. If we choose $k_I = 2$ and $k_D = 1$, we find from Eq. (2.57) that the system should be asymptotically stable if $k_P > 1/3$. In Fig. 2.6 we show three choices of k_P which correspond to asymptotically stable behavior ($k_P = 1$), marginally stable behavior ($k_P = 1/3$), and unstable behavior ($k_P = 0.1$). We also show the eigenvalues of the system matrix A_{CL} in all three cases. We see that stable behavior indeed corresponds to the case $k_P = 1$ where the non-zero eigenvalues of A_{CL} satisfy $|\arg \lambda_i| > \alpha\pi/2$, and unstable behavior corresponds to $k_P = 0.1$ where this eigenvalue condition is violated.

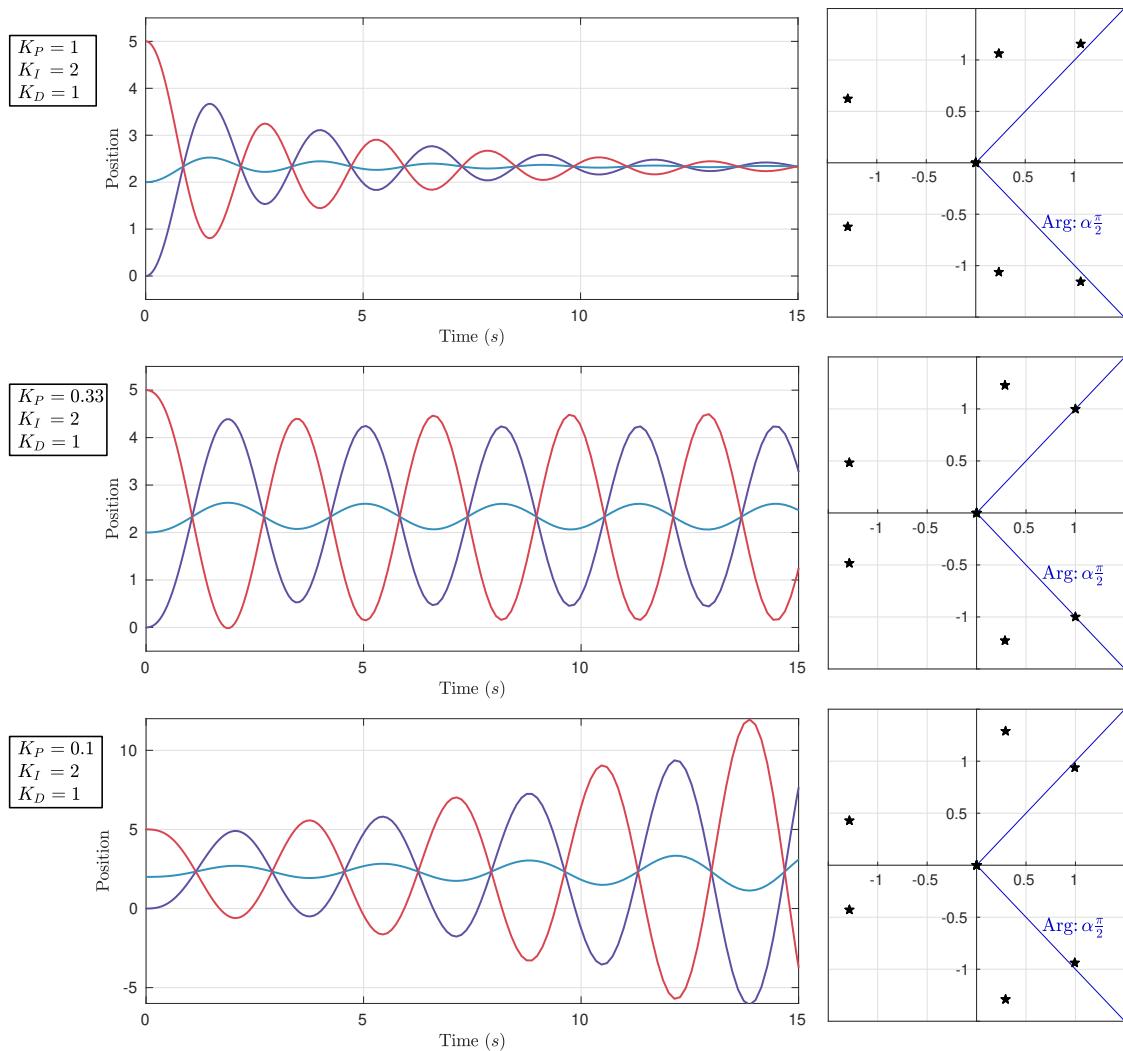


Figure 2.6: Illustrating the stability conditions on gains for a fully-connected system under fractional PID^α control. For this 3-agent system, we have taken derivative order $\alpha_D = 0.5$ and chosen $k_I = 2$ and $K_D = 1$. Using Eq. (2.57), we find that the system will be stable for $k_P > 0.33$. We show choices of k_P at, above, and below this stability limit. In the right panels we show the location of the eigenvalues of the system matrix A_{CL} corresponding to these choices of k_P . These plots agree with our result that the system is asymptotically stable if the non-zero eigenvalues of A_{CL} satisfy $|\arg \lambda_i| > \alpha \frac{\pi}{2}$.

2.4 NUMERICAL SIMULATIONS AND RESULTS

We now will report numerical simulation results of multiagent second-order systems under fractional control protocols of the form in Eq. (2.30). We first illustrate how the extra degree of freedom available from the fractional order α_D can generally lead to better controller performance by simulating the evolution of three benchmark systems and varying α_D . Lastly, we will show that the optimal choice of gains using the fractional PID $^\alpha$ controller outperforms the optimal choice of gains using the integer-order controller for the fully connected system, providing a concrete illustration of the advantages of fractional order controllers. All simulation results were generated using the fractional Chebychev collocation (FCC) method developed in [36, 38] and as described in Sec. 1.3.4.

2.4.1 Performance Comparisons Between PID and PID $^\alpha$ Controllers

We will now report the results of a survey over a subset of the parameter space which will illustrate the advantages of fractional PID $^\alpha$ controllers. We will again use the three benchmark 5-agent systems introduced in Sec. 2.1.1, with communication graphs illustrated in Fig. 2.2. The Laplacian matrices for these systems were given in Eq. (2.16), with corresponding eigenvalues given in Table 2.1.

Each of the five agents are assumed to have a mass of 1 kg, are initialized at rest at the positions $\mathbf{r}_0 = [1 \ 2 \ 4 \ 7 \ 9]^T$ (meters), and are controlled by the protocol of Eq. (2.30). Note that with these choices of units, the control signals will be in units of N/kg, which will be equal to forces in Newtons when applied to our 1 kg agents. We will set $k_P = 1$ universally, and survey over the remaining free parameters of the controller, $\{k_I, k_D, \alpha_D\}$. The consensus values in position $\bar{\mathbf{r}}$ are found from the left-eigenvector \mathbf{p} of the respective Laplacian matrices corresponding to the zero eigenvalue [5], $\bar{\mathbf{r}} = \mathbf{p}^T \mathbf{r}_0 / N$. For these three systems the consensus values are $\bar{r}_A = 4.6$ m, $\bar{r}_B = 1.0$ m, $\bar{r}_C = 4.6$ m. All numerical results were generated using the FCC method outlined in Sec. 1.3.4 and implemented in Matlab; results were periodically cross-checked against results obtained from the PECE method [45], showing good agreement.

We first survey over $\alpha_D = [0, 1]$ given $k_I = 1$ and k_D chosen such that the integer-order PID control gives stable behavior. For each point in this survey we compute settling time t_s and overshoot for the system. These performance specifications are determined as follows:

- **Settling time:** First we determine the average distance of the initial positions from the consensus value. We then compute the time it takes for each agent to fall and remain within 15% of this average initial distance. The agent which takes the longest to reach and remain within this window defines the settling time for the system.
- **Overshoot:** We first determine, when applicable, the time at which each agent crosses the consensus value. From this time forward, we determine the largest difference between the agent's position and the consensus value. Normalizing this difference to the initial difference between the position and the consensus value gives the percent overshoot for that agent. For over-damped systems for which the agent never crosses the consensus value, we set the overshoot for this agent to zero. The collective overshoot of the entire system is defined as the largest overshoot of all N agents.

The results of this survey are shown in Fig.2.7. We plot both settling time and overshoot as a function of fractional order α_D for each of the three systems. We also plot the trajectories of the five agents using the integer-order controller (red) and the fractional-order controller (blue), where in the latter case we have chosen an approximately optimal value from the survey over α_D .

In all three cases we see that performance — in terms of both settling time *and* overshoot — is optimized by taking the derivative to non-integer order. For System A the optimal fractional order is found to be $\alpha_D \approx 0.55$, giving a reduction in settling time of 41% (from 5.76 s to 3.35 s) and reduction in overshoot by 75% (from 42% to 11%). For System B the optimal fractional order is taken to be $\alpha_D \approx 0.6$ (though the optimal value depends on which performance specification we are interested in), giving a reduction in settling time of 75% (from ≈ 30 s to 7.17 s) and reduction in overshoot by 54% (from 83% to 38%). We notice that this system generally responds slower than the fully connected system, which is intuitively expected due to the fact that there are fewer edges in the communication topology. Finally,

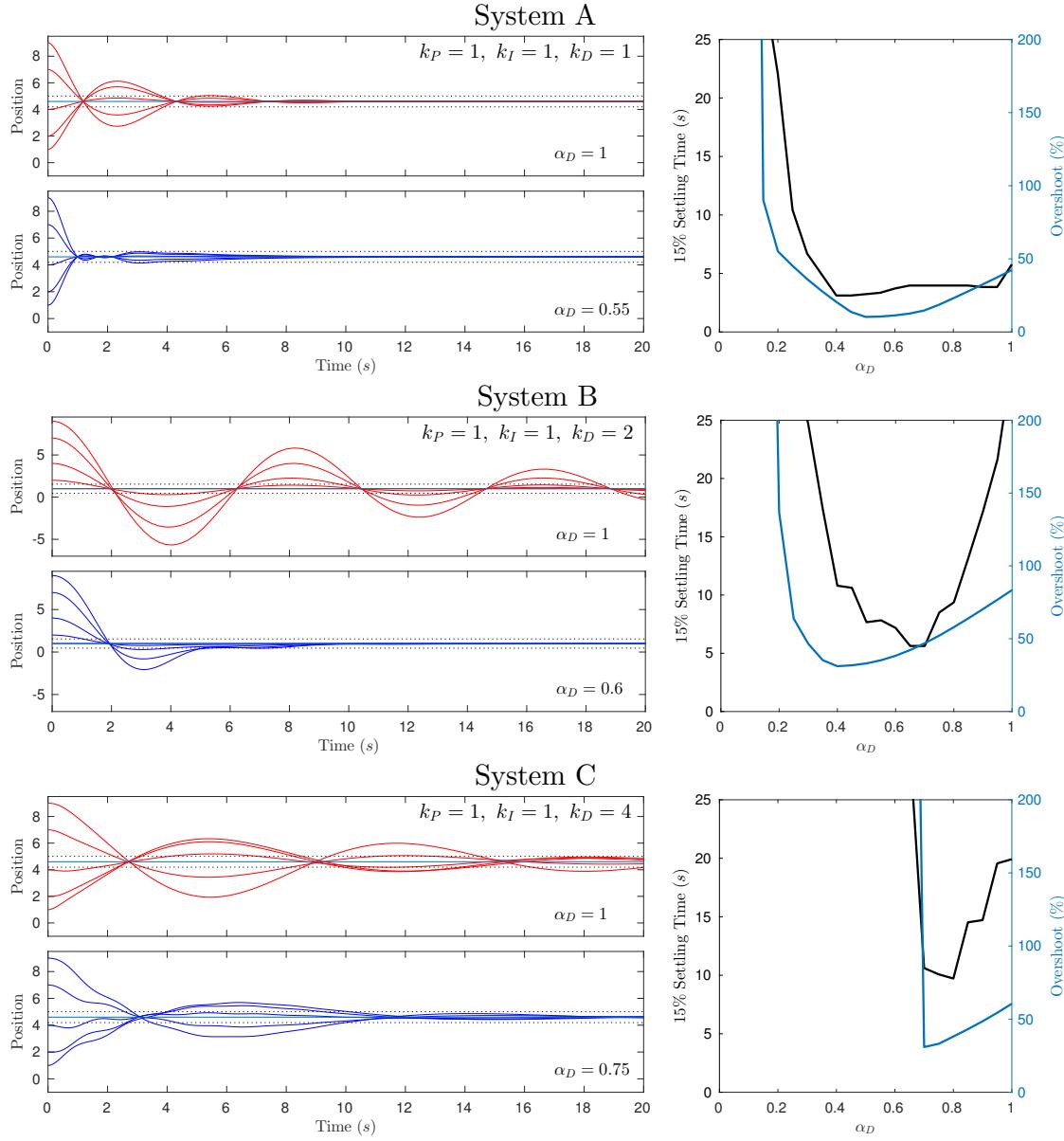


Figure 2.7: Comparisons between integer- and fractional order consensus for the three systems in Eq. (2.16). In the left panels we show the system response from an integer order PID controller in red and the response from a fractional order PID $^\alpha$ controller (using the approximately optimal choice of α_D) in blue. In the right panels we show the settling time and overshoot for the system as a function of fractional derivative order α_D .

for System C the optimal fractional order is found to be $\alpha_D \approx 0.75$, giving a reduction in settling time of 50% (from 20.0 s to 10.1 s) and reduction in overshoot by 46% (from 61%

to 33%).

For a more robust comparison between the fractional PID $^\nu$ and integer-order PID controller, we perform a survey over three of the four free parameters in our controller design, α_D , k_I , and k_D , keeping proportional gain constant at $k_P = 1$. We have simulated System A over a $(20 \times 20 \times 20)$ -dimensional grid in $\{k_D, k_I, \alpha_D\}$ parameter space, using 20 different values of α_D evenly spaced between 0 and 1. We can observe the effects of using fractional derivative control by plotting settling time for “slices” of $\{k_I, k_D\}$ parameter space along different values of fractional order α_D . The slice at $\alpha_D = 1$ represents the integer-order PID controller, and we find the optimal settling time to be $t_s = 1.067$ s at $k_I = 0.05$ and $k_D = 0.60$. Allowing α_D to vary, we find that we can reduce settling time for this system to a global minimum of $t_s = 0.862$ s by taking $\alpha_D = 0.75$, $k_I = 0.25$, and $k_D = 0.60$. This corresponds to a 19% reduction in settling time, which is a substantial improvement. We show the two relevant slices of parameter space as contour plots in Fig. 2.8.

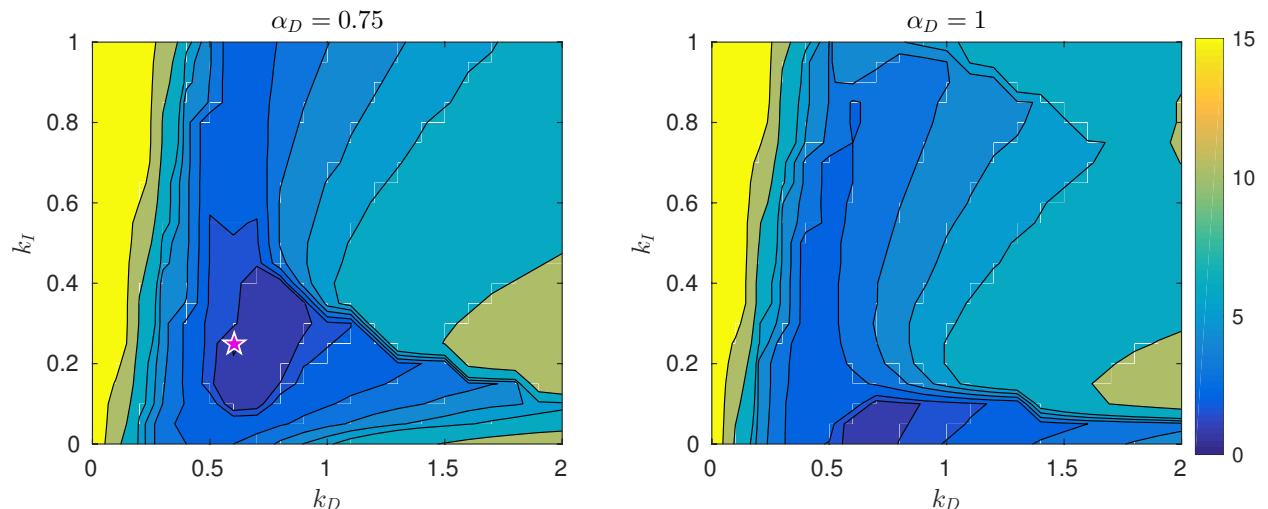


Figure 2.8: Contours of settling time as a function of k_I and k_D for System A and taking $k_P = 1$. At left we show the contours for the fractional PID $^\alpha$, taking $\alpha_D = 0.75$, and at right we show the same results using integer order controller. After surveying over α_D , we have found that the minimum settling time of $t_s = 0.862$ s is achieved with a fractional controller using $\alpha_D = 0.75$, $k_D = 0.6$, and $k_I = 0.25$; this point is marked with a purple star. The minimum settling time for the integer-order controller is $t_s = 1.067$ s.

We can also compare the fractional- and integer-order controllers with respect to combined performance specifications. One particularly useful combined measure for multivehicle control is the simultaneous optimization of settling time and integrated control effort (which in real-world systems would correspond to fuel cost). A straightforward measure of this combined performance is to compute the geometric mean of the two quantities,

$$J = \sqrt{U \cdot t_s}, \quad (2.59)$$

where integrated control effort U is computed as

$$U = \sum_{i=1}^N \int_0^T |u_i| dt.$$

This cost function is computed at all points in our survey grid; for our simulations we have taken $T = 30$ s. The relevant slices of parameter space for this survey are shown in Fig. 2.9.

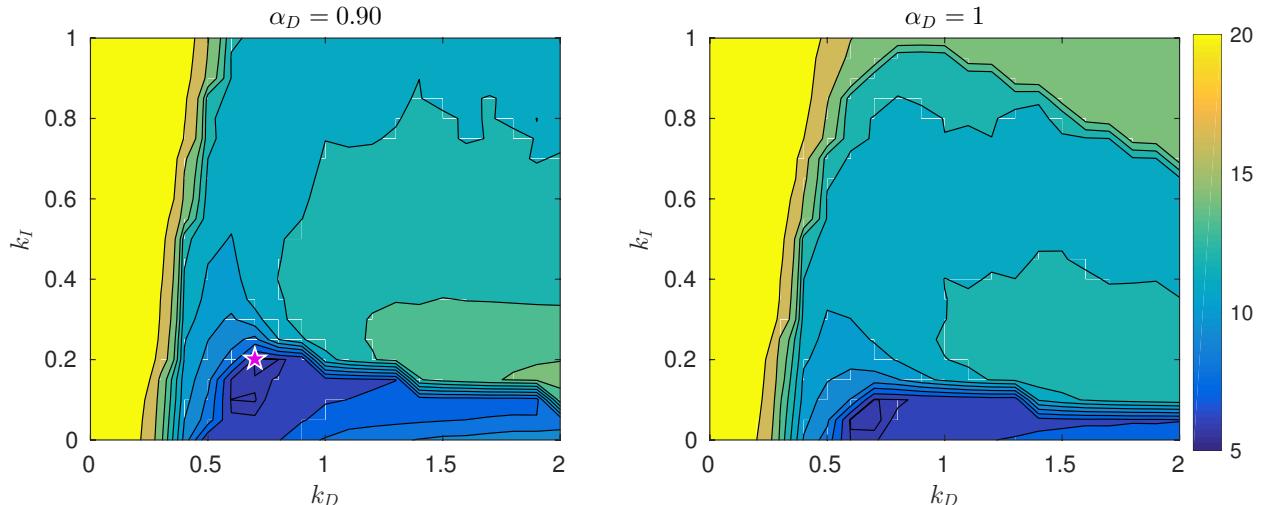


Figure 2.9: Same as Fig. 2.8, but showing contours of the geometric mean of settling time and integrated control effort. In this case we find the optimal value to be 5.78 using the fractional controller with $\alpha_D = 0.90$, $k_D = 0.7$, $k_I = 0.2$. The minimum value using the integer-order controller is 5.83.

For the integer-order controller, we find the optimal value to be $J = 5.83 \text{ N}^{1/2}\text{s}$ at $k_I = 0.05$ and $k_D = 0.70$. The global optimal value of J is again found for non-integer order α_D . We find this global optimum to be $J = 5.78 \text{ N}^{1/2}\text{s}$, at $\alpha = 0.90$, $k_I = 0.20$ and $k_D = 0.70$.

This corresponds to a slight reduction of 1%. Although the reduction is essentially negligible in this case, we take this as a proof of principle and note that cost function in Eq. (2.59) was constructed in a somewhat ad-hoc manner in order to eliminate the differences in scale between settling time and integrated cost.

By inspection of the left-panels of both Figs. 2.7 and 2.9 (where fractional derivative control is used) we find that performance is sub-optimal when integer gain k_I is taken to zero. As alluded to in Section 1.3.2, the increase in settling time that is observed as $k_I \rightarrow 0$ is caused by the “creeping” behavior illustrated in Fig. 1.5. Hence we see that integral control has the effect of mitigating this creeping behavior and bringing the system to consensus more rapidly. On the other hand, for integer-order control (right panels of Figs. 2.7 and 2.9) we find that optimal behavior (for these performance specifications) is achieved for $k_I \approx 0$. Nevertheless, integral control can have utility in integer-order controllers in cases where one needs more freedom in shaping the system’s response, or when mitigation of steady-state error is required.

The above simulation results provide a concrete demonstration of the benefits of using fractional PID $^\alpha$ consensus controllers for second-order multiagent systems. We note here that similar simulations have been performed for non-zero initial velocities, and similar results were obtained. Although this chapter considered motion in only one dimension, these results generalize easily to three-dimensional double integrators as long as the rotational dynamics do not lead to coupling of the individual spacial components of each agent.

CHAPTER 3

RELATIVE ORBIT CONTROL

This chapter discusses control strategies for systems of multiple spacecraft orbiting about a common celestial body. Each spacecraft obeys second-order dynamics given by the Newtonian gravitational equations of motion. We will treat our spacecraft as point-mass bodies so that the dynamics can be described by three-dimensional double-integrator systems with additional acceleration terms arising from the gravitational potential of the central body. The machinery laid out in Chapter 1 can therefore be applied here to design and analyze spacecraft control systems. Furthermore, the developments introduced in Chapter 2 can be appropriately generalized. We will first discuss relative orbit control in cases where the dynamics of relative spacecraft motion can be linearized. We will then explore how one can build fractional controllers for purposes of rendezvous. Lastly, we will show how consensus controllers can be used for efficiently arranging spacecraft in stable formations. This chapter assumes familiarity with orbital mechanics; a brief introduction to this topic was given in Sec. 1.4; dedicated expositions of this elegant subject can be found in [41, 46].

3.1 RELATIVE ORBITS

The term “relative orbit” entails describing the relative motion of two or more spacecraft in orbit about a common celestial body. A qualitative discussion of this topic was given in Sec. 1.4.3; I will now provide a more quantitative introduction to these concepts, introducing the necessary mathematical framework for a discussion of relative orbit control. A more complete treatment of this subject can be found in [46, 47].

To introduce any topic, one typically begins with the simplest possible case which captures the central details and phenomena of the subject at hand. We therefore will introduce this topic by considering two spacecraft orbiting the Earth. In what follows we will elevate the

status of one of these spacecraft and designate this the *chief*, and refer to the other spacecraft as the *deputy*.

Since this system comprises two bodies which are not connected in any way, 12 dynamical parameters are required to completely capture the motion: one six-element state for the chief and another six-element state for the deputy. In this section, we will represent these six-element states as column vectors. For the ECI position-velocity state, we have

$$\mathbf{X} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix}.$$

For the state-representation using orbital elements, we will find it convenient to first define a set of *modified orbit elements*, $\{a, \theta, i, q_1, q_2, \Omega\}$, with

$$\theta = \omega + f, \quad q_1 = e \cos \omega, \quad q_2 = e \sin \omega,$$

with θ referred to as the *argument of latitude*. This modified set somewhat obscures the physical picture encoded by the original set of orbit elements (which characterize an ellipse oriented in space by three Euler angles), but removes some difficulties that arise in the case of circular orbits. We can thus represent the six-element state of a orbiting spacecraft by the column vector

$$\boldsymbol{\epsilon} = [a \ \theta \ i \ q_1 \ q_2 \ \Omega]^T.$$

We will often refer to the modified orbit elements as simply the “orbital elements” unless it is necessary to distinguish between these and the original set $\{a, e, i, \omega, \Omega, f\}$; in these cases we refer to the original set as the *classical* orbit elements. We note here that other definitions and conventions for the orbital elements exist, such as the *equinoctial* elements or using *Euler parameters* $\{\beta_1, \beta_2, \beta_3, \beta_0\}$ in place of the Euler angles (see Section 2.4 of [47]). These other choices remove ambiguities that may arise in the classical and modified orbital elements, such as the ambiguity in ω/θ and Ω for equatorial orbits.

In what follows, we will represent the state of the chief using vectors $\mathbf{X} = [\mathbf{r}^T \ \mathbf{v}^T]^T$ or $\boldsymbol{\epsilon}$, and that of the deputy using subscripted vectors $\mathbf{X}_d = [\mathbf{r}_d^T \ \mathbf{v}_d^T]^T$ or $\boldsymbol{\epsilon}_d$. We will also find it convenient to represent the orbital element deputy state in terms of orbital element *differences* from the chief state,

$$\delta\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_d - \boldsymbol{\epsilon} = [\delta a \ \delta\theta \ \delta i \ \delta q_1 \ \delta q_2 \ \delta\Omega].$$

Therefore, at this point we can represent the complete 12-element state of a relative orbit in two different ways:

$$\begin{aligned} \text{ECI state: } & \{\mathbf{X}, \mathbf{X}_d\} \\ \text{Orbital elements state: } & \{\boldsymbol{\epsilon}, \delta\boldsymbol{\epsilon}\} \end{aligned}$$

Note that we can convert from one state to the other using the transformation discussed at the end of Sec. 1.4.1. Given some systematic way of dealing with the ambiguity in θ and Ω for equatorial orbits, this conversion is a one-to-one mapping which we can write symbolically as

$$\begin{aligned} \mathbf{r}, \mathbf{v} &\rightarrow \boldsymbol{\epsilon}(\mathbf{r}, \mathbf{v}) \\ \boldsymbol{\epsilon} &\rightarrow \mathbf{r}(\boldsymbol{\epsilon}), \mathbf{v}(\boldsymbol{\epsilon}) \end{aligned}$$

or more concisely as $\mathbf{X} \rightarrow \boldsymbol{\epsilon}(\mathbf{X})$ and $\boldsymbol{\epsilon} \rightarrow \mathbf{X}(\boldsymbol{\epsilon})$.

We now turn our attention to a third representation, which turns out to be very convenient when the chief-deputy relative distances are small. In these cases, the equations of motion can be linearized and the motion of the deputy about the chief can be expressed in a very simple form.

3.1.1 Relative Orbits in the LVLH Frame

In many practical applications, it is convenient to describe the position and velocity of the deputy relative to a coordinate system centered on the chief. That is, we would like to describe the position and velocity of the deputy from the vantage of the “pilot’s seat” of the chief spacecraft. To this end, we can define the *local vertical, local horizontal* (LVLH) frame, also known as the *Hill* frame, of the chief.

For an orbiting spacecraft, the LVLH frame¹ is a Cartesian coordinate system with origin

¹In this work, the LVLH frame is always defined with respect to the chief.

at the position of the spacecraft and with unit vectors defined by

$$\hat{x} = \frac{\mathbf{r}}{r} \quad (3.1a)$$

$$\hat{y} = \hat{z} \times \hat{x} \quad (3.1b)$$

$$\hat{z} = \frac{\mathbf{h}}{h}. \quad (3.1c)$$

where \mathbf{r} is the position vector of the spacecraft in ECI coordinates, $r = |\mathbf{r}|$, and $h = |\mathbf{h}|$ where $\mathbf{h} = \mathbf{r} \times \mathbf{v}$ is the angular momentum vector of the spacecraft with respect to the ECI frame. This coordinate frame is illustrated in Fig. 3.1, which was taken from [47]. We refer to the \hat{x} -, \hat{y} -, and \hat{z} -directions as the *radial*, *along-track*, and *cross-track* directions, respectively. As made clear from the figure, we will also refer to motion in the xy -plane as *in-plane* motion. Note that \hat{y} not necessarily aligned with the ECI velocity vector of the spacecraft. It is only aligned with this vector for circular orbits or at apoapsis and periapsis for elliptic orbits.

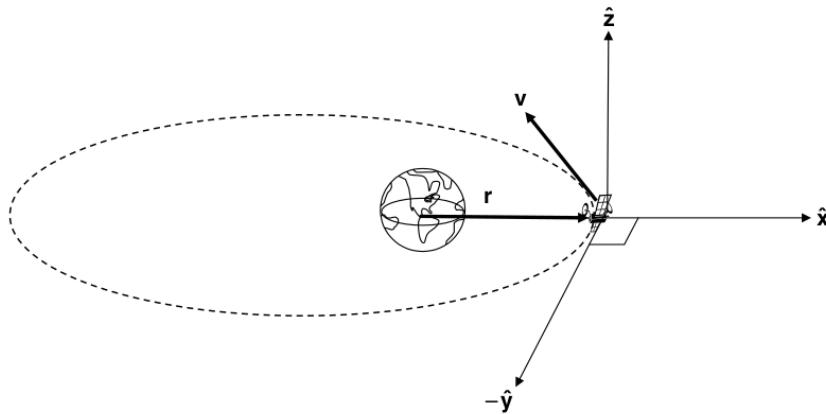


Figure 3.1: The LVLH coordinate frame for an orbiting spacecraft.

We can now represent the coordinates of the deputy spacecraft in the LVLH frame of the chief with position vector $\boldsymbol{\rho} = [x \ y \ z]^T$ and velocity vector $\dot{\boldsymbol{\rho}} = [\dot{x} \ \dot{y} \ \dot{z}]^T$. We collect these into the six-element LVLH state vector $\mathbf{x} = [\boldsymbol{\rho}^T \ \dot{\boldsymbol{\rho}}^T]^T$. Since these LVLH coordinates are dependent on the chief's ECI coordinates through Eqs. (3.1), the full state of the chief-deputy system again requires 12 elements. The LVLH coordinates of the deputy thus constitute a *third* way to represent the state of a relative orbit:

$$\text{LVLH state: } \{\mathbf{X}, \mathbf{x}\} \quad \text{or} \quad \{\boldsymbol{\epsilon}, \mathbf{x}\}$$

where we can define the LVLH coordinate system using either the chief's ECI state \mathbf{X} or orbital element vector $\boldsymbol{\epsilon}$.

3.1.2 Transforming Between Different Relative Orbit Descriptions

One can transform between the ECI and LVLH description of the chief-deputy system by realizing that

$$\mathbf{r}_d = \mathbf{r} + [\boldsymbol{\rho}]_{\text{ECI}} \quad (3.2)$$

where $[\boldsymbol{\rho}]_{\text{ECI}}$ represents the LVLH position vector expressed in ECI coordinates. Let us denote the ECI frame as \mathcal{N} and the LVLH frame as \mathcal{L} . These two frames are related by a rotation (plus an *affine* translation, so that vector magnitudes are unchanged); since the LVLH unit vectors are given in Eq. (3.1) in terms of ECI coordinates, the rotation matrix from frame \mathcal{N} to frame \mathcal{L} is given by

$$[LN] = \begin{bmatrix} \hat{\mathbf{x}}^T \\ \hat{\mathbf{y}}^T \\ \hat{\mathbf{z}}^T \end{bmatrix}, \quad (3.3)$$

and rotation from \mathcal{L} to \mathcal{N} given by $[NL] = [LN]^T$. We can therefore find the LVLH deputy position from chief and deputy ECI positions, through Eq. (3.2), as

$$\boldsymbol{\rho} = [LN](\mathbf{r}_d - \mathbf{r}). \quad (3.4)$$

This represents the upper-three components of the deputy LVLH state vector \mathbf{x} . The LVLH velocity can be found similarly: utilizing the so-called *transport* theorem, the inertial-frame time-derivative of $[\boldsymbol{\rho}]_{\text{ECI}}$ is given by

$$\frac{d}{dt}[\boldsymbol{\rho}]_{\text{ECI}} = [\dot{\boldsymbol{\rho}}]_{\text{ECI}} + \boldsymbol{\Omega} \times [\boldsymbol{\rho}]_{\text{ECI}} \quad (3.5)$$

where $[\dot{\boldsymbol{\rho}}]_{\text{ECI}}$ represents the rotating LVLH-frame time-derivative of the $\boldsymbol{\rho}$ vector, expressed in ECI coordinates, and $\boldsymbol{\Omega} = (\mathbf{r} \times \mathbf{v})/r^2$ is the angular velocity of the LVLH-frame with respect to the ECI frame (also expressed in ECI coordinates). Equation (3.5) can be rewritten in terms of ECI positions and velocities as

$$\begin{aligned} \frac{d}{dt}(\mathbf{r}_d - \mathbf{r}) &= [\dot{\boldsymbol{\rho}}]_{\text{ECI}} + \boldsymbol{\Omega} \times (\mathbf{r}_d - \mathbf{r}) \\ \Rightarrow [\dot{\boldsymbol{\rho}}]_{\text{ECI}} &= (\mathbf{v}_d - \mathbf{v}) - \boldsymbol{\Omega} \times (\mathbf{r}_d - \mathbf{r}). \end{aligned}$$

Transforming to LVLH provides the lower three components of the deputy LVLH state vector:

$$\dot{\rho} = [LN] \left((\mathbf{v}_d - \mathbf{v}) - \boldsymbol{\Omega} \times (\mathbf{r}_d - \mathbf{r}) \right). \quad (3.6)$$

Equations (3.4) and (3.6) provide the transformations which take us from ECI coordinates of our chief-deputy system, $\{\mathbf{X}, \mathbf{X}_d\}$, to LVLH coordinates of our system, $\{\mathbf{X}, \mathbf{x}\}$. To transform the deputy's LVLH coordinates to ECI coordinates, we can simply invert the above transformations. We note that these are *exact* transformations, since we are simply rotating the vectors and applying the kinematic transport theorem to transform from ECI frame to LVLH frame.

Orbital elements to LVLH

Given the relative-orbit state in terms of orbit elements, we can similarly transform to the LVLH state description $\{\mathbf{X}, \mathbf{x}\}$. A somewhat trivial way to do this, of course, is to first transform the orbital elements to ECI positions and velocities through the transformations mentioned at the end of Section 3.1. These are nonlinear transformations, which we will represent by

$$\begin{aligned} \mathbf{X} &= \mathbf{f}(\boldsymbol{\epsilon}) \\ \boldsymbol{\epsilon} &= \mathbf{g}(\mathbf{X}). \end{aligned}$$

With the positions and velocities in hand, we can now use the above results to find the LVLH coordinates of the deputy. We will represent this transformation as

$$\mathbf{x} = \mathbf{F}(\boldsymbol{\epsilon}, \delta\boldsymbol{\epsilon}) = \mathbf{F}(\mathbf{g}(\mathbf{X}), \delta\boldsymbol{\epsilon}). \quad (3.7)$$

Note that we require the chief ECI state $\boldsymbol{\epsilon}$ or \mathbf{X} , since the LVLH frame is defined by the chief. Also note that, since the transformation between orbital elements and ECI coordinates is nonlinear, we have in general

$$\delta\boldsymbol{\epsilon} = \mathbf{g}(\mathbf{X}_d) - \mathbf{g}(\mathbf{X}) \neq \mathbf{g}(\mathbf{X}_d - \mathbf{X}).$$

The transformation \mathbf{F} to LVLH coordinates is straightforward, but is nonlinear and quite complicated; this transformation, however, can be linearized when the components of $\boldsymbol{\rho}$ are small compared to the chief radius $|\mathbf{r}|$:

$$\mathbf{x} \approx \tilde{\mathbf{x}} = A(\epsilon)\delta\epsilon. \quad (3.8)$$

The transformation matrix is given by

$$A = \begin{bmatrix} \frac{r}{a} & \frac{v_r r}{v_t} & 0 & -\frac{r}{p}(2aq_1 + rc_f) & -\frac{r}{p}(2aq_2 + rs_f) & 0 \\ 0 & r & 0 & 0 & 0 & rc_i \\ 0 & 0 & rs_f & 0 & 0 & -rc_f s_i \\ -\frac{v_r}{2a} & h\left(\frac{1}{r} - \frac{1}{p}\right) & 0 & \frac{1}{p}(v_r aq_1 + hs_f) & \frac{1}{p}(v_r aq_2 - hc_f) & 0 \\ -\frac{3v_t}{2a} & -v_r & 0 & \frac{1}{p}(3v_t aq_1 + 2hc_f) & \frac{1}{p}(3v_t aq_2 + 2hs_f) & v_r c_i \\ 0 & 0 & v_t c_f + v_r s_f & 0 & 0 & (v_t s_f - v_r c_f) s_i \end{bmatrix},$$

where we have used the abbreviations $s_x \equiv \sin x$ and $c_x \equiv \cos x$. All components of this matrix are functions of chief coordinates ϵ : radial distance is given by the Kepler conic formula

$$r = \frac{a(1 - e^2)}{1 + e \cos f} = \frac{a(1 - q_1^2 - q_2^2)}{1 + q_1 \cos f - q_2 \sin f},$$

and radial and transverse velocities are

$$v_r = \dot{r} = \frac{h}{p}(q_1 \sin f - q_2 \cos f)$$

$$v_t = r\dot{\theta} = \frac{h}{p}(1 + q_1 \cos f + q_2 \sin f).$$

Additionally, angular momentum can be written $h = \sqrt{\mu p}$ and semilatus rectum is $p = a(1 - e^2)$. We leave the details of the derivation of this linearized transformation to [46].

We illustrate the above transformations using the benchmark relative orbit configuration in Table 3.1, given in terms of both orbit elements and ECI coordinates. The exact transformation to LVLH coordinates discussed at the beginning of this section (and represented by Eq. (3.7)) gives the following LVLH frame coordinates $\mathbf{x} = [\boldsymbol{\rho}^T \ \dot{\boldsymbol{\rho}}^T]^T$ of the deputy:

$$\boldsymbol{\rho} = \begin{bmatrix} -10.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix} \text{ km}, \quad \dot{\boldsymbol{\rho}} = \begin{bmatrix} 0 \\ 12.6205 \\ 11.0301 \end{bmatrix} \times 10^{-3} \text{ km/s}. \quad (3.9)$$

	a	e	i	Ω	ω	f_0
Chief	10^4 km	0	20°	30°	0°	0°
Deputy	10^4 km	0.001	20.1°	30°	0°	0°
\boldsymbol{r}_0 (10^3 km)						\boldsymbol{v}_0 (km/s)
Chief	[8.660; 5.000; 0]			[-2.966; 5.138; 2.159]		
Deputy	[8.652; 4.995; 0]			[-2.967; 5.140; 2.192]		

Table 3.1: Initial conditions for a benchmark relative orbit configuration.

The components of $\boldsymbol{\rho}$ are small compared to the chief orbit radius $r = a = 10^4$ km, so the linearized transformation Eq. (3.8) should give accurate results for the LVLH coordinates. Using this transformation, we find the LVLH state $\tilde{\boldsymbol{x}} = [\tilde{\boldsymbol{\rho}}^T \dot{\tilde{\boldsymbol{\rho}}}^T]^T$ as

$$\tilde{\boldsymbol{\rho}} = \begin{bmatrix} -10.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix} \text{ km}, \quad \dot{\tilde{\boldsymbol{\rho}}} = \begin{bmatrix} 0 \\ 12.6270 \\ 11.0191 \end{bmatrix} \times 10^{-3} \text{ km/s}. \quad (3.10)$$

This indeed well-matches the exact LVLH coordinates in Eq. (3.9). The full state of the relative orbit can now be given in terms of LVLH coordinates by specifying the orbit elements of the chief (or, equivalently, the ECI coordinates of the chief) along with either \boldsymbol{x} or $\tilde{\boldsymbol{x}}$ found above.

3.1.3 Dynamics and Linearized Equations of Motion

We now review how the relative orbit, parametrized by the LVLH coordinates of the deputy \boldsymbol{x} and by either \boldsymbol{e} or \boldsymbol{X} , evolves in time. As detailed in [46], we first represent the deputy position with respect to the ECI frame in LVLH coordinates as

$$\boldsymbol{r}_d = (r + x)\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}} + z\hat{\boldsymbol{z}},$$

with r representing the chief orbital radius. Differentiating twice with respect to the inertial frame provides

$$\ddot{\boldsymbol{r}}_d = (\ddot{r} + \ddot{x} - 2\dot{y}\dot{f} - \ddot{f}y - \dot{f}^2(r + x))\hat{\boldsymbol{x}} + (\ddot{y} + 2\dot{f}(\dot{r} + \dot{x}) + \ddot{f}(r + x) - \dot{f}^2y)\hat{\boldsymbol{y}} + \ddot{z}\hat{\boldsymbol{z}}$$

where we have utilized the transport theorem with angular velocity between LVLH and inertial frames given by $\boldsymbol{\omega} = \dot{f}\hat{\boldsymbol{z}}$, with chief true anomaly f . Further simplification by

making the replacements

$$\ddot{f} = -2\frac{\dot{r}}{r}\dot{f}, \quad \ddot{r} = r\dot{f}^2 \left(1 - \frac{r}{p}\right)$$

and then substituting \ddot{r}_d into the Newtonian gravitational dynamics $\ddot{r}_d = -(\mu/r_d^3)\mathbf{r}_d$ provides the exact LVLH dynamical equations of motion,

$$\ddot{x} = 2\dot{f} \left(\dot{y} + y\frac{\dot{r}}{r} \right) + x\dot{f}^2 + \frac{\mu}{r^2} - \frac{\mu}{r_d^3}(r+x) \quad (3.11a)$$

$$\ddot{y} = -2\dot{f} \left(\dot{x} + x\frac{\dot{r}}{r} \right) + y\dot{f}^2 - \frac{\mu}{r_d^3}y \quad (3.11b)$$

$$\ddot{z} = -\frac{\mu}{r_d^3}z \quad (3.11c)$$

where $r_d = \sqrt{(r+x)^2 + y^2 + z^2}$. Since no approximations have been made in this derivation, these dynamics describe the relative motion of any relative orbit configuration, including large chief-deputy separations. Given an initial configuration parametrized by \mathbf{x}_0 and either \mathbf{e}_0 or \mathbf{X}_0 , we can numerically integrate the above nonlinear equations to find the relative orbit trajectory $\mathbf{x}(t)$. Since these equations are exact, the resulting trajectory will exactly match the trajectory we find by solving for the individual Keplerian motion of both chief and deputy, finding the position and velocity differences, and transforming these difference vectors to LVLH frame.

The equations of motion can be simplified and/or linearized in several ways. If the orbit of the chief is circular, $\dot{r} = 0$ and $\dot{f} = n$, Eqs. (3.11) simplify to

$$\ddot{x} = 2n\dot{y} + xn^2 + \frac{\mu}{r_c^2} - \frac{\mu}{r_d^3}(r_c+x) \quad (3.12a)$$

$$(\text{Circular chief orbit}) \quad \ddot{y} = -2n\dot{x} + yn^2 - \frac{\mu}{r_d^3}y \quad (3.12b)$$

$$\ddot{z} = -\frac{\mu}{r_d^3}z. \quad (3.12c)$$

These equations are still nonlinear and exact.

On the other hand, if the relative orbit coordinates (the components of $\boldsymbol{\rho}$) are small compared to the chief orbit radius, Eqs. (3.11) can be linearized by approximating

$$r_d = r\sqrt{1 + 2\frac{x}{r} + \frac{x^2 + y^2 + z^2}{r^2}} \approx r\sqrt{1 + 2\frac{x}{r}}, \quad \text{and thus} \quad \frac{\mu}{r_d^3} \approx \frac{\mu}{r^3} \left(1 - 3\frac{x}{r}\right),$$

which, after dropping higher-order terms (see [46] for details), gives

$$\ddot{x} = x\dot{f}^2 \left(1 + 2\frac{r}{p}\right) + 2\dot{f} \left(\dot{y} - y\frac{\dot{r}}{r}\right) \quad (3.13a)$$

$$(Small \ separations) \quad \ddot{y} = -2\dot{f} \left(\dot{x} - x\frac{\dot{r}}{r}\right) + y\dot{f}^2 \left(1 - \frac{r}{p}\right) \quad (3.13b)$$

$$\ddot{z} = -\frac{r}{p}\dot{f}^2 z. \quad (3.13c)$$

In this case no assumption is made on the eccentricity of the chief, so these equations are approximately valid for all relative orbits as long as the spacecraft separation is small. These equations will be referred to as the *linearized equations of relative motion* (LERM).

Finally, if we assume small separations *and* a circular chief orbit, we find

$$\ddot{x} = 2n\dot{y} + 3n^2 x \quad (3.14a)$$

$$(Circular \ chief, \ small \ separations) \quad \ddot{y} = -2n\dot{x} \quad (3.14b)$$

$$\ddot{z} = -n^2 z \quad (3.14c)$$

These linearized equations are known as the *Hill-Clohessy-Wiltshire* (HCW) equations [48]. They are extremely convenient for analyzing the motion and designing controllers for spacecraft in close proximity to one another. Being linear equations of the form

$$\ddot{\rho} = A_1\rho + A_2\dot{\rho}$$

with $\rho = [x \ y \ z]^T$, the HCW equations can be put into the form (*c.f.*, Eq. (1.4))

$$\dot{x} = \begin{bmatrix} \mathbf{0} & I_3 \\ A_1 & A_2 \end{bmatrix} x, \quad A_1 = \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.15)$$

Since the matrices A_1 and A_2 are time independent, we see that the HCW equations describe a second-order LTI system.

The A_1 and A_2 submatrices capture the non-inertial motion of the deputy in the LVLH frame of the chief. Furthermore, we see that the presence of off-diagonal components within A_2 indicates that the equations of motion for the x and y LVLH components are coupled. Since, however, there are no off-diagonal elements corresponding to the z -component, the

motion of the deputy in the z direction decouples. This is in fact directly apparent from the second-order form of the HCW equations, Eq. (3.14), so that we can separately consider the *in-plane* dynamics

$$\begin{aligned}\ddot{x} &= 2n\dot{y} + 3n^2x \\ \ddot{y} &= -2n\dot{x}\end{aligned}$$

and the *out-of-plane* dynamics

$$\ddot{z} = -n^2z,$$

the latter of which corresponds to simple harmonic motion. Thus, regardless of the LVLH initial conditions, the deputy will simply oscillate in the z direction with frequency n , or with period equal to the orbital period of the chief.

3.1.4 Relative Orbits in HCW Dynamics

The HCW equations Eqs. (3.14) describe trajectories of an orbiting body as viewed from the reference frame of another orbiting body in a circular orbit. As mentioned, these equations are accurate descriptions only when the trajectory components $[x(t), y(t), z(t)]$ are small compared to the distance between the center of the gravitating body and the origin of the coordinate system.

The HCW equations can be integrated directly to give general solutions of the form

$$\begin{aligned}x(t) &= A_0 \cos(nt + \alpha) + \frac{2d}{n} \\ y(t) &= -2A_0 \sin(nt + \alpha) - 3dt + y_0 \\ z(t) &= B_0 \cos(nt + \beta),\end{aligned}\tag{3.16}$$

where $d = \dot{y}_0 + 2nx_0$, A_0 , B_0 , and phase angles are determined from initial conditions. The only non-periodic time-dependent term in these equations is the second term in $y(t)$, $-3dt$. If initial conditions are such that

$$d = 0 \quad \Rightarrow \quad \dot{y}_0 = -2nx_0,\tag{3.17}$$

then this secular term vanishes and the trajectory is a closed path in LVLH coordinates. This corresponds to a relative orbit where, from the point of view of the chief, the deputy traverses an elliptical path in a period equal to the orbital period of the chief, $T = 2\pi/n$. Note that in this case the in-plane (x - y) motion is always traces a 2:1 ellipse.

To illustrate, we will numerically integrate the HCW equations starting from the initial conditions corresponding to the relative orbit in Table 3.1. Specifically, we will chose the linearized LVLH state, Eq. (3.10) as our initial conditions $\mathbf{x}_0 = \tilde{\mathbf{x}}$. Since $n = \sqrt{\mu/a^3} = 6.3135 \times 10^{-4}\text{s}^{-1}$, these initial conditions satisfy the no-drift condition Eq. (3.17), so we expect the trajectory to give a closed ellipse.

The computed LVLH trajectory is shown as the purple line in Fig. 3.2. We indeed see that the trajectory is a closed ellipse, agreeing with expectations from the general solution Eq. (3.16) and Eq. (3.17). We also superimpose on this plot the trajectory found by propagating the Keplerian orbits of the chief and deputy separately, then subsequently transforming the difference in these trajectories to the LVLH frame. The plotted trajectories show excellent agreement, demonstrating the validity of the HCW dynamics in this case.

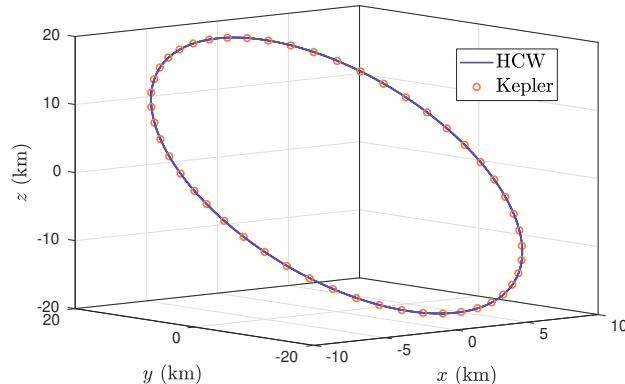


Figure 3.2: Relative orbit trajectory for the initial conditions in Table 3.1. We plot the trajectory from integrating the HCW equations (purple) along with the trajectory found by propagating the Keplerian motion separately for chief and deputy and subsequently transforming to the LVLH frame.

Calibration of HCW initial conditions

In the above numerical example, we chose to use the LVLH initial conditions $\tilde{\mathbf{x}}$ from the linearized transformation to propagate the HCW equations, as we found that these initial conditions satisfied the no-drift condition. This may seem curious; since the orbit elements for chief and deputy have identical semi-major axis, their orbital periods are identical and we expect the *exact* transformation to LVLH frame, Eq. (3.9), to describe a closed relative orbit. In fact, the exact LVLH initial conditions do lead to a closed-relative orbit *as long as we use the exact relative orbit dynamics in Eq. (3.12)*. The HCW equations, however, introduce linearization error into the trajectory, and for this reason the exact LVLH initial conditions do not quite satisfy the HCW no-drift condition Eq. (3.17).

The fact that the *linearized* transformation to LVLH relative orbit state $\tilde{\mathbf{x}}$ gives rise in this case to a more accurate trajectory under the *linearized* equations of motion is no accident. As discussed in [49, 50], there exists an equivalence between the linearized equations and the linearized OE transformation to LVLH coordinates. It is shown in these references by direct substitution that the linearized transformation Eq. (3.8) is a solution to the LERM, Eq. (3.13), for general ϵ and $\delta\epsilon$. If, on the other hand, we propagate the exact LVLH IC determined from Eq. (3.7) using the LERM, linearization error is essentially injected twice into the solution.

This is illustrated as follows: we can represent the exact nonlinear equations of motion Eq.(3.11) functionally as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}). \quad (3.18)$$

Linearization of this equation lead us to the LERM, Eq. (3.13), and this linearization can be expressed in this notation as

$$\dot{\mathbf{x}} \approx \mathbf{f}(\mathbf{0}) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{0}} (\mathbf{x} - \mathbf{0})$$

Clearly $\mathbf{x} = \mathbf{0}$ is an equilibrium point, since this represents two spacecraft in the exact same orbit (so that there will be no relative motion). Therefore $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and

$$\dot{\mathbf{x}} \approx \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{0}} \mathbf{x} \equiv \mathbf{F}\mathbf{x}. \quad (3.19)$$

The matrix \mathbf{F} therefore represents the LTV system matrix corresponding to the LERM in Eq. (3.13).

We can also represent the exact nonlinear transformation from $\mathbf{y} \equiv [\boldsymbol{\epsilon}, \delta\boldsymbol{\epsilon}]$ to \mathbf{x} as

$$\mathbf{x} = \mathbf{a}(\mathbf{y})$$

which similarly linearizes as

$$\mathbf{x} \approx \left. \frac{\partial \mathbf{a}}{\partial \mathbf{y}} \right|_0 \mathbf{y} \equiv A_0 \mathbf{y}.$$

Similar manipulations lead to the inverted analogues of the above relationships:

$$\mathbf{y} = \mathbf{b}(\mathbf{x}) \Rightarrow \mathbf{y} \approx \left. \frac{\partial \mathbf{b}}{\partial \mathbf{x}} \right|_0 \mathbf{x} \equiv B_0 \mathbf{x}$$

and

$$\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}) \Rightarrow \dot{\mathbf{y}} \approx \left. \frac{\partial \mathbf{g}}{\partial \mathbf{y}} \right|_0 \mathbf{y} \equiv \mathbf{G} \mathbf{y}.$$

By taking the time derivative of the transformation from \mathbf{y} to \mathbf{x} ,

$$\mathbf{x} = \mathbf{a}(\mathbf{y}) \Rightarrow \dot{\mathbf{x}} = \left. \frac{\partial \mathbf{a}}{\partial \mathbf{y}} \right|_0 \dot{\mathbf{y}} \equiv A(\mathbf{y}) \dot{\mathbf{y}} = \underbrace{A(\mathbf{b}(\mathbf{x})) \mathbf{g}(\mathbf{b}(\mathbf{x}))}_{\mathbf{f}(\mathbf{x})}$$

and then utilizing Eq. (3.18) and (3.19), we can find the relationship between the LTV system matrices for the LVLH (\mathbf{x}) and linearized orbital element difference (\mathbf{y}) dynamics:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) \approx \mathbf{F}\mathbf{x} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_0 \mathbf{x} \\ \mathbf{F} &= \left. \frac{\partial}{\partial \mathbf{x}} \left(A(\mathbf{b}(\mathbf{x})) \mathbf{g}(\mathbf{b}(\mathbf{x})) \right) \right|_0 \\ &= \left[\frac{\partial A(\mathbf{b}(\mathbf{x}))}{\partial \mathbf{x}} \mathbf{g}(\mathbf{b}(\mathbf{x})) + A(\mathbf{b}(\mathbf{x})) \frac{\partial \mathbf{g}(\mathbf{b}(\mathbf{x}))}{\partial \mathbf{x}} \right] \\ &= A_0 \mathbf{G} B_0 \end{aligned} \tag{3.20}$$

With the above relationships now in hand, let us explore how one would propagate the initial LVLH conditions using the LERM. Since these equations are linear, we can propagate an initial condition using the STM:

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)} \mathbf{x}_0 = A_0 e^{\mathbf{G}(t-t_0)} \underbrace{B_0 \mathbf{x}_0}_{\approx \mathbf{y}_0} \approx A_0 \mathbf{y}(t)$$

where in the second equality we have utilized Eq. (3.20) and the matrix exponential property $e^{PM(t)Q} = Pe^{M(t)}Q$ derived in Chapter 1. Inspection of this last equality reveals that linearization error is injected *twice* in order to propagate the exact LVLH initial state forward in time using the LERM: Once by using the linearized transformation from \mathbf{x}_0 to \mathbf{y}_0 , and once again by using the linearized transformation from $\mathbf{y}(t)$ to $\mathbf{x}(t)$. To avoid error injected from the linearized transformation from \mathbf{y} to \mathbf{x} , we can use the exact transformation $\mathbf{y} = \mathbf{b}(\mathbf{x})$. We therefore define the *calibrated* solution as

$$\tilde{\mathbf{x}}(t) \equiv A_0 e^{\mathbf{G}(t-t_0)} \mathbf{b}(\mathbf{x}_0).$$

Inserting $B_0 A_0 = I$, (which follows from the fact that B_0 and A_0 are inverse linearized transformations), this becomes

$$\tilde{\mathbf{x}}(t) = A_0 e^{\mathbf{G}(t-t_0)} B_0 A_0 \mathbf{b}(\mathbf{x}_0) = e^{\mathbf{F}(t-t_0)} A_0 \mathbf{b}(\mathbf{x}_0) = e^{\mathbf{F}(t-t_0)} A_0 \mathbf{y}_0 = e^{\mathbf{F}(t-t_0)} \tilde{\mathbf{x}}_0$$

We therefore realize that by using the linearized transformation of the initial condition, $\tilde{\mathbf{x}}_0 = A_0 \mathbf{y}_0$, this is properly propagated by its corresponding STM derived from the LERM.

In a sense, then, we are *calibrating* the initial condition for propagation in the linearized equation of motion. We therefore refer to the LVLH state $\tilde{\mathbf{x}}$ from the linearized transformation as the *calibrated* initial condition, and the resulting trajectory we call the *calibrated* solution or trajectory. We demonstrate this calibration numerically in Fig. 3.3, where we plot the components of the trajectory position error $\delta\tilde{\mathbf{x}} = \tilde{\mathbf{x}} - \mathbf{x}_{\text{true}}$, where \mathbf{x}_{true} is the true relative orbit trajectory found by propagating the chief and deputy Keplerian orbits separately and subsequently transforming to the LVLH frame. Note that \mathbf{x}_{true} corresponds to the “Kepler” data points in Fig. 3.2, and this trajectory is identical to the trajectory one would find by propagating the nonlinear relative motion equations Eqs. (3.11) using the exact initial condition. We also plot the trajectory position error $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}_{\text{true}}$ using the exact initial conditions in the HCW equations (*i.e.*, the *uncalibrated* solution). We see that the magnitude of the propagation error is smaller overall for the calibrated solution. Furthermore, we see that this error is purely periodic, whereas in the case of the uncalibrated solution there is a secular growth to the error in the y -direction. This secular growth arises due to the fact that the exact ICs do not satisfy the HCW no-drift condition.

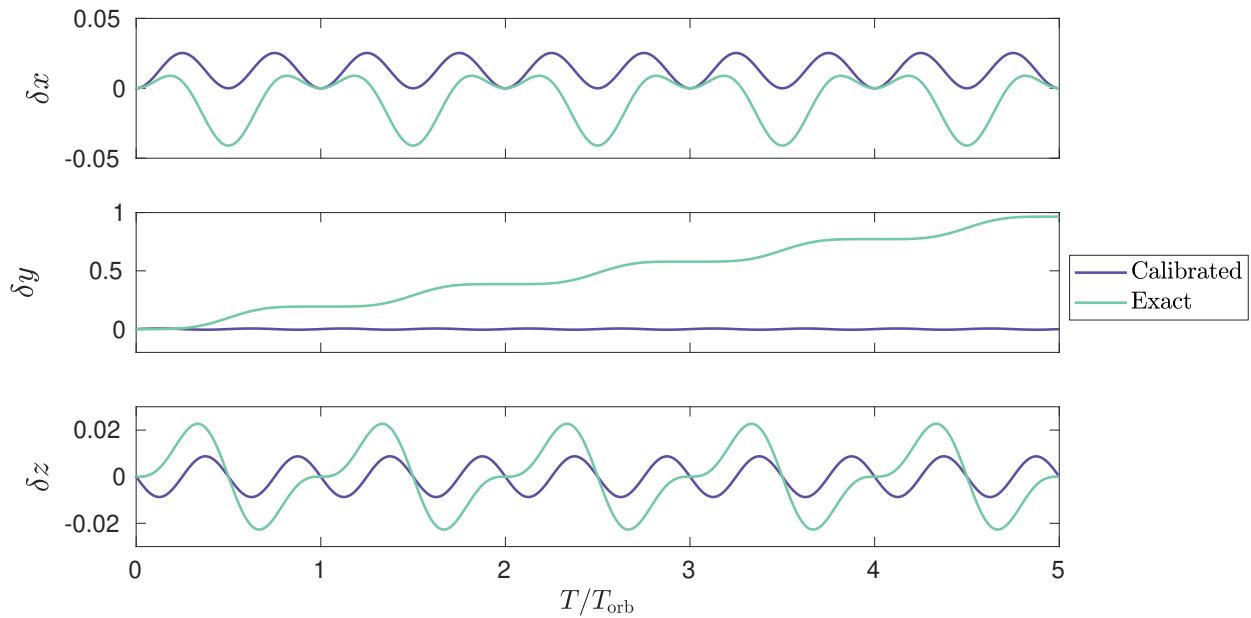


Figure 3.3: Position error of HCW-propagated relative orbit trajectory compared to true relative orbit found by propagating chief and deputy Keplerian orbits separately. We plot trajectories using calibrated initial conditions $\tilde{\mathbf{x}}$ (purple) and exact initial conditions \mathbf{x} (teal).

In the controlled trajectories considered herein, calibration of our initial state is largely unnecessary since the resulting trajectory differences are negligible compared to effects of the controller. I have nonetheless introduced it here as I believe it is an important and underappreciated method, which can be applied broadly in systems obeying linearized approximations of the true dynamics. Moreover, in [51] these calibration concepts were incorporated in *control laws* for linearized systems. It would therefore be interesting to see how the refinement afforded through calibrated control affects the trajectories and control efficiencies considered in this work.

3.1.5 Relative Orbit Control

We will now introduce continuous controllers for the deputy satellite for the purposes of altering the relative orbit. As a simple application for such a controller, we will imagine that we wish to achieve rendezvous between the chief and deputy for the purposes of docking. In later sections we will generalize these control goals and consider controllers which achieve

some formation of spacecraft.

We will assume continuous control protocols in what follows, but it should be generally possible to later “digitize” these controllers for use in high-frequency impulse thrusters. We will also assume for now that the thruster which performs the control protocol is either appropriately gimballed and variable, or that a set of orthogonally directed variable thrusters exist, so that the force vector can be pointed in any direction at any given time. Note that LVLH position ρ and velocity $\dot{\rho}$ will be used as feedback to the controller, and the control vector will also be expressed in the LVLH frame as $\mathbf{u} = u_x \hat{\mathbf{x}} + u_y \hat{\mathbf{y}} + u_z \hat{\mathbf{z}}$.

We recall that the full nonlinear equations of motion for the free (uncontrolled) LVLH coordinates of the deputy were given by Eqs. (3.11). We can express these second-order differential equations in first-order form, and add the thrust induced by the controller as follows:

$$\begin{aligned}\frac{d}{dt} \rho &= \dot{\rho} \\ \frac{d}{dt} \dot{\rho} &= \mathbf{F}(\rho, \dot{\rho}) + \mathbf{u}.\end{aligned}$$

Here, the nonlinear vector function \mathbf{F} represents the right-hand side of Eq. (3.11). Assuming a circular chief orbit and linearizing provided the HCW equations in Eqs. (3.14), which we expressed as the 6-dimensional LTI uncontrolled state equation in Eq. (3.15). We can thus express the controlled HCW equations as the LTI state equation

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{0} & I_3 \\ A_1 & A_2 \end{bmatrix}}_{A_{CW}} \mathbf{x} + \underbrace{\begin{bmatrix} \mathbf{0} \\ I_3 \end{bmatrix}}_B \mathbf{u}, \quad (3.21)$$

where A_1 and A_2 we defined in Eq. (3.15) and are repeated here for convenience:

$$A_1 = \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.22)$$

We note here that if we are interested in non-circular chief orbits, an analogous state equation can be extracted from the LERM, Eqs. (3.13); in this case the linearized system will no longer be time invariant (*i.e.*, the analogues of A_1 and A_2 will have time-dependent terms proportional to \dot{f} , r , and \dot{r}).

If our desired maneuver is a chief-deputy rendezvous, then we want our controller to drive $\mathbf{x} \rightarrow \mathbf{0}$; that is, we seek a regulation controller. A simple way to design such a controller is to use full-state feedback of the form $\mathbf{u} = -K\mathbf{x}$, where $K = [K_P \ K_D]$, so that we can “close the loop”. Note that since $\mathbf{x} = [\boldsymbol{\rho}^T \ \dot{\boldsymbol{\rho}}^T]^T$, $K_P, K_D \in \mathbb{R}^{3 \times 3}$ correspond to the *proportional* and *derivative* gain matrices introduced in Sec. 1.1.5 in the context of 3D double integrator systems. The controlled equations of motion therefore become

$$\dot{\mathbf{x}} = (A_{CW} - BK)\mathbf{x} = \begin{bmatrix} \mathbf{0} & I_3 \\ (A_1 - K_P) & (A_2 - K_D) \end{bmatrix} \mathbf{x} = \underbrace{\begin{bmatrix} \mathbf{0} & I_3 \\ \tilde{A}_1 & \tilde{A}_2 \end{bmatrix}}_{A_{CL}} \mathbf{x} \quad (3.23)$$

where in the last equality we have defined the controlled HCW submatrices \tilde{A}_1 and \tilde{A}_2 , along with the closed-loop system matrix A_{CL} . To guarantee that rendezvous is achieved asymptotically, we thus require A_{CL} to be Hurwitz.

In general, the submatrices \tilde{A}_1 and \tilde{A}_2 have nonvanishing off-diagonal terms; these arise from the off-diagonal terms in A_2 along with any off-diagonal terms we have chosen in K_P and K_D in our controller design. Therefore, as alluded to in Sec. 1.1.5, it is by no means obvious how one chooses our controller gains such that the controlled system is stable (asymptotically reaches rendezvous). One possibility is to perform a brute-force survey over the elements of K in order to optimize some performance measure; being an 18-dimensional parameter space,² this method is neither computationally straightforward nor elegant. Alternatively, we can either choose our gains such that all components decouple, or determine our gains through LQR optimization (or some alternative optimization scheme).

Decoupling HCW through PD control

The spatial components of the controlled HCW equations will decouple if the submatrices \tilde{A}_1 and \tilde{A}_2 are diagonal. In this case, each component will evolve according to a damped harmonic oscillator as long as all diagonal elements of these matrices are negative. These

²In fact, since the z -component dynamics decouples from the dynamics of the in-plane components, one can most likely simply optimize these two dynamical sub-systems separately. This reduces the dimension of the parameter space to 10.

matrices are explicitly written as

$$\begin{aligned}\tilde{A}_1 &= (A_1 - K_P) = \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix} - \begin{bmatrix} k_{P_x} & 0 & 0 \\ 0 & k_{P_y} & 0 \\ 0 & 0 & k_{P_z} \end{bmatrix} = \begin{bmatrix} a_x & 0 & 0 \\ 0 & a_y & 0 \\ 0 & 0 & a_z \end{bmatrix} \\ \tilde{A}_2 &= (A_2 - K_D) = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} k_{D_x} & k_{D_{xy}} & 0 \\ k_{D_{yx}} & k_{D_y} & 0 \\ 0 & 0 & k_{D_z} \end{bmatrix} = \begin{bmatrix} b_x & 0 & 0 \\ 0 & b_y & 0 \\ 0 & 0 & b_z \end{bmatrix}.\end{aligned}\quad (3.24)$$

For all nonzero elements of \tilde{A}_1 and \tilde{A}_2 to be negative, we therefore find the following requirements on the gain matrix elements:

$$k_{P_x} > 3n^2, \quad k_{P_y} > 0, \quad k_{P_z} > -n^2, \quad (3.25a)$$

$$k_{D_x} > 0, \quad k_{D_y} > 0, \quad k_{D_z} > 0, \quad k_{D_{xy}} = -k_{D_{yx}} = 2n \quad (3.25b)$$

Under these conditions, the gain matrices which decouple the HCW spatial components are given by

$$K_P = \begin{bmatrix} 3n^2 + a_x & 0 & 0 \\ 0 & a_y & 0 \\ 0 & 0 & -n^2 + a_z \end{bmatrix}, \quad K_D = \begin{bmatrix} b_x & 2n & 0 \\ -2n & b_y & 0 \\ 0 & 0 & b_z \end{bmatrix} \quad (3.26)$$

Although the diagonal components of K_D can be arbitrarily set, one sensible way of picking these gain coefficients is to require that each component of the closed-loop system is critically damped. In this case, the equations of motion will be described by

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & I_3 \\ \tilde{A}_1 & \tilde{A}_2 \end{bmatrix} \mathbf{x}, \quad \tilde{A}_1 = \begin{bmatrix} -a_x & & \\ & -a_y & \\ & & -a_z \end{bmatrix}, \quad \tilde{A}_2 = \begin{bmatrix} -2\sqrt{a_x} & & \\ & -2\sqrt{a_y} & \\ & & -2\sqrt{a_z} \end{bmatrix} \quad (3.27)$$

with all omitted elements zero.

We now simulate the trajectory under this control law using the benchmark relative orbit configuration in Table 3.1. To minimize fuel cost, we will take advantage of the natural dynamics by setting $k_{P_z} = 0$, so that $a_z = n^2$. We will pick the x - and y -component gains so that Eqs. (3.25a) are satisfied:

$$k_{P_x} = 1.1(3n^2), \quad k_{P_y} = 0.1(3n^2) \Rightarrow a_x = 0.1(3n^2), \quad a_y = 0.1(3n^2) \quad (3.28)$$

(note that the factors “1.1” and “0.1” were design choices). The diagonal components of K_D are now set to give the critically damped system Eq. (3.27):

$$b_x = k_{D_x} = 2\sqrt{a_x}, \quad b_y = K_{D_y} = 2\sqrt{a_y}, \quad b_z = k_{D_z} = 2\sqrt{a_z} \quad (3.29)$$

These gains now determine the equations of motion, Eq. (3.23), under decoupling PD control. The propagation of the trajectory characterized by the initial conditions Eq. (3.10) is shown in Fig. 3.4.

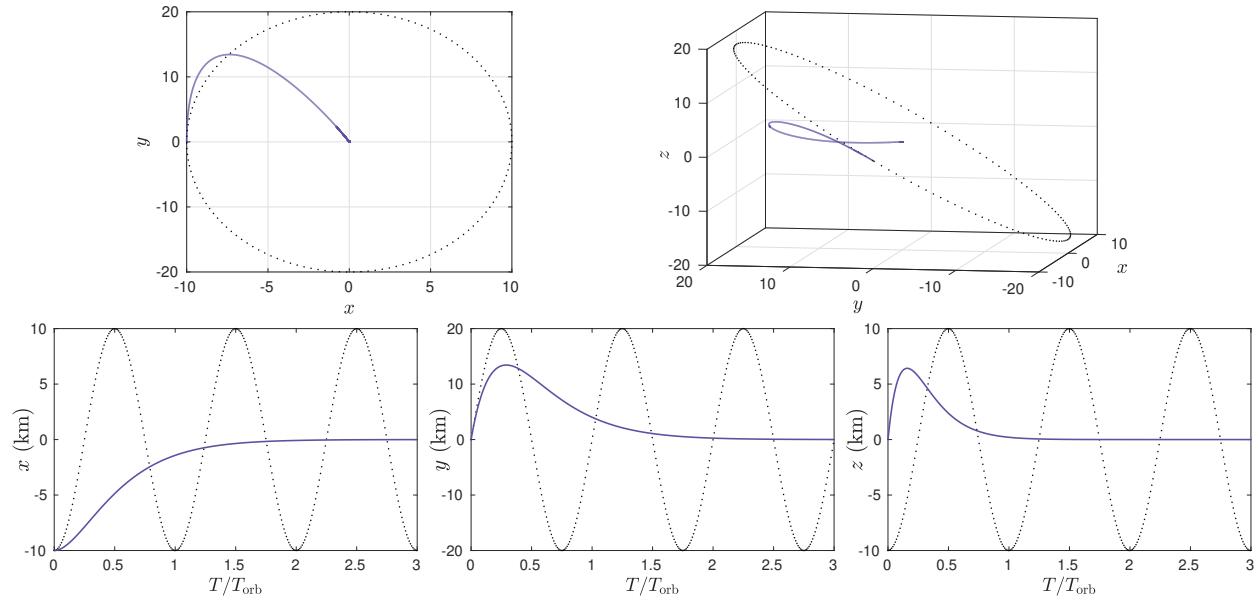


Figure 3.4: Relative orbit trajectory under the decoupling PD controller described by Eqs. (3.23) and (3.24). Gains are chosen so that each component is critically damped. At top we show the 3D trajectory from two different view points, and in the bottom row we show each component of the trajectory separately. The uncontrolled relative orbit is shown for reference as the dotted lines.

HCW control through LQR design

Although the above strategy is straightforward, the fact that the components of ρ and $\dot{\rho}$ are coupled through the HCW equations implies that a diagonal form for the closed-loop equations will not in general correspond to the most efficient controller. For instance, it may be possible to more efficiently use the natural dynamics of the system, in concert with the controller, to more efficiently bring the system to equilibrium ($\mathbf{x} \rightarrow 0$).

Since the controlled HCW equations in Eq. (3.21) are LTI, we can design an optimal regulation feedback controller of the form $\mathbf{u} = -K\mathbf{x}$ utilizing the linear quadratic regulator

(LQR) formalism introduced in Sec. 1.1.4. To this end, let us assume that we would like to minimize the cost function

$$J = \int_0^\infty (\mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}) dt, \quad (3.30)$$

so that integrated displacement, velocity, and control effort associated with the trajectory are all optimized. LQR theory determines the controller gain to be

$$K = R^{-1} B^T P$$

where P is determined from the continuous algebraic Riccati equation

$$A^T P + PA - PBR^{-1}B^T P = -Q.$$

We will again illustrate by propagating the trajectory of the benchmark system in Table 3.1 under this controller. Based on the relative scales of the LVLH position, velocity, and control effort, we choose the following forms for the LQR matrices in order to optimize both the trajectory and the integrated control effort:

$$Q = \begin{bmatrix} I_3 & \mathbf{0} \\ \mathbf{0} & \frac{1}{n^2} I_3 \end{bmatrix}, \quad R = \frac{100}{n^4} I_3 \Rightarrow K_{LQR} = 10^{-4} \begin{bmatrix} 3.27e-3 & -2.74e-4 & 0 & 2.07 & 2.24 & 0 \\ 8.23e-3 & -2.89e-4 & 0 & 2.24 & 6.79 & 0 \\ 0 & 0 & 1.99e-5 & 0 & 0 & 0.892 \end{bmatrix}. \quad (3.31)$$

We note that the zero entries in the optimal gain matrix computed above reflect the fact that the z -component is decoupled from the x - and y -components in the HCW equations.

The resulting trajectory is plotted in Fig. 3.5. We find that the LQR successfully stabilizes the system, driving the relative orbit trajectory $\mathbf{x} \rightarrow 0$. This trajectory also, of course, optimizes the cost function Eq. (3.30) for the specific Q and R matrices we have chosen in Eq. (3.31).

We notice, however, prolonged oscillations along the z -axis, however, which may be undesirable for specific practical applications. This can be remedied by increasing the *weight* corresponding to these component in the cost function. Specifically, we can adjust the Q matrix as

$$Q = \begin{bmatrix} I_3 & \mathbf{0} \\ \mathbf{0} & \frac{1}{n^2} I_3 \end{bmatrix} \rightarrow Q = \begin{bmatrix} \begin{pmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & c_1 \end{pmatrix} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n^2} \begin{pmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & c_2 \end{pmatrix} \end{bmatrix}$$

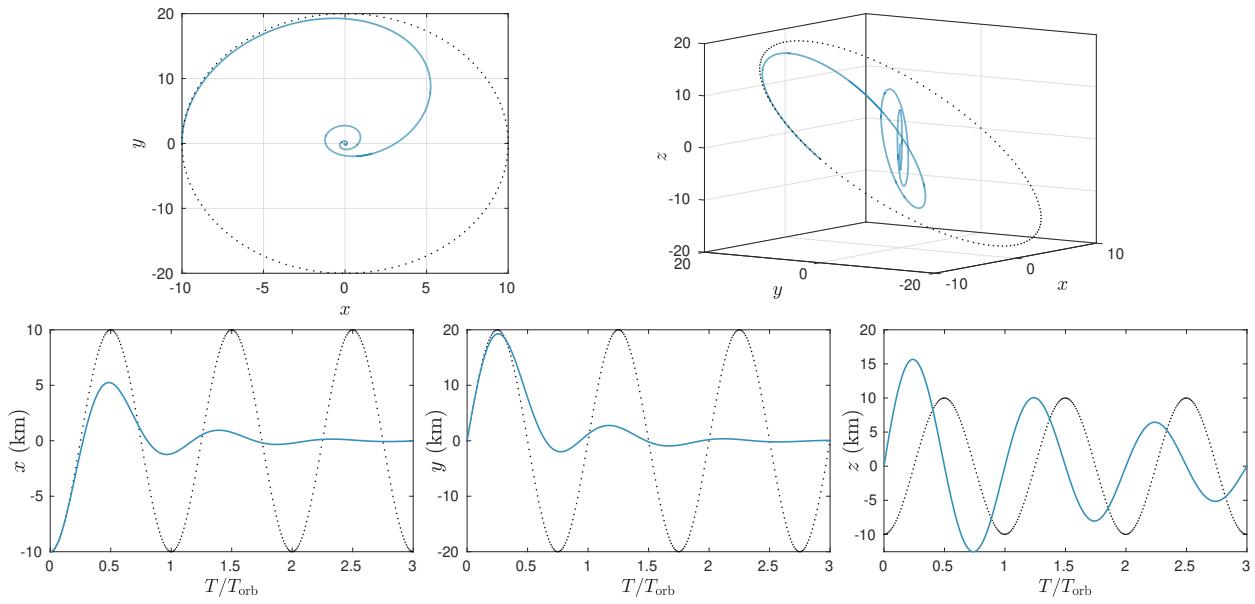


Figure 3.5: Relative orbit trajectory under the LQR-designed controller described by Eqs. (3.23) and (3.31). The uncontrolled relative orbit is shown for reference as the dotted lines.

where $c_1, c_2 > 1$.

Alternatively, since the z -component is decoupled from the x and y components, we can optimize these subsystems separately. To gain some insight into the differences between the LQR-designed controller and the decoupled PD controller discussed above, we will choose the z -component gains in these two controllers to match, and optimize the x and y component gains using the reduced 2-dimensional system

$$\dot{\underline{x}} = \begin{bmatrix} \mathbf{0} & I_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \underline{x} + \begin{bmatrix} \mathbf{0} \\ I_2 \end{bmatrix} \underline{u}$$

where $\underline{x} = [x \ y \ \dot{x} \ \dot{y}]^T$ (and where in this equation $\mathbf{0}$ represents the 2×2 matrix of zeros), and where the cost function matrices are

$$\underline{Q} = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & \frac{1}{n^2} I_2 \end{bmatrix}, \quad \underline{R} = \frac{100}{n^4} I_2.$$

The resulting gains are of course identical to the gains we found in Eq. (3.31) corresponding to the x and y components. Picking the same z component gains as in Eqs. (3.28) and (3.29),

the new LQR-optimized gain matrix is

$$K'_{LQR} = 10^{-4} \begin{bmatrix} 3.27e-3 & -2.74e-4 & 0 & 2.07 & 2.24 & 0 \\ 8.23e-3 & -2.89e-4 & 0 & 2.24 & 6.79 & 0 \\ 0 & 0 & 0 & 0 & 0 & 12.627 \end{bmatrix}. \quad (3.32)$$

We now plot the trajectories and control signals from this LQR-designed HCW controller in Fig. 3.6. We also superimpose the same data from the decoupled PD controlled system. We see that both controllers give rise to approximately the same settling time (*i.e.*, $\mathbf{x} \rightarrow 0$ in approximately the same amount of time), but the LQR-designed system achieves this using much less control effort.

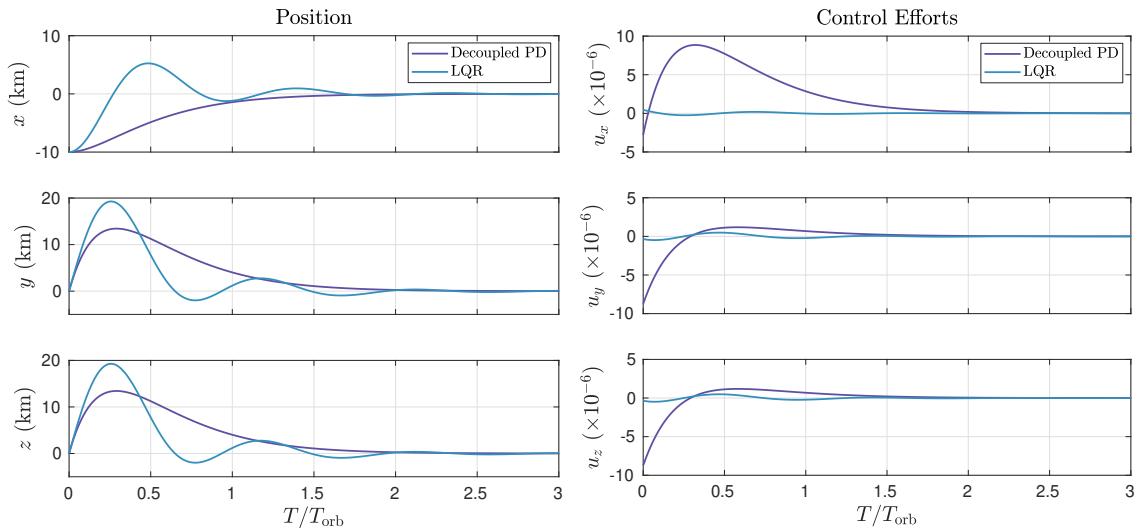


Figure 3.6: Comparison of trajectory (left) and control effort (right) for closed-loop system using decoupled PD control design and LQR control design. Control signals are in units of km/s 2 .

We should note here that Fig. 3.6 is simply for comparison purposes of different control strategies, and it perhaps may be misleading. These plots do not necessarily imply that one control design strategy is superior to another. The controllers have been designed in completely different ways; for the decoupled controller we have demanded that the response is critically damped, whereas for the LQR controller we have made no similar demand. One could imagine that by tuning the gains in the decoupled PD controller so that overshoot is increased, at some point both control effort *and* system response may be comparable. Indeed,

one of the main purposes of Fig. 3.6 is to show how sensitive the controlled trajectory is to the off-diagonal elements in the system and gain matrices.

3.2 FRACTIONAL CONTROL OF RELATIVE ORBITS

In Sec. 3.1.5 we outlined two LVLH-frame controllers which, when applied by the deputy, bring the relative orbit coordinates to zero. Both controllers utilized state-feedback of the form $\mathbf{u} = -K\mathbf{x}$ so that the closed-loop equations were of the form $D\mathbf{x} = (A - BK)\mathbf{x}$. In the former controller, we chose K such that the closed-loop system reduced to three independent damped-harmonic oscillators (or, equivalently, three separate point masses in 1D under PD control), while in the latter the gain matrix K was found through LQR optimization. In either case, the controller can be written in PD form as

$$\mathbf{u} = -K_P\boldsymbol{\rho} - K_D\dot{\boldsymbol{\rho}} = -K_P\boldsymbol{\rho} - K_DD\boldsymbol{\rho} \quad (3.33)$$

so that the closed-loop equations can be expressed compactly as the state equation

$$D \begin{bmatrix} \boldsymbol{\rho} \\ \dot{\boldsymbol{\rho}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & I_3 \\ (A_1 - K_P) & (A_2 - K_D) \end{bmatrix}}_{A_{CL}} \begin{bmatrix} \boldsymbol{\rho} \\ \dot{\boldsymbol{\rho}} \end{bmatrix},$$

where the HCW submatrices A_1 and A_2 were given in Eq. (3.22).

We now seek to apply fractional control to the system, generalizing Eq. (3.33) to

$$\mathbf{u} = -K_P\boldsymbol{\rho} - K_DD^{\alpha_D}\boldsymbol{\rho}. \quad (3.34)$$

This is an example of a PD-type fractional controller or, more concisely, a PD^α controller. Due to the fractional term $D^{\alpha_D}\boldsymbol{\rho}$, the closed-loop system cannot be put into the form of an LTI state equation as before. This is due to the fact that the controller cannot be expressed in a linear state-feedback form,

$$-K_P\boldsymbol{\rho} - K_DD^{\alpha_D}\boldsymbol{\rho} \iff -K\mathbf{x} = -K \begin{bmatrix} \boldsymbol{\rho} \\ \dot{\boldsymbol{\rho}} \end{bmatrix}.$$

We are thus forced to write the closed-loop fractionally-controlled HCW equations either as a nonlinear fractional differential equation

$$\ddot{\boldsymbol{\rho}} = \mathbf{F}(\boldsymbol{\rho}, D^{\alpha_D}\boldsymbol{\rho}), \quad (3.35)$$

or, if we restrict the derivative order as $0 < \alpha_D \leq 1$ and $\alpha_D \in \mathbb{Q}$, by writing the system as a pseudostate equation of commensurate order α

$$D^\alpha \boldsymbol{\xi} = \bar{A} \boldsymbol{\xi}.$$

We will utilize the former approach for our simulations (as will be outlined below), and for now simply illustrate how one builds the pseudostate equation for the case $\alpha_D = \frac{1}{2}$. In this case, the closed-loop fractional HCW equations Eq. (3.35) are

$$\begin{aligned} \ddot{\boldsymbol{\rho}} &= A_1 \boldsymbol{\rho} + A_2 \dot{\boldsymbol{\rho}} - K_P \boldsymbol{\rho} - K_D D^{\frac{1}{2}} \boldsymbol{\rho} \\ \Rightarrow D^2 \boldsymbol{\rho} &= (A_1 - K_P) \boldsymbol{\rho} + A_2 (D \boldsymbol{\rho}) - K_D (D^{\frac{1}{2}} \boldsymbol{\rho}) \end{aligned} \quad (3.36)$$

The commensurate order for this system is $\alpha = \frac{1}{2}$ (in this case it is equal to α_D). The commensurate form for the equation of motion Eq. (3.36) is

$$\underbrace{D^\alpha \boldsymbol{\rho}}_{D^\alpha (D^{3\alpha} \boldsymbol{\rho})} = (A_1 - K_P) \boldsymbol{\rho} - K_D D^\alpha \boldsymbol{\rho} + A_2 D^{2\alpha} \boldsymbol{\rho}.$$

Defining the pseudostate³ $\boldsymbol{\xi} = [\boldsymbol{\rho}; D^\alpha \boldsymbol{\rho}; D^{2\alpha} \boldsymbol{\rho}; D^{3\alpha} \boldsymbol{\rho}]$, the pseudostate equation is now given by

$$D^\alpha \begin{bmatrix} \boldsymbol{\rho} \\ D^\alpha \boldsymbol{\rho} \\ D^{2\alpha} \boldsymbol{\rho} \\ D^{3\alpha} \boldsymbol{\rho} \end{bmatrix} = \begin{bmatrix} 0 & I_3 & 0 & 0 \\ 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & I_3 \\ A_1 - K_P & -K_D & A_2 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\rho} \\ D^\alpha \boldsymbol{\rho} \\ D^{2\alpha} \boldsymbol{\rho} \\ D^{3\alpha} \boldsymbol{\rho} \end{bmatrix}, \quad (\alpha = \frac{1}{2})$$

or, simply, as $D^\alpha \boldsymbol{\xi} = \bar{A} \boldsymbol{\xi}$.

In order to simulate the fractionally-controlled HCW equations while varying α_D , we will instead work from the closed-loop equations given in Eq. (3.35). Defining $\boldsymbol{\sigma}_1 = \boldsymbol{\rho} = [x; y; z]$ and $\boldsymbol{\sigma}_2 = \dot{\boldsymbol{\rho}} = [x; y; z]$, these equations can be rewritten as

$$D\boldsymbol{\sigma}_1 = \boldsymbol{\sigma}_2 \quad (3.37a)$$

$$D\boldsymbol{\sigma}_2 = (A_1 - K_P) \boldsymbol{\sigma}_1 + A_2 \boldsymbol{\sigma}_2 - K_D D^\alpha \boldsymbol{\sigma}_1. \quad (3.37b)$$

³We will again revert to the column-array notation $[\mathbf{a}; \mathbf{b}; \mathbf{c}] \equiv [\mathbf{a}^T \ \mathbf{b}^T \ \mathbf{c}^T]^T$ for general column vectors \mathbf{a} , \mathbf{b} , and \mathbf{c} .

Here, the derivative orders on the separate components of $\boldsymbol{\rho}$ can differ; we have thus used a boldface “ α ” in the above notation, so that

$$D^\alpha \boldsymbol{\sigma}_1 = D^\alpha \boldsymbol{\rho} \equiv \begin{bmatrix} D^{\alpha_x} x \\ D^{\alpha_y} y \\ D^{\alpha_z} z \end{bmatrix}.$$

The most general PD $^\alpha$ -type controller therefore has 18 free parameters from the tunable gains (the elements of K_P and K_D), plus three free parameters associated with the fractional-derivative orders of the x -, y -, and z - components of $\boldsymbol{\rho}$. Though a brute-force approach utilizing a numerical survey over the parameter space can be used, our goal will be to develop a more methodical strategy of controller design.

3.2.1 Decoupling the HCW Equations Using Fractional Control

In general, the gain matrices K_P and K_D have nonzero off-diagonal elements, so that all components are coupled. Previously, we were able to decouple the three components of $\boldsymbol{\rho}$ by choosing K_P and K_D appropriately as to cancel the off-diagonal elements of the system matrices A_1 and A_2 ; this gave the closed-loop system in Eq. (3.27) which described three separate damped-harmonic oscillators. In the fractional case considered here, this is *no longer possible* since there is no way to cancel the off-diagonal elements in the last two terms of Eq. (3.37). For instance, if we attempted to choose the gains as before, which decoupled the components in the integer-order case, the fractional equations of motion Eq. (3.37) written in terms of components are given by

$$\begin{aligned}\ddot{x} &= -\tilde{a}_x x + 2n(\dot{y} - D^{\alpha_y} y) - \tilde{b}_x D^{\alpha_x} x \\ \ddot{y} &= -\tilde{a}_y y - 2n(\dot{x} - D^{\alpha_x} x) - \tilde{b}_y D^{\alpha_y} y \\ \ddot{z} &= -\tilde{b}_z z - \tilde{b}_z D^{\alpha_z} z.\end{aligned}$$

We see that the z component decouples as before, becoming an independent *fractionally-damped* harmonic oscillator. The x and y components, however, remain coupled due to the fact that the fractional derivatives of the controller do not cancel the integer-order derivatives of the dynamics.

These manipulations, however, give us some insight into how one can build a fractional controller which successfully decouples the three components. If we instead utilize a PDD $^\alpha$ controller (*i.e.*, containing both integer-order and fractional-order derivatives),

$$\mathbf{u} = -K_P \boldsymbol{\rho} - K_{D_1} D \boldsymbol{\rho} - K_{D_2} D^\alpha \boldsymbol{\rho}$$

with

$$\begin{aligned} K_P &= A_1 - \tilde{A}_1 \\ K_{D_1} &= A_2 \\ K_{D_2} &= -\tilde{A}_2 \end{aligned} \tag{3.38}$$

with \tilde{A}_1 and \tilde{A}_2 again diagonal, then the closed-loop fractionally controlled equations of motion become

$$\begin{aligned} \ddot{x} &= -\tilde{a}_x x - \tilde{b}_x D^{\alpha_x} x \\ \ddot{y} &= -\tilde{a}_y y - \tilde{b}_y D^{\alpha_y} y \\ \ddot{z} &= -\tilde{a}_z z - \tilde{b}_z D^{\alpha_z} z. \end{aligned} \tag{3.39}$$

Note that in the fractional case, the condition $\tilde{b}_i = 2\sqrt{\tilde{a}_i}$ does not necessarily correspond to critical damping. We can now choose the diagonal components of \tilde{A}_1 and \tilde{A}_2 independently to adjust the damped behavior of each component, as before. The general form of the gain matrices which decouple the spatial components in the PDD $^\alpha$ controller are given by

$$K_P = \begin{bmatrix} 3n^2 + \tilde{a}_x & 0 & 0 \\ 0 & \tilde{a}_y & 0 \\ 0 & 0 & -n^2 + \tilde{a}_z \end{bmatrix}, \quad K_{D_1} = \begin{bmatrix} 0 & -2n & 0 \\ 2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad K_{D_2} = \begin{bmatrix} \tilde{b}_x & 0 & 0 \\ 0 & \tilde{b}_y & 0 \\ 0 & 0 & \tilde{b}_z \end{bmatrix}. \tag{3.40}$$

The efficiency (in terms of control effort) of this method of controller design is by no means guaranteed. It is quite possible, however, that PD $^\alpha$ - or PDD $^\alpha$ -type controller, with appropriately tuned off-diagonal gains, can out-perform the integer-order LQR-designed controller for a given cost function.

Numerical simulation of the PDD $^\alpha$ -decoupled HCW system

We will now compare the trajectories and performance of the PDD $^\alpha$ -controlled HCW system with the integer-order PD-controlled system from Sec. 3.1.5. We again use the benchmark

relative orbit defined in Table 3.1. In both cases, we decouple the spatial components of the closed-loop equations of motion as above, and design the PD-controlled system to be critically damped in each component.

We therefore choose our integer-order PD gains such that the decoupled system is

$$\begin{aligned}\ddot{x} + 2\sqrt{a_x}\dot{x} + a_x x &= 0 \\ \ddot{y} + 2\sqrt{a_y}\dot{y} + a_y y &= 0 \\ \ddot{z} + 2\sqrt{a_z}\dot{z} + a_z z &= 0.\end{aligned}$$

This determines the gain matrices through Eq. (3.26), with $b_i = 2\sqrt{a_i}$, $i = x, y, z$. We then design our PDD^α controller so that the decoupled equations of motion Eq. (3.39) have the same coefficients $\tilde{a}_i = a_i$ and $\tilde{b}_i = b_i$:

$$\begin{aligned}\ddot{x} + 2\sqrt{a_x}D^{\alpha_x}x + a_x x &= 0 \\ \ddot{y} + 2\sqrt{a_y}D^{\alpha_y}y + a_y y &= 0 \\ \ddot{z} + 2\sqrt{a_z}D^{\alpha_z}z + a_z z &= 0.\end{aligned}$$

This determines the fractional gain matrices through Eqs. (3.40).

We can now compare the PD and PDD^α controllers by picking values for the coefficients a_x, a_y, a_z (common to both controllers) and varying the fractional orders $\{\alpha_x, \alpha_y, \alpha_z\}$. We first simulate the trajectory taking $a_x = a_y = 0.1n^2$, $a_z = n^2$, and setting all fractional orders equal, $\alpha_x = \alpha_y = \alpha_z$. The resulting trajectories, from two different view points, are shown in Fig. 3.7.

To gain a better insight into the trajectory differences, we plot each component separately as a function of time in Fig. 3.8. We generally see that integer-order control leads to higher overshoot. Fractional-order control reduces this overshoot, but the effects of the creeping behavior (see discussion of the Mittag-Leffler function in Sec. 1.3.2) are noticeable and non-negligible, resulting in a much slower approach to rendezvous.

Through inspection of these trajectory components, however, we can attempt to design a better controller by “mixing” the fractional orders. For instance, to minimize overshoot and response time simultaneously, we can choose $\alpha_x = 1$, $\alpha_y = 0.85$, and $\alpha_z = 0.85$. The

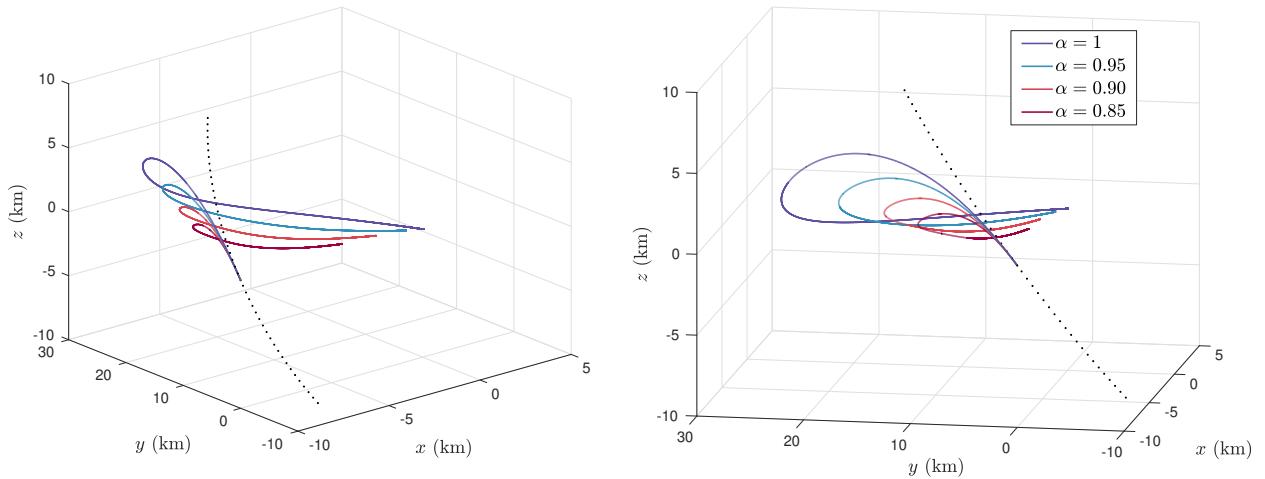


Figure 3.7: Trajectories under decoupled fractional PDD^α control, for equal fractional orders $\alpha_x = \alpha_y = \alpha_z$. We show various values of fractional order, with $\alpha = 1$ corresponding to the standard integer-order controller.

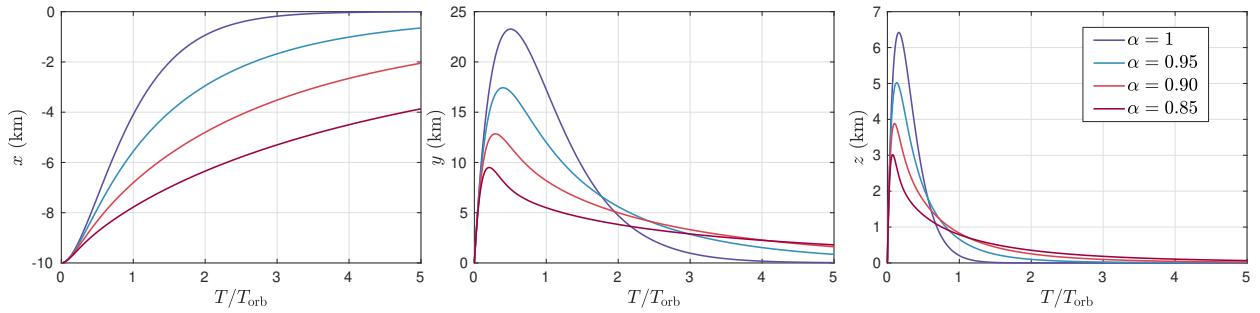


Figure 3.8: Position components of the trajectories in Fig. 3.7.

resulting trajectory as compared to the original integer-order PD-controlled trajectory, is shown in Fig. 3.9.

The fractional controller, as tuned above, has clear benefits over the integer-order controller. Comparing the two trajectories, we find that the fractionally controlled trajectory approaches rendezvous in a more direct manner; in other words, from the perspective of the chief, the direction of the deputy has a much smaller variation during the maneuver. It therefore seems that we have more freedom in how to shape the approach utilizing fractional controllers. Of course in these plots we have kept control gains constant. Varying these gains gives additional freedom in shaping the trajectory so that one can design an approach depending on mission requirements (*e.g.*, docking port locations, maximum

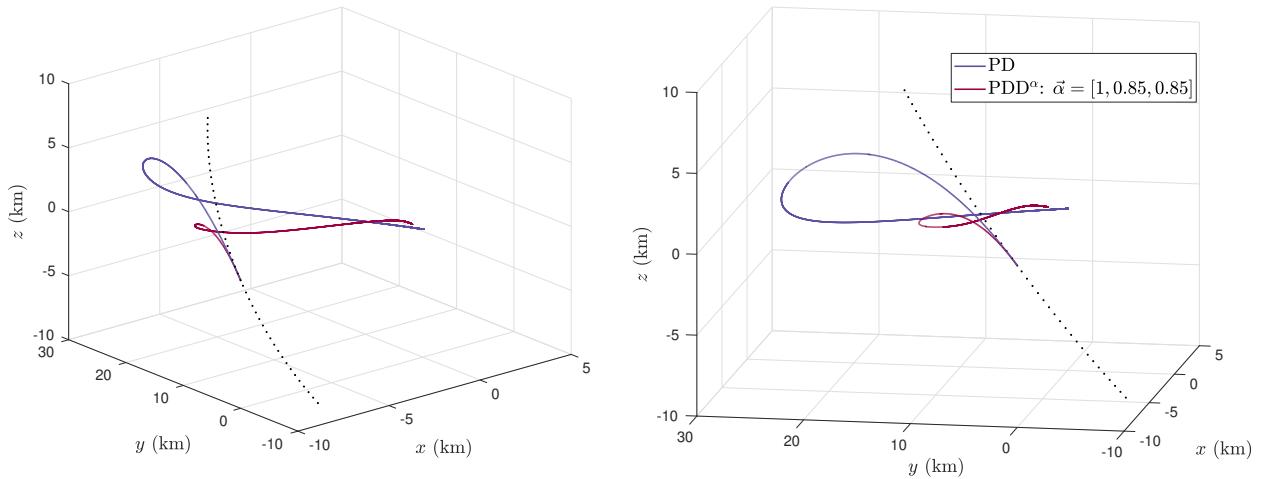


Figure 3.9: Trajectories under the tuned fractional PDD^α control, $\boldsymbol{\alpha} = [1.00, 0.85, 0.85]$, as compared to the integer-order PD-controlled trajectory.

approach velocity, *etc.*). The fractional derivatives essentially allow us to vary both settling time (which corresponds to approach velocity) and overshoot, while when using a standard PD controller these properties are inherently dependent on each other.

3.2.2 LQR-Designed Controller Using Fractional Derivatives

We will now test another approach in how to utilize the extra freedom afforded to us through the fractional order. Let us return to the LQR-designed controller considered previously, as represented by the gain matrix in Eq. (3.32). This sets the gains for feedback terms proportional to the position coordinates and velocity coordinates as

$$\mathbf{u} = -K\mathbf{x} = -[K_P \ K_D] \begin{bmatrix} \boldsymbol{\rho} \\ \dot{\boldsymbol{\rho}} \end{bmatrix} = -K_P \boldsymbol{\rho} - K_D D \boldsymbol{\rho}.$$

Let us now simply use these *same* gains while adjusting the derivative terms:

$$\mathbf{u} = -K_P \boldsymbol{\rho} - K_D \begin{bmatrix} D^{\alpha_x} x \\ D^{\alpha_y} y \\ D^{\alpha_z} z \end{bmatrix}. \quad (3.41)$$

This direct substitution is somewhat *ad-hoc*. After all, these gain matrices were determined, through the LQR procedure, to optimize the trajectory only in the case of the integer-order

PD-controlled system. The fractional control law gives a completely different dynamical system. We should therefore not expect that these choices for K_P and K_D optimize the trajectory when fractional control is used. However, for now we can use these as *approximately optimal* as long as we do not allow our fractional orders to stray far from unity.

In what follows we will only be interested in the x - and y -component trajectories. The main difference between this controller and previous controllers is the nonzero coupling gains in for these components; the z -component trajectory decouples as before. We first take $\alpha_y = \alpha_z = 1$ and vary α_x . The resulting x and y trajectory components, along with the control signals in these two dimensions, are shown in Fig. 3.10. We also simulate the resulting trajectory after setting $\alpha_x = \alpha_z = 1$ and varying α_y . These trajectories, along with the control signals, are shown in Fig. 3.11.

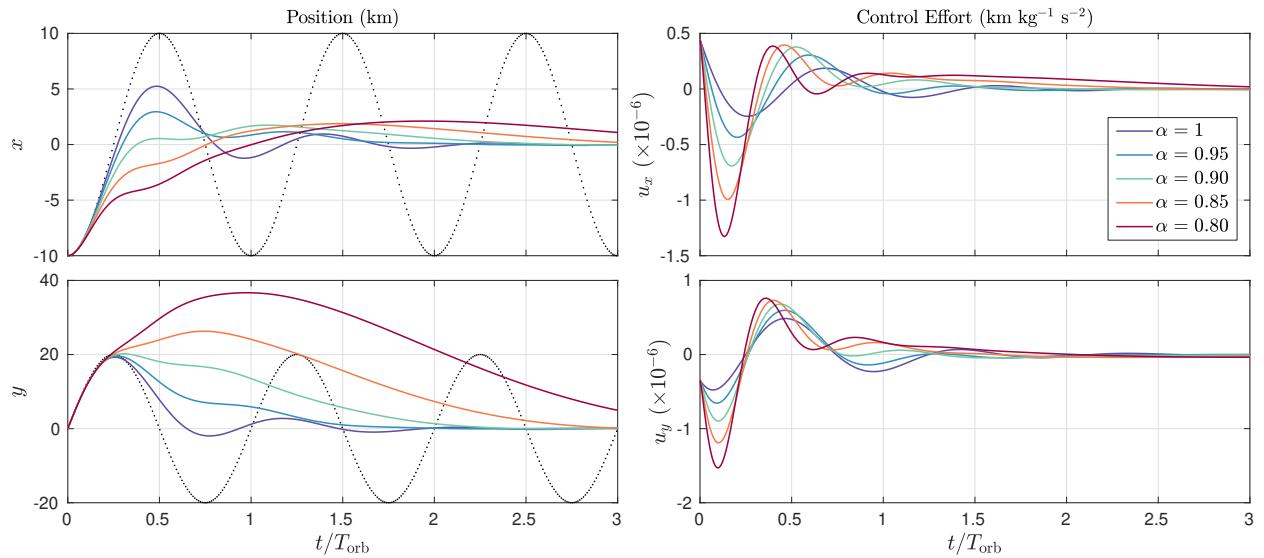


Figure 3.10: Components of relative position (left) and control effort (right) for relative orbit trajectories under fractional control. We set $\alpha_y = 1$ and vary the order of the x -component derivative, α_x , using the LQR designed gains. The uncontrolled trajectory is shown as a dotted line.

A number of observations can be made from these numerical simulations. First, we see that the trajectory is much more sensitive to the y -component derivative order. Indeed, even

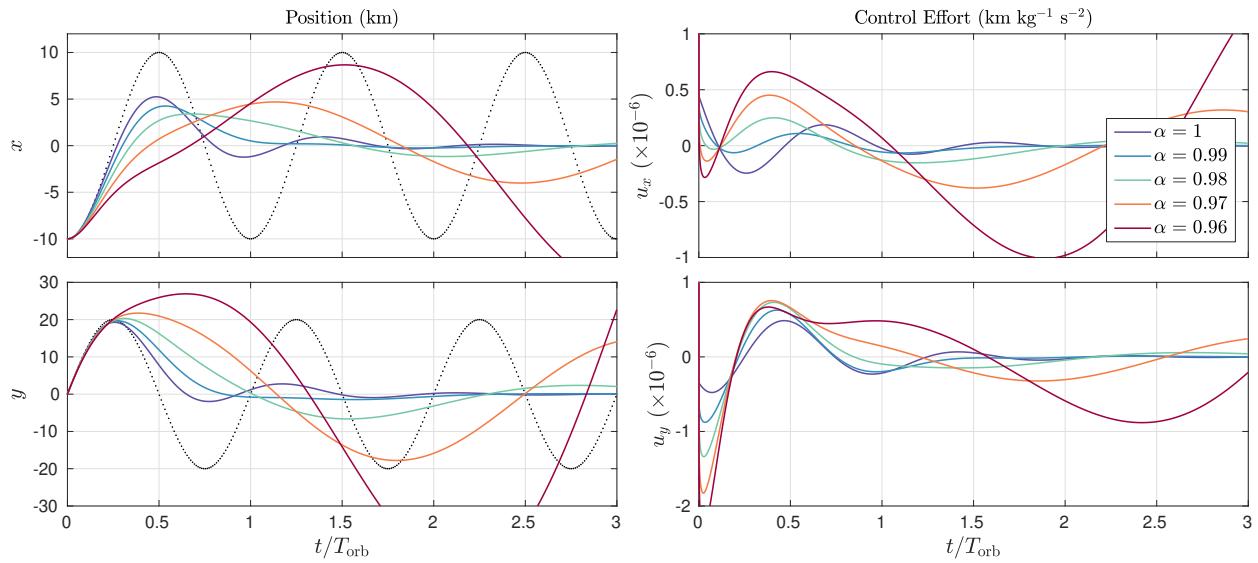


Figure 3.11: Same as Fig. 3.10, but setting $\alpha_x = 1$ and varying the order of the y -component derivative, α_y .

taking $\alpha_y = 0.99$ causes a very noticeable change in the trajectory. In fact, stability is lost when $\alpha_y \lesssim 0.96$ (and thus the range in which we vary α_y in the figures is smaller).

As far as the overall shape of the trajectories, we see that introducing fractional derivatives generally leads to less overshoot, which in relative orbit trajectories corresponds to less ‘circling’ of the deputy about the chief. This behavior was also seen when we compared the fractional and integer-order PD controllers, as seen previously in Fig. 3.8. With this reduction in overshoot, however, control effort increases accordingly.

Lastly, we mention that since we have strayed only slightly away from the integer-order LQR controller in the above simulations (the components of $\boldsymbol{\alpha}$ were all close to unity), we expect the computed value of the cost function, Eq. (3.30), to vary about the integer-order computed cost. The controllers in these simulations, however, are not *optimized* in any sense, since the LQR gains only optimize the integer-order system. It seems very possible that certain regions of $\{\alpha_x, \alpha_y, \alpha_z\}$ parameter space can actually reduce the total (integer-order LQR) cost of the maneuver, giving a quantifiable advantage of the fractional controller introduced herein.

3.2.3 Numerical Survey Techniques in Fractional Controller Design

We can now return to the general PD^α controller as given in Eq. (3.41). As previously mentioned, this controller contains 21 free parameters, 18 of which are the elements of the K_P and K_D gain matrices and three of which are the fractional orders $\boldsymbol{\alpha} = [\alpha_x; \alpha_y; \alpha_z]$. We recall from Sec. 3.1.5, however, that since the HCW dynamics only couple the x and y components, we can optimize this subsystem separately from the z component. In other words, to optimize a relative orbit trajectory under PD^α control, we can separately optimize the two controllers

$$\begin{aligned} \begin{bmatrix} u_x \\ u_y \end{bmatrix} &= - \begin{bmatrix} k_{11}^P & k_{12}^P & k_{11}^D & k_{12}^D \\ k_{21}^P & k_{22}^P & k_{21}^D & k_{22}^D \end{bmatrix} \begin{bmatrix} x \\ y \\ D^{\alpha_x}x \\ D^{\alpha_y}y \end{bmatrix} \\ u_z &= - [k_{Pz} \quad k_{Dz}] \begin{bmatrix} z \\ D^{\alpha_z}z \end{bmatrix}. \end{aligned}$$

Since we have already discussed control for fractionally damped oscillators, we will focus on the coupled xy -component system in what follows. We see that this controller has 10 free parameters which can be independently tuned.

It will first be determined whether a fractional controller can outperform the integer-order LQR-designed controller for a given choice of Q and R matrices. Our initial relative orbit will be taken to be the benchmark system in Table 3.1, and we make the following choices for the optimization matrices

$$Q = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & \frac{1}{n^2}I_2 \end{bmatrix}, \quad R = \frac{1}{n^4}I_2.$$

The integer-order controller, represented by setting $\alpha_x = \alpha_y = 1$, corresponds to a cost function $J_{\text{LQR}} = 1.004 \times 10^6$. Let us now survey over the parameter space of the fractional controller as follows. We will survey over a 10×10 grid of $\{\alpha_x, \alpha_y\} \in [0.95, 1]$. For each value within this grid we will optimize the gain matrix $K = [K_P \ K_D]$. Since the fractional orders are in the vicinity of unity, we can use the LQR derived gains as our starting point for the survey. We therefore define a coefficient matrix as

$$K = \begin{bmatrix} a_{11} & a_{12} & b_{11} & b_{12} \\ a_{21} & a_{22} & b_{21} & b_{22} \end{bmatrix} \circ K_{\text{LQR}}$$

where $P \circ Q$ represents the *Hadamard product* of equal-dimension matrices P and Q ; the Hadamard product multiplies components of P and Q element-by-element, so that the element $[P \circ Q]_{ij} = [P]_{ij}[Q]_{ij}$. We can now survey over the dimensionless coefficients a_{ij} and b_{ij} over some domain about unity. This survey determines the optimal gain matrix (in terms of the LQR cost function) for our particular grid point in $\{\alpha_x, \alpha_y\}$. The results of this survey are shown as a contour plot in Fig. 3.12. We also show the survey results using an alternative relative orbit, with chief orbit defined as in Table 3.1 and deputy orbit defined by the orbit element differences $\delta e = 0.001$ and $\delta\Omega = 0.1^\circ$. The contours in this plot separate different values of the relative cost J^*/J_{LQR} , where J^*_{frac} is the optimal cost (found by survey over the fractional gain matrix) for the corresponding grid point in the figure.

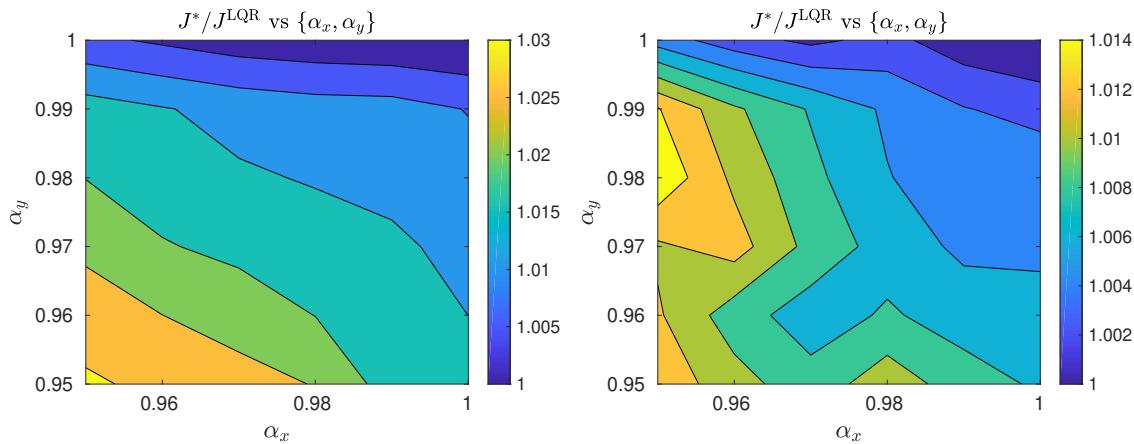


Figure 3.12: Relative performance of the optimal PD^α controller as compared to the optimal integer-order controller. At left we show the results for the relative orbit defined in Table 3.1; the right plot corresponds to an alternative set of initial conditions as given in the text. In both cases, the optimal controller corresponds to the integer-order case $\alpha_x = \alpha_y = 1$.

The results of these plots imply that the PD^α controller is unable to outperform the LQR designed integer-order controller. This result is rather unexpected considering the additional freedom afforded by the tunable orders α_x and α_y . We would naively expect that leaving the “confines of the integer-order plane” would open up more optimal trajectories, some of which would result in smaller values of the integrated quantity J . It is possible, however, that this result is connected to the fact that the LQR cost is inherently tied to the trajectory

of a system under integer-order PD control.

It will therefore be illuminating to perform a similar survey over parameter space to optimize some alternative performance measure such as settling time or some combined optimization of settling time and control effort. We expect from the results of Chapter 2 that in these cases, fractional controllers can outperform their integer-order counterparts. The results of such a survey are shown in Fig. 3.13. Here we are considering only the (decoupled) z -component of the relative-orbit trajectory, and minimizing settling time. In this case, to simplify the survey algorithm we will transform the HCW system to non-dimensional coordinates as

$$x \rightarrow u = \frac{x}{r}, \quad y \rightarrow v = \frac{y}{r}, \quad z \rightarrow w = \frac{z}{r}, \quad t \rightarrow s = nt = \frac{t}{T_{\text{orb}}}, \quad (3.42)$$

where r and T_{orb} is the radius and period, respectively, of the chief orbit (see, *e.g.*, [46]). The equation of motion for the w -component is now given by

$$w'' - k_{D_w} D^{\alpha_w} w - k_{P_w} w = 0$$

where $(\cdot)'$ represents a derivative with respect to the dimensionless coordinate s (which, incidentally, is simply the angular position of the chief in its orbit). Using the dimensionless HCW equations, the gains will be of order $\mathcal{O}(1)$; physical values of gains (along with physical trajectories) can be found by reversing the transformations Eq. (3.42). We have surveyed over the gain values $k_{P_w}, k_{D_w} \in [0, 2]$, where the upper limit of this window corresponds to some upper limit in control effort (in a practical application, this can correspond to the maximum thrust available to onboard thrusters).

Figure 3.13 shows values of settling time as a function of gains k_{P_w} and k_{D_w} , for particular choices of α_w . At right we show the slice of parameter space corresponding to the standard integer-order HCW controller ($\alpha_w = 1$), and at left we show the *optimal* slice of α_w parameter space resulting from the survey. The gains which minimize the settling time are denoted by red stars.

Using these results, we show the trajectories corresponding to these optimal point in Fig. 3.14, along with the associated control signals. The threshold position deviation which

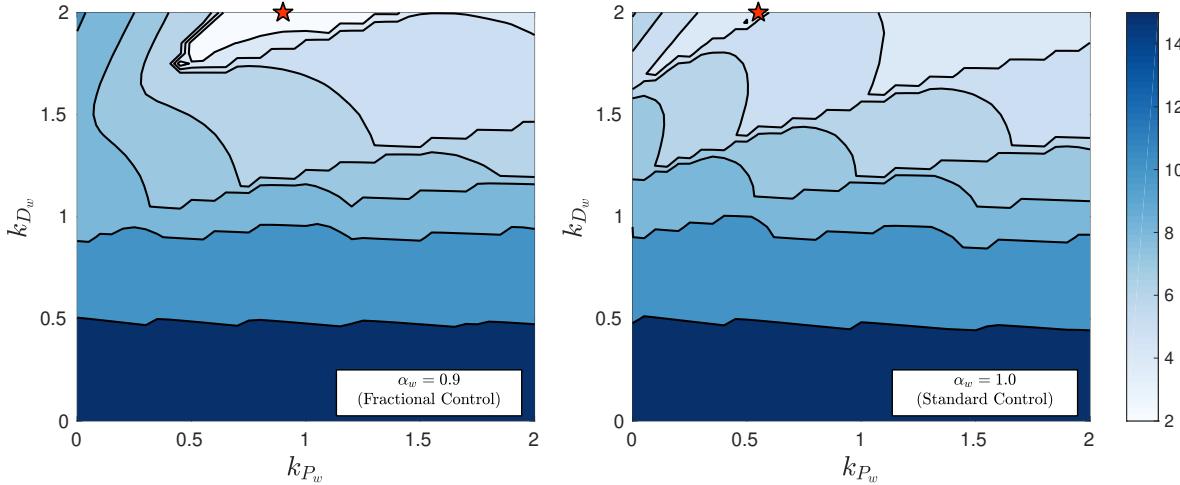


Figure 3.13: Settling time as a function of controller gains k_{P_w} and k_{D_w} . Optimal choices of gains are marked with a red star. At left we show the optimal slice of α_w parameter space; the right panel corresponds to standard PD-type control.

defines settling time for this system is shown as dotted lines in the left panel. From this figure it is clear that settling time can be substantially reduced by using fractional control. Numerically, we find an optimal settling time of $\tau_{\text{int}} = 0.625T_{\text{orb}}$ for the integer-order controller and $\tau_{\text{frac}} = 0.455T_{\text{orb}}$ for the fractional controller, corresponding to a $\approx 27\%$ decrease in settling time when fractional control is used.

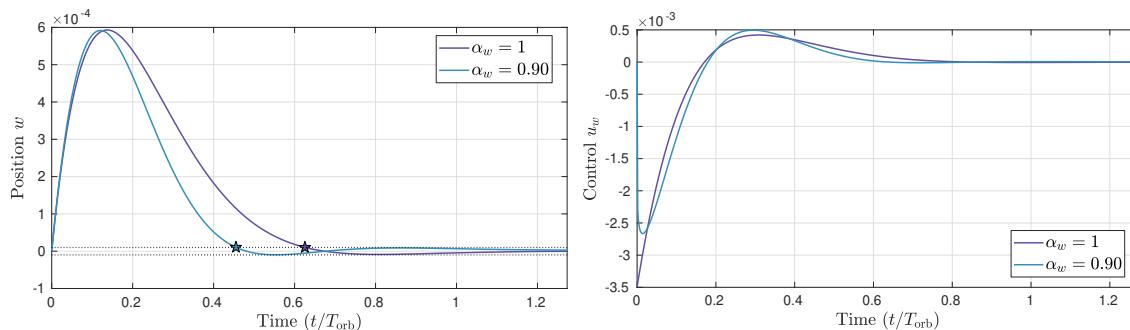


Figure 3.14: Trajectories which optimize settling time for both integer- and fractional-order control (left) along with corresponding control signals (right).

It is instructive to also compute the total integrated control signal for each trajectory,

$$U = \int_0^{\infty} |u_w(t')| dt'.$$

This corresponds to the amount of fuel spent during the maneuver, and in practical applications maneuvers which spend less fuel are more desirable. For these two trajectories, we find $U_{\text{int}} = 0.00242$ and $U_{\text{frac}} = 0.00232$, so that less fuel is spent using the fractional controller. We therefore find that the fractional controller reduces settling time by $\approx 27\%$, *and does so using less fuel* than the standard integer-order HCW controller.

3.3 CONSENSUS AND FORMATION CONTROL FOR HCW DYNAMICS

In this final section, we will see how one can utilize the formalism of multiagent cooperative control for relative orbit trajectories of multiple spacecraft. We will therefore now assume that we have a system of *more than* two spacecraft. Let us consider for now a chief SC (which defines the LVLH frame) and two deputies. The dynamics of each deputy in the chief's LVLH frame are given by Eq. (3.21),

$$\dot{\mathbf{x}}_1 = A_{\text{CW}}\mathbf{x}_1 + B\mathbf{u}_1, \quad \dot{\mathbf{x}}_2 = A_{\text{CW}}\mathbf{x}_2 + B\mathbf{u}_2. \quad (3.43)$$

Imagine that the deputies can communicate their state to each other, and that we wish for the deputies to reach consensus, $(\mathbf{x}_1 - \mathbf{x}_2) \rightarrow \mathbf{0}$ as $t \rightarrow \infty$. It seems reasonable to assume a that a control law of the form

$$\begin{aligned} \mathbf{u}_1 &= -K(\mathbf{x}_1 - \mathbf{x}_2) = K_1(\boldsymbol{\rho}_2 - \boldsymbol{\rho}_1) + K_2(\dot{\boldsymbol{\rho}}_2 - \dot{\boldsymbol{\rho}}_1) \\ \mathbf{u}_2 &= -K(\mathbf{x}_2 - \mathbf{x}_1) = K_1(\boldsymbol{\rho}_1 - \boldsymbol{\rho}_2) + K_2(\dot{\boldsymbol{\rho}}_1 - \dot{\boldsymbol{\rho}}_2) \end{aligned} \quad (3.44)$$

will accomplish this task.

Previously, we have discussed how this consensus can be achieved for a system of 3-dimensional double-integrators,

$$\frac{d}{dt} \begin{bmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & I_3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{A_{\text{di}}} \begin{bmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{0} \\ I_3 \end{bmatrix}}_B \mathbf{u}_i, \quad i = 1, 2, \dots, N, \quad (3.45)$$

using the local voting protocol

$$\mathbf{u}_i = - \sum_{j=1}^N a_{ij} \left(K_P(\mathbf{r}_i - \mathbf{r}_j) + K_D(\mathbf{v}_i - \mathbf{v}_j) \right), \quad (3.46)$$

where K_P and K_D are (3×3) gain matrices. Note that the input matrix B is equivalent to the input matrix for the HCW system; this simply reflects the fact that in both cases the bodies in question obey second-order Newtonian dynamics. By defining the local-representation state vector for the entire system as $\mathbf{X}_L = [\mathbf{r}_1; \mathbf{v}_1; \mathbf{r}_2; \mathbf{v}_2; \dots; \mathbf{r}_N; \mathbf{v}_N]$, the closed-loop equations of motion for this system are given by

$$\dot{\mathbf{X}}_L = \left((I_N \otimes A_{\text{di}}) - (L \otimes BK_{\text{di}}) \right) \mathbf{X}_L, \quad (3.47)$$

where $K_{\text{di}} = [K_P \ K_D]$. The ability to write the closed-loop equations in this form in fact had no reliance on the specific dynamics encoded in A_{di} , only that the dynamical equation of motion for each agent could be expressed as an LTI system as in Eq. (3.45).

Indeed, since both deputies can communicate their state to each other, the communication topology is associated with a graph Laplacian matrix

$$L = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (3.48)$$

and, defining the local-representation 2-deputy state $\mathbf{x}_L = [\mathbf{x}_1; \mathbf{x}_2]$, the HCW dynamics of the deputies along with the control protocols Eq. (3.44) can be written similarly as

$$\dot{\mathbf{x}}_L = \left((I_N \otimes A_{\text{CW}}) - (L \otimes BK) \right) \mathbf{x}_L, \quad (3.49)$$

where $K = [K_1 \ K_2]$. In other words, the form of the closed-loop equations of motion under consensus control are unchanged regardless of the specific dynamics encoded in system matrices A and input matrices B .

We should clarify what we mean by consensus in this particular context. Recall that consensus is generally defined as $(\mathbf{x}_i - \mathbf{x}_j) \rightarrow 0$ as $t \rightarrow \infty$ for all $i, j \in \{1, 2, \dots, N\}$, where N is the number of agents in the system. In this case, the “system” consists of only the deputy spacecraft. The chief spacecraft simply defines the LVLH coordinate system in which these dynamics are defined; we typically assume this chief is uncontrolled. In fact, in the present context (where we are bringing multiple deputies to consensus) the chief *need not correspond to any physical spacecraft*. In this case, the consensus control law serves to achieve rendezvous of *multiple* spacecraft, these spacecraft being the deputies of some virtual chief.

After consensus is achieved, the spacecraft will have converged to some common relative orbit about the virtual chief.

We will now illustrate the above scenario through example. Let us consider an Earth-orbiting chief SC (which may or may not be a physical SC) with the following orbit elements,

$$[a, e, i, \Omega, \omega, M_0] = [10^4 \text{ km}, 0, 20^\circ, 30^\circ, 0^\circ, 0^\circ], \quad (3.50)$$

and two deputy SC which have the following LVLH coordinates relative to the chief

$$\begin{aligned} \boldsymbol{\rho}_1 &= \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}, & \dot{\boldsymbol{\rho}}_1 &= \begin{bmatrix} 0.003 \\ -0.012627 \\ 0 \end{bmatrix} \\ \boldsymbol{\rho}_2 &= \begin{bmatrix} 8 \\ 5 \\ 2 \end{bmatrix}, & \dot{\boldsymbol{\rho}}_2 &= \begin{bmatrix} 0 \\ -0.010102 \\ 0.010 \end{bmatrix} \end{aligned} \quad (3.51)$$

where we have suppressed the units with the understanding that $\boldsymbol{\rho}$ components are in km and $\dot{\boldsymbol{\rho}}$ components are in km/s. We have chosen these initial conditions such that they satisfy the no-drift condition Eq. (3.17); that is, both of these initial conditions lead to closed elliptical trajectories in the LVLH frame. The resulting free trajectories obtained by integrating the HCW equations are shown in Fig. 3.15.

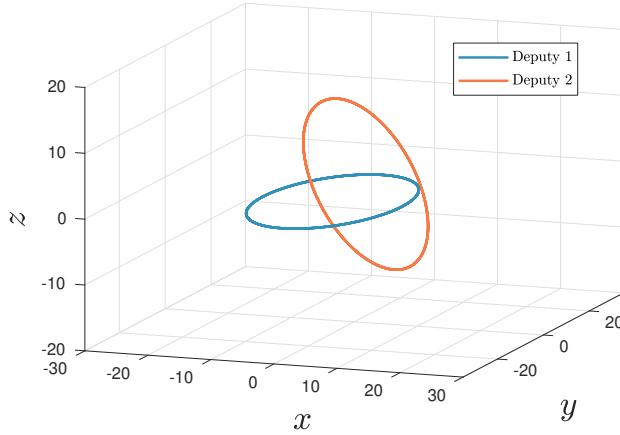


Figure 3.15: LVLH-frame free trajectories of the two deputy relative orbit configuration defined by Eqs. (3.50) and (3.51).

It will now be assumed that the deputies communicate as characterized by the Laplacian in Eq. (3.48), and this communicated information is utilized by the consensus control protocol

Eq. (3.46). We choose the gains such that the components of the HCW equations decouple, $K = [(A_1 + K_P) \ (A_2 + K_D)]$, with K_P and K_D diagonal:

$$K_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times 10^{-6}, \quad K_D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times 10^{-3}.$$

The closed-loop equations Eq. (3.49), with the same initial conditions as above, are now integrated to propagate the trajectories. These controlled trajectories are shown in Fig. 3.16, with the corresponding free trajectories in Fig. 3.15 superimposed for reference. We see that the consensus relative orbit trajectory achieved by both deputies lies between the two original trajectories at an intermediate orientation. This is analogous to the consensus state for double-integrator systems,

$$r_i = \bar{r} + \bar{v}t, \quad v_i = \bar{v}, \forall i = 1, 2, \dots, N,$$

where \bar{r} and \bar{v} are the weighted averages of the initial conditions of each agent.

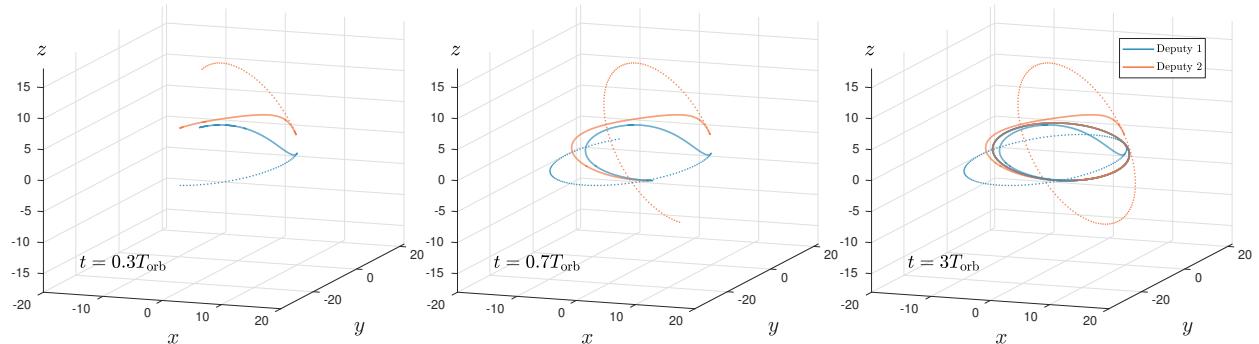


Figure 3.16: LVLH-frame controlled trajectories (*c.f.*, Fig. 3.15). We show the trajectory at three different times during the propagation to show how consensus reached. We plot the free trajectories of both deputies as dotted lines for reference.

Let us now repeat the above example with different initial conditions for the deputies. In this example we will assume that Deputy 1 satisfies the no-drift condition Eq. (3.17) while

Deputy 2 drifts relative to the LVLH frame. Specifically, we choose

$$\begin{aligned}\boldsymbol{\rho}_1 &= \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}, & \dot{\boldsymbol{\rho}}_1 &= \begin{bmatrix} 0.003 \\ -0.012627 \\ 0 \end{bmatrix} \\ \boldsymbol{\rho}_2 &= \begin{bmatrix} 8 \\ 5 \\ 2 \end{bmatrix}, & \dot{\boldsymbol{\rho}}_2 &= \begin{bmatrix} 0 \\ -0.011 \\ 0.010 \end{bmatrix}\end{aligned}\tag{3.52}$$

which are equal to the previous initial conditions Eq. (3.51) with the exception of \dot{y}_2 , which has been tuned slightly so that the no drift condition is violated.

The free and consensus-controlled trajectories are now shown in Fig. 3.17. We see that once consensus is reached between the two deputies, this consensus state drifts relative to the LVLH frame. This may be undesirable in scenarios where the chief is a real SC about which the deputies are needed to orbit. In scenarios where we simply need multiple SC to reach consensus, however, we can take the chief to be a *virtual* SC which simply defines the initial LVLH frame. Once consensus is reached among the (real) deputies, the LVLH frame is no longer needed as all deputies are following some identical (single) orbit trajectory.

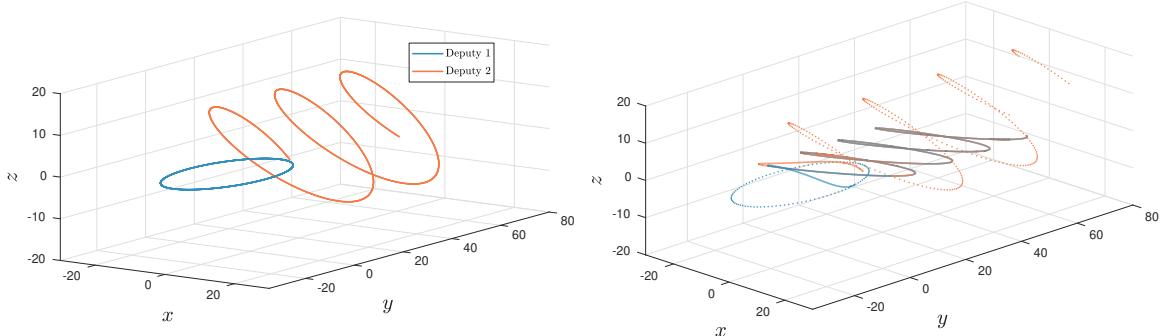


Figure 3.17: Free (left) and consensus-controlled (right) trajectories when Deputy 2 has non-zero initial drift in the LVLH frame. The consensus relative orbit now drifts relative to the LVLH frame. In the figure at right we also show the uncontrolled trajectories of both deputies as dotted lines for reference.

3.3.1 Arranging Spacecraft Formations Using Consensus Protocols

In spacecraft formation flying, a collection of spacecraft are required to maintain some relative spacial configuration while orbiting a common celestial body. We will now show how the HCW consensus protocol introduced above can achieve certain formation flying objectives.

A particularly interesting application of formation flying is the proposed Laser Interferometer Space Antenna (LISA) mission [52, 53], which will serve as a space-based gravitational wave (GW) observatory. This mission proposes to build a laser interferometer with arm-length on the order of 10^6 km by placing three individual spacecraft into a triangular constellation in heliocentric orbit. The relative orbits making up the constellation are such that the path-length between each spacecraft remains constant throughout the heliocentric orbit and the spacecraft form an equilateral triangle (see Fig. 3.18, taken from [52]). This mission is meant to function as a complementary observatory to Earth-based GW interferometers such as LIGO, which in September of 2015 made the first experimental observation of gravitational waves; these space-based gravitational wave detectors will be sensitive to a different bandwidth of the cosmic GW spectrum due to the longer arm length and absence of geoseismic background interference.

Due to the sensitivity of the interferometric observations, control thrusters are not intended to keep the spacecraft in formation. Rather, the spacecraft are to be placed in some relative orbit configuration which naturally maintains the inter-spacecraft distance as each SC individually orbits the central body. Onboard thrusters are therefore only used for periodic station-keeping maneuvers and for the initial maneuver to form the constellation.

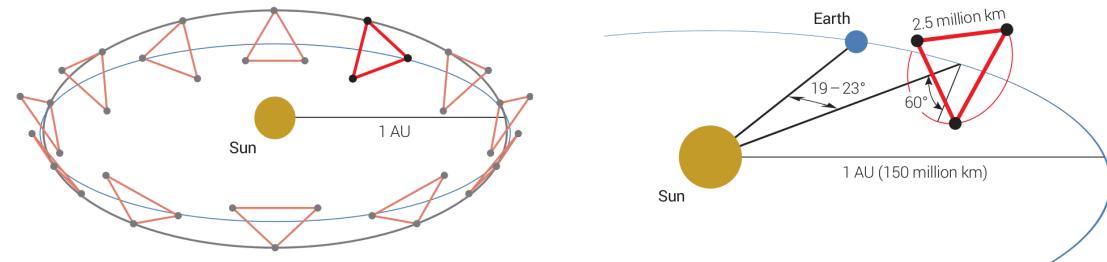


Figure 3.18: Illustration of the LISA gravitational wave observatory.

It will now be illustrated how the initial formation maneuver can be achieved using the consensus protocols introduced in the previous section. We imagine that a single rocket payload delivers the three spacecraft to the Earth-trailing heliocentric orbit illustrated in Fig. 3.18. This can be accomplished by a simple two-impulse phase-changing maneuver as follows: Let the first Δv impulse raise the heliocentric semi-major axis of the payload to a_{trans} . After a complete orbit, the second impulse is used to return the payload to the original semi-major axis. The payload will now trail the Earth by some phase. Taking this phase to be -20° , we find that the intermediate transfer orbit must have $a_{\text{trans}} \approx 1.04\text{AU}$. At this point, a controller is now implemented which brings the three separate SC from this common center to the necessary relative orbit configuration which maintains a constant inter-SC spacing.

Let us further assume that the inter-SC separation is small relative to the initial orbital radius of the delivered payload. This assumption is validated in the case of the proposed LISA mission, as the ratio of inter-SC separation to heliocentric orbit radius is on the order of 10^{-2} . The dynamics can therefore be approximated using the HCW equations, and using the initial orbit of the payload as the origin of the LVLH frame. This origin therefore can be thought of as a virtual chief and the constellation SC will be considered to be the deputies.

Our goal is to place each deputy in a relative orbit such that the distances between each deputy remains constant in the absence of controller thrust. This can be accomplished by placing each of the three spacecraft in a circular relative orbit about the chief. We recall the general solutions for a chief-deputy relative orbit derived in Eq. (3.16). For closed relative orbits, the initial conditions must satisfy the no-drift condition $\dot{y}_0 = -2nx_0$, and the general solution for the trajectory is given by

$$\begin{aligned}x(t) &= A_0 \cos(nt + \alpha) \\y(t) &= -2A_0 \sin(nt + \alpha) + y_0 \\z(t) &= B_0 \cos(nt + \beta).\end{aligned}$$

If we choose $y_0 = 0$, $\alpha = \beta$, and $B_0 = \sqrt{3}A_0$, the radial chief-deputy distance is given by

$$r(t) = \sqrt{x(t)^2 + y(t)^2 + z(t)^2} = \sqrt{4A_0^2 \cos^2(nt + \alpha) + 4A_0^2 \sin^2(nt + \alpha)} = \sqrt{4A_0^2} = 2A_0.$$

These choices therefore describe a circular relative-orbit trajectory [54] of constant radius $2A_0$. By evaluating the position at $t = 0$ with phase angle $\alpha = 0$, we see that the plane of the circular trajectory is oriented away from the in-plane direction by

$$\theta = \arctan \frac{z}{x} = \arctan \sqrt{3} = 60^\circ,$$

agreeing with the illustration in the right panel of Fig. 3.18.

It is therefore possible to form a triangular formation, with constant inter-SC distance, by placing three deputies at the LVLH coordinates

$$\left. \begin{array}{l} \boldsymbol{\rho}_1 = \boldsymbol{\rho}_{\text{circ}}(t; A_0, 0^\circ) \\ \boldsymbol{\rho}_2 = \boldsymbol{\rho}_{\text{circ}}(t; A_0, 120^\circ) \\ \boldsymbol{\rho}_3 = \boldsymbol{\rho}_{\text{circ}}(t; A_0, 240^\circ) \end{array} \right\} \quad \text{with} \quad \boldsymbol{\rho}_{\text{circ}}(t; A_0, \alpha) = \begin{bmatrix} A_0 \cos(nt + \alpha) \\ -2A_0 \sin(nt + \alpha) \\ \sqrt{3}A_0 \cos(nt + \alpha) \end{bmatrix}, \quad (3.53)$$

with corresponding velocities given by

$$\left. \begin{array}{l} \dot{\boldsymbol{\rho}}_1 = \dot{\boldsymbol{\rho}}_{\text{circ}}(t; A_0, 0^\circ) \\ \dot{\boldsymbol{\rho}}_2 = \dot{\boldsymbol{\rho}}_{\text{circ}}(t; A_0, 120^\circ) \\ \dot{\boldsymbol{\rho}}_3 = \dot{\boldsymbol{\rho}}_{\text{circ}}(t; A_0, 240^\circ) \end{array} \right\} \quad \text{with} \quad \dot{\boldsymbol{\rho}}_{\text{circ}}(t; A_0, \alpha) = \begin{bmatrix} -A_0 n \sin(nt + \alpha) \\ -2A_0 n \cos(nt + \alpha) \\ -\sqrt{3}A_0 n \sin(nt + \alpha) \end{bmatrix}. \quad (3.54)$$

Our goal is to now construct a controller which will deliver the individual SC from the origin of the LVLH coordinates (which we assume is defined by the orbit into which the bundled experiment was launched) to these circular trajectories. Since these desired trajectories are time dependent and not inertial, it is not obvious how this can be done with the regulation or tracking controllers we have previously discussed.

A simple solution to this control problem is to utilize a triplet of leader-follower consensus control protocols. Let us initially only consider SC1, described by the state \boldsymbol{x}_1 , and associate with this deputy a virtual deputy with state $\bar{\boldsymbol{x}}_1$. We will consider this deputy and its virtual counterpart to make up a 2-body cooperative system described by the state $\boldsymbol{X}_1 = [\bar{\boldsymbol{x}}_1; \boldsymbol{x}_1]$ with communication topology characterized by the Laplacian

$$L = \begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix}.$$

This Laplacian corresponds to a leader-follower system, with the virtual deputy taking the role of the leader. That is, regardless of the motion of the actual (physical) deputy, the

virtual deputy will move along unaffected according to its natural dynamics. By introducing the second-order consensus control protocol (in the local representation),

$$\dot{\mathbf{X}}_1 = \left((I_N \otimes A_{\text{CW}}) - (L \otimes BK) \right) \mathbf{X}_1, \quad (3.55)$$

the trajectory of the virtual and physical deputies will reach consensus, implying due to the leader-follower topology that the trajectory of the physical deputy will converge to the uncontrolled natural motion of the virtual deputy.

We can therefore initialize our virtual deputy at an appropriate point on the desired circular relative orbit. This virtual deputy will traverse this path through the natural HCW dynamics, providing the necessary control signal to drive the physical deputy to our desired circular relative orbit. That is, we initialize

$$\mathbf{x}_1(0) = \begin{bmatrix} \boldsymbol{\rho}_1(0) \\ \dot{\boldsymbol{\rho}}_1(0) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}, \quad \bar{\mathbf{x}}_1(0) = \begin{bmatrix} \bar{\boldsymbol{\rho}}_1(0) \\ \dot{\bar{\boldsymbol{\rho}}}_1(0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\rho}_{\text{circ}}(0; A_0, 0^\circ) \\ \dot{\boldsymbol{\rho}}_{\text{circ}}(0; A_0, 0^\circ) \end{bmatrix} \Rightarrow \mathbf{X}_1(0) = \begin{bmatrix} \bar{\mathbf{x}}_1(0) \\ \mathbf{x}_1(0) \end{bmatrix}.$$

where $\boldsymbol{\rho}_{\text{circ}}$ and $\dot{\boldsymbol{\rho}}_{\text{circ}}$ are given in Eqs. (3.53) and (3.54). We show in Fig. 3.19 the resulting evolution of the physical and virtual deputy trajectories using these initial conditions and the consensus control law Eq. (3.55). We use the nominal values of the LISA configuration geometry, so that an inter-SC separation of $|\Delta \mathbf{r}| = 2.5 \times 10^6$ km corresponds to a circular relative orbit of radius $r_{\text{circ}} = |\Delta \mathbf{r}| \sin(30^\circ) / \sin(120^\circ) \approx 1.44 \times 10^6$ km, which therefore gives a circular trajectory parameterized by $A_0 = r_{\text{circ}}/2 \approx 7.22 \times 10^5$ km. Note that we are defining the initial LVLH frame (the Earth-trailing orbit of the payload at the initial time of the constellation deployment) by $a = a_\oplus = 1.496 \times 10^8$ km, so that $n = \sqrt{\mu_\odot/a_\oplus^3} \approx 1.99 \times 10^{-7}$ s⁻¹. The gain matrices of the controller are designed through LQR optimization of the HCW system under PD control, using the optimization matrices

$$Q = \begin{bmatrix} 10I_3 & \mathbf{0} \\ \mathbf{0} & \frac{1}{n^2} I_3 \end{bmatrix}, \quad R = \frac{100}{n^4} I_3$$

The entire formation can now be built by using two additional copies of the above described controller. The equations of motion for these (virtual) leader-follower systems are given by Eq. (3.55), replacing \mathbf{X}_1 with \mathbf{X}_2 and \mathbf{X}_3 , initialized as

$$\mathbf{X}_2(0) = \begin{bmatrix} \boldsymbol{\rho}_{\text{circ}}(0; A_0, 120^\circ) \\ \dot{\boldsymbol{\rho}}_{\text{circ}}(0; A_0, 120^\circ) \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \quad \text{and} \quad \mathbf{X}_3(0) = \begin{bmatrix} \boldsymbol{\rho}_{\text{circ}}(0; A_0, 240^\circ) \\ \dot{\boldsymbol{\rho}}_{\text{circ}}(0; A_0, 240^\circ) \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}.$$

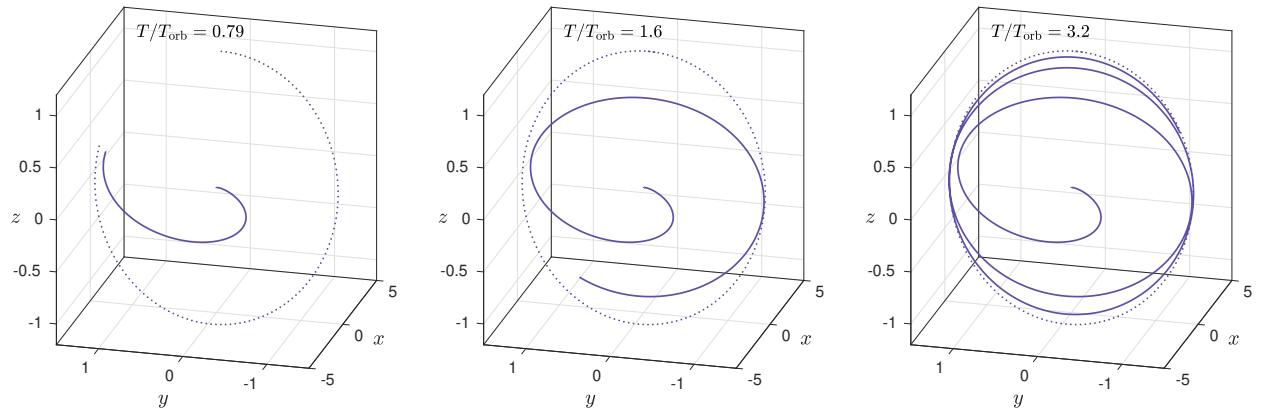


Figure 3.19: Placement of SC1 into the circular relative orbit using leader-follower consensus control. Solid line represents the physical deputy while dotted line represents the virtual (leader) deputy. Units are in 10^6 km.

The evolution of the entire constellation under these consensus controllers is shown in Fig. 3.20. We see that the control strategy discussed above successfully brings the constellation from the bundled payload starting point (the center of the LVLH frame) to the points maximally separated along a circular relative orbit. In Fig. 3.21 we show the evolution of the radial distances $|\boldsymbol{\rho}_i|$ and relative distances $\Delta r_{ij} = |\boldsymbol{\rho}_i - \boldsymbol{\rho}_j|$ of the three (physical) spacecraft in the formation as a function of time. We see from this figure that all spacecraft reach the desired radius $r_{\text{circ}} \approx 1.44 \times 10^6$ km and desired inter-SC separation of 2.5×10^6 km after approximately 4 orbits of the LVLH frame.

We note that this time frame corresponds to 4 solar orbits, or 4 years. A quicker response time can of course be achieved by increasing the gains of the controller appropriately, which can be achieved in an optimized way by increasing the LQR matrix elements of Q relative to those of R .

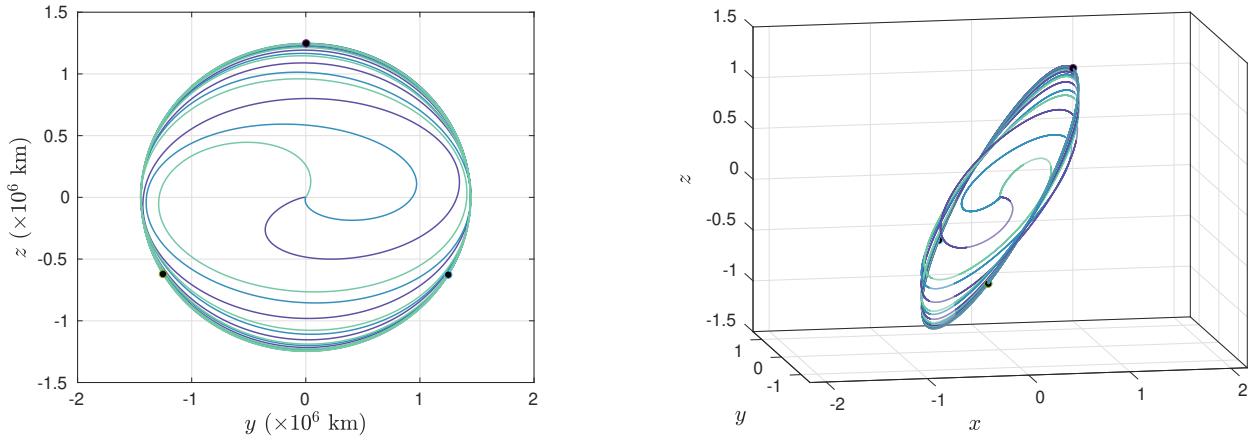


Figure 3.20: Evolution of the LISA constellation under the leader-follower consensus control laws of the form Eq. (3.55). The black dots indicate the positions of the constellation after 6 complete orbits of the LVLH frame. We see that these controllers successfully bring the constellation from the payload origin to the points maximally separated along a circular relative orbit.

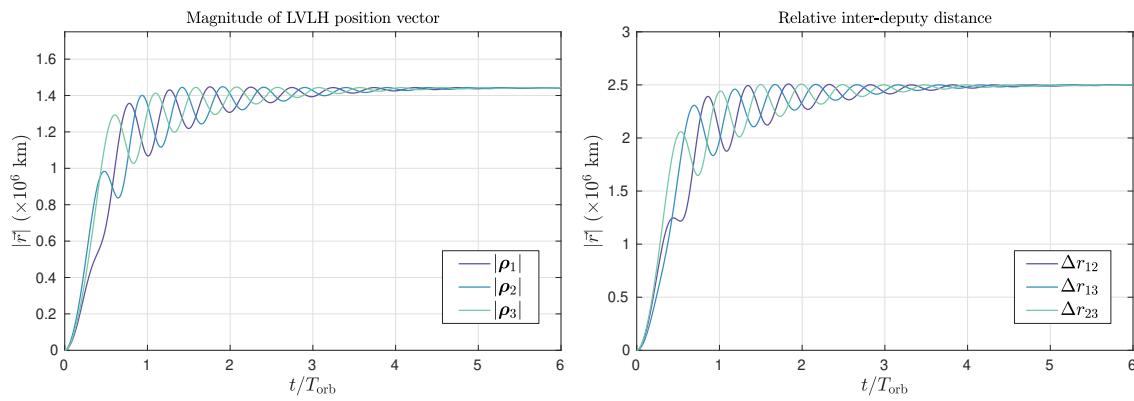


Figure 3.21: Radial distances of the three LISA constellation members from the LVLH origin (left). Shown at right are the relative distances between the constellation members. In both cases we see that these distances become constant (and equal) as the arrangement is formed.

CHAPTER 4

CONCLUSIONS

The automated control of multivehicle systems is a topic which has gained a considerable amount of interest in recent years. This work has presented a number of results relevant to this field, including several generalizations and applications of the standard consensus control protocol. In particular, in Chapter 2 the standard PD consensus controller was generalized to a PID consensus controller; it was proved that this controller is asymptotically stable as long as certain conditions on the controller gains and communication topology of the multiagent system are satisfied. To my knowledge these are novel results, and the stability conditions we derived for the controller gains simplify the tuning of such PID consensus controllers by eliminating unstable regions of parameter space. As a corollary to this latter result, we find that a fully-connected system under PID consensus control will be stable if $k_I < N k_P k_D$, where N is the number of agents in the system.

A further generalization of the consensus controller was achieved by introducing fractional integral and derivative operators into the consensus protocol. The particular *fractional consensus controller* introduced in Chapter 2 incorporates a derivative term whose order can be tuned between 0 and 1; the order of this fractional derivative thus serves as an extra “knob” that one can tune, allowing more freedom in how one can shape the response of the system. It was proved that, under the condition that all non-zero eigenvalues of the pseudostate system matrix satisfy $|\arg \lambda_i| > \alpha\pi/2$, the controller drives the multiagent system to consensus asymptotically. This condition on the non-zero eigenvalues was then related to the free parameters of the controller, $\{\alpha_D, k_P, k_I, k_D\}$, and stability conditions on the gains were similarly obtained.

The efficacy of the PID^α consensus controller was then demonstrated numerically by simulating the behavior of several benchmark 5-agent systems. For a 5-agent system with fully-connected communication topology, we have shown that the fractional PID^α outper-

forms the optimal integer-order PID controller (in terms of a few important performance specifications) by surveying over the controller parameter space. This provides a concrete demonstration that fractional controllers can outperform integer-order controllers for certain common performance measures.

Chapter 3 extended the concepts developed in Chapter 2 to multiagent systems obeying nontrivial dynamics. We considered two or more spacecraft in orbit about a common celestial body, and applied both fractional and consensus control to the *relative orbit* dynamics. In all cases considered we assumed that these relative orbit dynamics could be linearized to the HCW dynamics. Therefore the controller discussed in this Chapter are relevant to important applications such as rendezvous and the formation of small constellations of multiple spacecraft in orbit.

I have shown how one can apply fractional controllers to relative orbit trajectories obeying the HCW equations. It was again demonstrated that the main advantage of these types of controllers is the extra freedom in shaping the response afforded by the fractional derivative orders. This additional freedom allows one to reduce overshoot and settling time somewhat independently, in contrast to standard PD-type controllers where these response characteristics are inherently intertwined. Additionally, it was demonstrated that the creeping behavior exhibited by the controlled trajectories (and typically exhibited by solutions to fractional differential equations) can potentially be advantageous in rendezvous/docking maneuvers, where it is desirable for the final stage of approach to be performed at low velocity and from a constant angular direction. Proof-of-principle simulations of trajectories under these control protocols were performed and discussed.

In the case of consensus control for HCW dynamics, we have not provided a proof of stability, nor attempted to derive the resulting consensus state (the consensus relative orbit trajectory) as a function of initial conditions (as was done for double-integrator systems in Chapter 2). The ability to predict stability and/or the resulting consensus state has obvious importance in practical application. These seem like very tractable problems, and are the obvious next step in the study of relative orbit consensus control.

There are a number of other issues that were uncovered throughout the course of this research which oblige further study. The sensitivity of the LQR-designed controller to the y -component derivative is curious, and the reason for this sensitivity is not presently understood. We can possibly attack this problem through an inspection of the closed-loop transfer functions from the components of \mathbf{u} to the components of $\boldsymbol{\rho}$ [55], and observing how these transfer functions are affected as one alters the derivative term (*i.e.*, takes $s \rightarrow s^\alpha$).

Additionally, in Chapters 2 and 3 we resorted to brute-force numerical surveys over the controller parameter space in order to find optimal fractional controllers. Of course, an analytical optimization technique analogous to LQR design for integer-order LTI controllers would in these cases be highly desirable. Preliminary investigation of these topics was carried out, the results of which were not incorporated into this work. The situation is, not surprisingly, quite a bit more complicated. Nonetheless, solid foundational work has already been carried out in the development of the necessary *variational fractional calculus*, in large part by Agrawal [56], and the application of this formalism has been applied to certain optimal fractional control problems [57]. In general, however, it appears to be that optimal fractional control problems are often intractable due to the much more complicated structure of the *transversality conditions*. It seems that the underlying reason rests upon the fact that the fractional derivative is nonlocal operator (as discussed in Chapter 1), which in turn leads to the fact that the Hamiltonian for an autonomous fractional-order system is no longer conserved. Based on the numerical optimization results reported in this thesis, further exploration along this path could potentially be a very rewarding endeavor.

REFERENCES

- [1] D. Yaylali, E. A. Butcher, and A. Dabiri, *Fractional PID Consensus Control Protocols for Second-Order Multiagent Systems*. San Diego, CA: AIAA Guidance, Navigation, and Control Conference, 2019. Accepted for inclusion, June 2018.
- [2] D. Yaylali, E. A. Butcher, and A. J. Sinclair, *Fractional Control Protocols in Linearized Relative Orbit Dynamics*. Ka'anapali, HI: AAS/AIAA Space Flight Mechanics Meeting, 2019. Accepted for inclusion, October 2018.
- [3] W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control,” *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, 2007.
- [4] J. A. Fax and R. M. Murray, “Information flow and cooperative control of vehicle formations,” *Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.
- [5] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative Control of Multi-Agent Systems*. New York: Springer, 2014.
- [6] W. Ren and E. M. Atkins, “Distributed multi-vehicle coordinated control via local information exchange,” *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10, pp. 1002–1033, 2007.
- [7] W. Ren, “On consensus algorithms for double-integrator dynamics,” *Transactions on Automatic Control*, vol. 53, pp. 1503–1509, July 2008.
- [8] T. Yucelen and E. N. Johnson, “Control of multivehicle systems in the presence of uncertain dynamics,” *International Journal of Control*, vol. 86, no. 9, pp. 1540–1553, 2013.
- [9] N. Wheeler, “Construction and physical application of the fractional calculus,” February 1997. Reed College lecture notes.
- [10] Y. Chen, I. Petras, and D. Xue, “Fractional order control – a tutorial,” *2009 American Control Conference*, pp. 1397–1411, 2009.
- [11] K. B. Oldham and J. Spanier, *The Fractional Calculus*. San Diego, California: Academic Press, Inc., 1974.
- [12] R. E. Gutierrez, J. M. Rosario, and J. T. Machado, “Fractional order calculus: Basic concepts and engineering applications,” *Mathematical Problems in Engineering*, vol. 3758, 2010.

- [13] R. L. Bagley and R. A. Calico, “Fractional order state equations for the control of viscoelastically damped structures,” *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 2, pp. 304–311, 1991.
- [14] A. Oustaloup, *La commande CRONE: Commande Robuste d’Ordre Non Entier*. Paris: Hermé’s, 1991.
- [15] I. Podlubny, “Fractional-order systems and $\text{PI}^\lambda \text{D}^\mu$ -controllers,” *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 208–214, 1999.
- [16] D. Matignon, “Stability results for fractional differential equations with applications to control processing,” *Computational Engineering in Systems Applications*, pp. 963–968, July 1996.
- [17] D. Xue, C. Zhao, and Y. Chen, *Fractional order PID control of a DC-motor with elastic shaft: a case study*. Minneapolis, MN: Proceedings of the American Control Conference, 2006.
- [18] E. A. Butcher, M. Nazari, A. Dabiri, and A. K. Sanyal, *Fractional PID Control of Spacecraft Attitude Dynamics using Rotation Matrices*. IAC, Guadalajara: Proceedings of the International Astronautical Congress, 2016.
- [19] M. Nazari, E. A. Butcher, and A. K. Sanyal, “Spacecraft attitude fractional feedback control using rotation matrices and exponential coordinates,” *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 2185–2198, 2018.
- [20] M. Nazari, E. A. Butcher, and A. K. Sanyal, *Fractional Control of Rigid Body Attitude Dynamics using Exponential Coordinates*. Kissimmee, FL: AIAA Guidance, Navigation, and Control Conference, 2018.
- [21] C. Song and J. Cao, “Consensus of fractional-order linear systems,” *9th Asian Control Conference*, pp. 1–4, 2013.
- [22] T. Yucelen and M. Egerstedt, “Control of multiagent systems under persistent disturbances,” *2012 American Control Conference*, pp. 5264–5269, 2012.
- [23] W. L. Brogan, *Modern Control Theory*. Upper Saddle River, NJ: Prentice Hall, 3 ed., 1991.
- [24] H. Goldstein, *Classical Mechanics*. Menlo Park, Ca: Addison-Wesley, 2 ed., 1980.
- [25] D. E. Kirk, *Optimal Control Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1970.
- [26] J. S. Caughman, G. Lafferriere, J. J. P. Veerman, and A. Williams, “Decentralized control of vehicle formations,” *Systems and Control Letters*, vol. 54, no. 9, pp. 899–910, 2005.

- [27] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*. London: Springer-Verlag, 2008.
- [28] K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*. New York, New York: John Wiley & Sons, Inc., 1993.
- [29] I. Podlubny, *Fractional Differential Equations*. San Diego: Academic Press, 1999.
- [30] Y. Li, Y. Chen, and I. Podlubny, “Stability of fractional-order nonlinear dynamic systems: Lyapunov direct method and generalized Mittag–Leffler stability,” *Computers and Mathematics with Applications*, vol. 59, pp. 1810–1821, 2010.
- [31] D. Valerio and J. S. da Costa, *An Introduction to Fractional Control*. London: The Institution of Engineering and Technology, 2013.
- [32] R. Caponetto, G. Dongola, L. Fortuna, and I. Petras, *Fractional Order Systems: Modeling and Control Applications*. New Jersey: World Scientific, 2010.
- [33] C. A. Monje, Y. Q. Chen, B. M. Vinagre, D. Xue, and V. Feliu, *Fractional-order Systems and Controls: Fundamentals and Applications*. New York: Springer, 2010.
- [34] E. H. Doha, A. H. Bhrawy, and S. S. Ezz-Eldien, “Efficient Chebyshev spectral methods for solving multi-term fractional orders differential equations,” *Applied Mathematical Modelling*, vol. 35, no. 12, pp. 5662–5672, 2011.
- [35] A. Dabiri and E. A. Butcher, “Stable fractional Chebyshev differentiation matrix for numerical solution of fractional differential equations,” *Nonlinear Dynamics*, vol. 90, no. 1, pp. 185–201, 2017.
- [36] A. Dabiri and E. A. Butcher, “Efficient modified Chebyshev differentiation matrices for fractional differential equations,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 50, pp. 584–610, 2017.
- [37] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*. Moneola, New York: Dover, 2000.
- [38] A. Dabiri and E. A. Butcher, “Numerical solution of multi-order fractional differential equations with multiple delays via spectral collocation methods,” *Applied Mathematical Modeling*, vol. 56, pp. 424–448, 2018.
- [39] A. Dabiri, E. A. Butcher, M. Poursina, and M. Nazari, “Optimal periodic-gain fractional delayed state feedback control for linear fractional periodic time-delayed systems,” *Transactions on Automatic Control*, vol. 63, no. 4, pp. 989–1002, 2018.
- [40] E. A. Butcher, A. Dabiri, and M. Nazari, *Time Delay Systems: Theory, Numerics, Applications, and Experiments*, vol. 7 of *Advances in Delays and Dynamics*, ch. Stability and Control of Fractional Periodic Time-Delayed Systems. 2017. Invited chapter.

- [41] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*. San Francisco: McGraw-Hill Companies, Inc., 1997.
- [42] K. Diethelm and N. J. Ford, “Numerical solution of the Bagley-Torvik equation,” *BIT Numerical Mathematics*, vol. 42, no. 3, pp. 490–507, 2002.
- [43] A. Dabiri and E. A. Butcher, *Design of Optimal Fractional Luenberger Observers for Linear Fractional-Order Systems*. ASME, Cleveland, OH: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2017.
- [44] I. N’doye, M. Zasadzinski, M. Darouach, and N. Radhy, “Observer-based control for fractional-order continuous-time systems,” in *Proceedings on the Joint 48th IEEE Conference on Decision and Control*, (Shanghai, China), pp. 1932–1937, 2010.
- [45] R. Garrappa, “On linear stability of predictor-corrector algorithms for fractional differential equations,” *International Journal of Computer Mathematics*, vol. 87, no. 10, pp. 2281–2290, 2010.
- [46] H. Schaub and J. Junkins, *Analytical Mechanics of Space Systems*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2009.
- [47] K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. S. Breger, *Spacecraft Formation Flying*. Oxford, UK: Elsevier Ltd., 2010.
- [48] W. H. Clohessy and R. S. Wiltshire, “Terminal guidance system for satellite rendezvous,” *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.
- [49] P. Sengupta and S. R. Vadali, “Relative motion and the geometry of formations in Keplerian elliptic orbits,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 953–964, 2007.
- [50] A. J. Sinclair, R. E. Sherrill, and T. A. Lovell, “Calibration of linearized solutions for satellite relative motion,” *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1362–1367, 2014.
- [51] A. J. Sinclair, R. E. Sherrill, and T. A. Lovell, “Use of Cartesian-coordinate calibration for satellite relative-motion control,” *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 9, pp. 1842–1847, 2015.
- [52] P. Amaro-Seoane *et al.*, “Laser Interferometer Space Antenna,” *ArXiv e-prints*, Feb. 2017.
- [53] P. Amaro-Seoane *et al.*, “eLISA/NGO: Astrophysics and cosmology in the gravitational-wave millihertz regime,” *GW Notes*, vol. 6, pp. 4–110, 2013.

- [54] S. S. Vaddi, S. R. Vadali, and K. T. Alfriend, “Formation flying: Accommodating non-linearity and eccentricity perturbations,” *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 214–223, 2004.
- [55] A. J. Sinclair. private communication.
- [56] O. P. Agrawal, “A general formulation and solution scheme for fractional optimal control problems,” *Nonlinear Dynamics*, vol. 38, no. 1-4, pp. 323–337, 2004.
- [57] S. Pooseh, R. Almeida, and D. F. M. Torres, “Fractional Order Optimal Control Problems with Free Terminal Time,” *ArXiv e-prints*, Feb. 2013.