

Bicycle dynamics: modelling and experimental validation

By

DALE LUKAS PETERSON

B.S. (University of California Santa Barbara) 2004

M.S. (University of California Davis) 2007

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Chair Mont Hubbard, Chair

Ronald A. Hess

Fidelis O. Eke

Committee in Charge

2013

Copyright © 2013 by
Dale Lukas Peterson
All rights reserved.

To my family.

CONTENTS

List of Figures	v
List of Tables	vi
Abstract	vii
Acknowledgments	viii
1 Introduction	1
2 Mathematical model of bicycle motion	4
2.1 Introduction	4
2.2 Bicycle parameters	4
2.2.1 Meijaard parameters	5
2.2.2 Cylindrical gyrostat	7
2.2.3 Bicycle gyrostat parameters	8
2.2.4 Parameter conversion	10
2.3 Kinematics	12
2.4 Dynamics	18
3 Linearization of equations of motion of constrained multibody systems	24
3.1 Introduction	24
3.2 Kane's method, briefly	27
3.3 Rolling disk	33
3.4 Derivation of linearization procedure	38
3.5 Rolling disk, revisited	42
3.6 Discussion	43
3.7 Conclusions	45
4 Robot bicycle	50
4.1 Introduction and motivation	50
4.2 Mechanical construction	51
4.2.1 Design	51
4.2.2 Physical parameter measurement	56

4.3	Electrical system description	57
4.3.1	Microcontroller	57
4.3.2	Sensors	58
4.3.3	Actuators	63
4.3.4	Batteries	65
4.3.5	Wireless communication	65
4.4	Control system design	66
4.4.1	Data logging	66
4.5	Feedback control system	67
4.5.1	Rear wheel rate controller	67
4.5.2	Yaw rate controller	68
4.5.3	User interface	70
4.6	Experiments	71
4.6.1	Example results	73
4.6.2	Discussion	75
4.6.3	Conclusions	80
5	Conclusions and suggestions for future work	88

LIST OF FIGURES

2.1	Bicycle gyrostat parameters.	10
4.1	Robot bicycle viewed from the right side.	51
4.2	Robot bicycle viewed from the rear right side.	52
4.3	Robot bicycle viewed from the rear.	53
4.4	Robot bicycle viewed from the rear left.	54
4.5	Battery plate, fork motor mount, and digital motor drives.	55
4.6	Calibration of steer angle encoder, accelerometer, and rate gyroscope.	60
4.7	Rate gyroscope and accelerometer sensor placement.	61
4.8	Yaw rate control block diagram.	69
4.9	Taking the robot bicycle on a jog.	72
4.10	Run 000 speed and steer response.	75
4.11	Run 000 lean rate measured and estimated.	76
4.12	Run 000 steer rate measured and estimated.	76
4.13	Run 000 rear wheel and steer torque response.	77
4.14	Steer, lean rate, steer torque time response at 1.0 m s^{-1} .	78
4.15	Steer and lean rate frequency response at 1.0 m s^{-1} .	79
4.16	Steer, lean rate, steer torque time response at 4.0 m s^{-1} .	81
4.17	Steer and lean rate frequency response to $v_c = 2.0 \text{ m s}^{-1}$.	82
4.18	Steer, lean rate, steer torque time response at 4.0 m s^{-1} .	83
4.19	Steer and lean rate frequency response at 4.0 m s^{-1} .	84

LIST OF TABLES

2.1	Bicycle gyrostat parameters	9
2.2	Benchmark bicycle parameters converted to gyrostat parameters.	13
2.3	Bicycle generalized coordinates	16
3.1	Constrained multibody system governing definitions and equations	33
4.1	Robot bicycle physical parameters	56
4.2	Microcontroller development board functionality.	57
4.3	Summary of experiments.	72
4.4	Mean and standard deviation of speed, steer, lean rate, and steer torque.	82

ABSTRACT

Bicycle dynamics: modelling and experimental validation

This dissertation explores bicycle dynamics through an extension of the Whipple bicycle model and validation of the model equations of motion through the implementation of a robotic bicycle. An extended Whipple bicycle model is presented which makes uses of a unique set of physical parameters based on cylindrical gyrostats. The nonlinear equations of motion for this model are derived, linearized, and validated against a set of benchmark model parameters. A general formulation for the linearization of a system with configuration and velocity constraints is presented, and is demonstrated on an idealized rolling disk. The method of linearization is directly applicable to the equations of motion which result from the application of Kane's method. The linearization procedure is used to formulate the linear state space equations of motion for the bicycle model, which are then used as the plant model to design the robotic bicycle control system. The mechanical, electrical, and software aspects of the robotic bicycle are presented, along with representative results from a set of experiments.

ACKNOWLEDGMENTS

This work would not have been possible without the support of many people. First and foremost, I would like thank my parents, John and Lyn. They have been there for me at every stage of my life and I am thankful for their support, love, care, and for inspiring me to be inquisitive about the world around me. Next, I would like to thank my fiancée Cassandra Ann Paul, who has been essential to this dissertation. Cassandra has encouraged me when I needed it most, chastized me when I deserved it, and kept me positive throughout a very stressful final year of graduate school. Cassandra is a reminder of what is important in life, and I look forward to building our lives together.

I have had many excellent teachers in my life and this dissertation is a reflection of them me. I would like to thank Joan Owen, Joyce Hodgkinson, and Elizabeth Terwilliger for their thoughtful and forward thinking work during my early years at Tam Creek School. They planted the seeds of civic duty and environmental stewardship in many young children and I am thankful to have been one of their students. I would like to thank my high school chemistry teacher, Nina Tychinin, for encouraging me to explore my interests in science. Near the end of my undergraduate education, after returning from a two year activation with the United States Marine Corps, I enrolled in an applied dynamics course with Dr. Steven Shaw and a control systems course with Dr. João P. Hespanha. These two teachers are directly responsible for igniting my fascination with dynamics and control, and I would like to thank them for careful and inspired presentation of these two beautiful subjects.

I would like to thank my Ph.D. advisor Dr. Mont Hubbard for his patience and clarity of thought, and for the many discussions about dynamics, control, and bicycles. It has been a true pleasure to work with Mont and I am grateful to have learned so much from him; I hope I was as good a student to him as he was an advisor to me. I would like to thank Dr. Ron Hess for his support and feedback throughout the period of the NSF grant and specifically for his help with design of the control system of the robotic bicycle. I would like to thank Dr. Fidelis Eke for his extremely clear presentation of multibody dynamics and Kane's method in particular. I would like to thank Dr. James Crutchfield for his inspirational course on nonlinear dynamics in which I was first exposed to the Python programming language.

The work of Dr. Arend L. Schwab, Dr. Jeremy Papadopoulos, Dr. Andy Ruina, and Dr. Jaap Meijaard has had a large impact on this work. Their clarity of presentation and the

many thoughtful discussions we have had about bicycle dynamics and control have significantly improved my understanding of bicycles and dynamics. I would like to extend special thanks to Dr. Arend Sewab and Dr. Jeremy Papadopoulos for the many in-depth discussions and emails we have had. It is my hope that the work in this dissertation is an important addition to the groundwork that you and many others have laid.

There are several students I would like to thank. First and foremost, I would like to thank my friend Jason Moore. Jason has enriched my life both inside and outside the walls of the academy (which we are both trying to tear down). I have enjoyed discussing bicycles, religion, computers, math, and philosophy with Jason and I credit him with keeping me on my toes and testing my views on the world. Thomas Johnston was a great companion in graduate school, and I am thankful for the many hours of discussion of dynamics we have had. I would like to thank all of the members, past and present, of the Sports Biomechanics Laboratory.

I would like to thank Ondřej Čertík for his substantial commitment to symbolic mathematics with the SymPy project and for helping me build a tool with which to study dynamics. Ondřej has demonstrated excellent leadership and has created a wonderful community for symbolic mathematics. His selflessness and commitment have helped to make my dissertation possible.

In the summer of 2011 Gilbert Gede and I worked closely to add classical mechanics functionality to the SymPy project. It was also during this time that the ideas for Chapter 3 of this dissertation were planted and initially developed. Gilbert's hard work and collaboration on the development of these ideas is greatly appreciated.

A large number of people contributed to the success of the robot bicycle. Armando Lee provided crucial welded several parts and operated the wire electronic discharge machine for several parts of the robotic bicycle. Benny Brown provided hours of much need electrical engineering experience and know how. Kenny Koller helped me understand embedded systems and provoked a lot of thought about careful design of software which interacts directly with hardware resources. Kenneth Lyons helped with steer angle calibration, and Derek Pell helped fabricate a number of components used for measuring the physical parameters of the bicycle. I appreciate all of your help.

Oliver Z. Lee deserves special recognition for his help with the robot bicycle. Oliver and I worked shoulder to shoulder for the first 7 months of 2013 and we came to know each other quite well. Oliver's contribution to this dissertation was significant and essential; in particular, the

robot bicycle would not have balanced had it not been for his hard work, focus, and impressive skills in software and control systems.

Finally, I would like to thank Martin Hansen, Tony Merz, Ron Blinn, Erik Gambera, Allen Donhauser, Red Ross, Dan Novak, Dan Freudenberg, and Steve Wyrostok from Sunshine Bicycle Center for their support and the many interesting discussions about bicycle we had. My fascination with bicycles was kindled by these discussions.

This material is partially based upon work supported by the National Science Foundation under Grant No. 0928339. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Chapter 1

Introduction

As early as 3350BC, humans utilized wheels to ease their lives. For more than five thousand years, wheels were used for many things in many ways, but it wasn't until 1817 when Karl von Drais invented the "running machine": the first single-track two-wheeled balancing vehicle. Developments to the "running machine", notably pneumatic tires (1845), pedalled drive trains (1866), steel spoked wheels (1868), two-speed hub gear (1896), and butted frame tubes (1897) resulted in what we would comfortably call a bicycle [3]. The technology of the bicycle has been under constant development, but by the early 1900's the form of the bicycle had stabilized to a large degree. A mathematical framework for the motion of bicycle began around this same time [2]. Since Whipple's work, hundreds of authors have approach the analysis of bicycles and motorcycles, with many different goals: some to simply understand how this amazing machine can balance, others to understand how humans are able to control bicycles, and still others to understand how the design of the vehicle can be adjusted to meet some desired requirements.

It took more than 100 years for the academic community to come to agreement on how the model presented by Whipple in 1899 should behave. This model makes a large number of simplifying assumptions to render the analysis tractable. Models which include many more degrees of freedom or have less restrictive modelling assumptions are much less well understood. The benchmark paper by Meijaard [1] went a long way to reducing the fallacies and misinformation about how bicycles behave. The same type of benchmarking needs to be applied to more complicated models before there can be any hope of having an intelligent discussion of their dynamics.

The first goal of this dissertation is to improve the understanding understanding of mechanistic aspects of bicycle modelling, in particular the Whipple model. Chapter 2 presents the

derivation of nonlinear equations of motion for the Whipple bicycle model which have been verified to give identical results as those results presented by well established researchers in the field. While the results are equivalent to previous work, the derivation utilizes a unique choice of parameters which offer several advantages over other common choices. In particular, fewer parameters are need to describe the same class of bicycles, the parameters are not coupled as are the common choice of parameters, and the parameters exploit the front/rear geometric symmetry of the bicycle to streamline the derivation and clarify the resulting dynamic equations.

The second goal of this dissertation is to present a clear and explicit treatment of the linearization of the nonlinear equations of motion for systems with configuration and velocity constraints. The bicycle is such a system, and the work of [Chapter 3](#) generalizes techniques which were initially developed to linearize the bicycle dynamic equations. The techniques presented in [Chapter 3](#) address common tasks such as stability analysis and formulation of correct linearized state space equations for use in control system design. With substantial contributions from Gilbert Gede, this chapter has been submitted for publication and is pending review. All of the techniques presented in [Chapter 3](#) have been applied to the Whipple bicycle model and the results have been validated numerically.

Finally, the modelling techniques of [Chapter 2](#) and the linearization techniques of [Chapter 3](#) are applied to the non trivial task of building a robot bicycle which permits experimental validation of the Whipple model. Only a handful of studies to date have performed rigorous experimental validation of the Whipple model. In particular, none of the studies to date tackle the issue of experimentally validating the dynamic behavior outside the stable speed range. The real time control system (implemented using the techniques of [Chapter 2](#) and [Chapter 3](#)) permits exactly this.

It is my sincere hope that this work is useful to students and researchers of dynamics, control, and those interested in two-wheeled vehicles. The bicycle has provided me with a fertile playground for all of these topics and I have no doubt it will do the same for many others for many years to come.

References for Chapter 1

- [1] Jaap P. Meijaard, Jeremy M. Papadopoulos, Andy Ruina, and Arend L. Schwab. “Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463.2084 (Aug. 2007), pages 1955–1982. ISSN: 1364-5021. DOI: [10.1098/rspa.2007.1857](https://doi.org/10.1098/rspa.2007.1857). URL: <http://rspa.royalsocietypublishing.org/cgi/doi/10.1098/rspa.2007.1857> (cited on page 1).
- [2] Francis J. W. Whipple. “The stability of the motion of a bicycle”. In: *Quarterly Journal of Pure and Applied Mathematics* 30 (1899), pages 312–348 (cited on page 1).
- [3] David Gordon Wilson. *Bicycle Science*. 3rd. Cambridge, MA, USA: The MIT Press, 2004 (cited on page 1).

Chapter 2

Mathematical model of bicycle motion

2.1 Introduction

The bicycle model we use assumes the bicycle is composed of four rigid bodies: two wheels, a frame, and a fork. These rigid bodies are assumed to be connected by three revolute joints: one between the rear wheel and frame, one between the front wheel and fork, and one between the bicycle frame and the fork. Three internal torques, one for each revolute joint, are applied between each respective body; these torques can be used to encompass the effects of friction, motor torques, or braking torques. The rigid wheels are modelled as tori which make point contact with the ground plane. The frame and fork are each assumed to be inertially symmetric about their respective XZ planes. The revolute joint connecting the frame and fork is assumed to be parallel to the Z axis of each respective body. It is assumed that each wheel mass center lies in the XZ symmetry plane of the body to which it is connected, and that each wheel revolute joint axis is parallel to the Y axis of each respective body.

This chapter is divided into three sections. [Section 2.2](#) describes the parameters used to describe the physical characteristics of the four rigid bodies and lays the ground work for the kinematics ([section 2.3](#)) and the dynamics ([section 2.4](#)).

2.2 Bicycle parameters

The physical parameters used to describe the four rigid bodies, their interface with each other and the ground plane is of practical concern for several reasons. The choice of parameters determines how many quantities must be measured or calculated when characterizing a real bicycle; some parameters are more difficult to measure than others. The choice of parameters has a direct effect on the complexity of the equations of motion [4], [14] and, by extension, the

computational cost associated with simulating or performing stability analysis of the equations. Most importantly, the choice of parameters can greatly affect the ease of understanding how changing a single parameter affects the dynamics. Finally, having a common, agreed-upon set of parameters which permit direct comparisons is essential for communicating results – if everybody uses different parameters to describe the same model, comparisons of results become difficult and error prone.

This section provides a discussion of the bicycle parameters presented in [3], which we refer to as the “Meijaard parameters”, and motivates the use of another set of parameters which we refer to as the “gyrostat parameters”. The gyrostat parameters address several deficiencies with the Meijaard parameters while simultaneously reducing the number of parameters needed to describe the same four rigid bodies. This reduction is possible because the inertial properties of rear frame and rear wheel are independent of the orientation of the rear wheel relative to the rear frame (and similarly for the front fork and front wheel). Finally, we present the conversion from the Meijaard parameters to the gyrostat parameters.

2.2.1 Meijaard parameters

The Meijaard parameters have been adopted as the *de facto* standard by many authors [11]. They can be measured reasonably simply [6], and many of the parameters are familiar to those outside academic circles (i.e., the bicycle industry and everyday cyclists). Examples of commonly recognizable parameters available at your local bike shop include wheelbase w , trail c , and steer axis tilt λ . For these reasons, this choice of parameters is indispensable. Twenty five parameters are used to describe the geometric and inertial properties of the four rigid bodies and their connection to each other.

However, there are strong reasons to prefer other choices of parameters. The Meijaard parameters are ideally suited to direct derivation of linearized equations of motion, about the reference configuration (lean and steer equal to zero). The reason for this is that the Meijaard parameters are defined with respect to the bicycle *reference configuration* and with respect to a set of body fixed coordinates that are aligned with the inertial frame *only in this configuration*. Deriving nonlinear equations of motion with this parameter set is cumbersome and requires a number of intermediate geometric quantities to be introduced.

Perhaps the most serious disadvantage of the Meijaard parameter set is the coupling between the parameters. Consider, for example, a parameter study investigating the effect of front wheel

radius on stability. Using the Meijaard parameters, a naive approach might be to choose a set of parameters and compare the eigenvalues for that set with the eigenvalues when only the wheel front wheel radius is changed. Unfortunately, this does not represent the act of taking a real bicycle (with some set of parameters) and simply changing the front wheel to one with identical mass and inertia but different radius. Changing the front wheel radius of a real bicycle changes the steer axis tilt, wheelbase, trail, center of mass locations relative to the rear wheel contact, and, by virtue of the inertia scalars being defined relative to an inertial frame, six of the inertia scalars. Thus, if the goal of an analyst or designer is to understand how front wheel radius affects a bicycle stability (a reasonable goal), and that person uses the Meijaard parameter set to describe the bicycle, no fewer than *thirteen* additional parameters must be adjusted: the wheelbase w , trail c , steer axis tilt λ , the central inertia scalars I_{Bxx} , I_{Bzz} , I_{Bxz} , I_{Hxx} , I_{Hzz} , I_{Hxz} , and the scalars describing the center of mass locations of the bicycle frame and the fork x_B , z_B , x_H , and z_H .

This coupling of parameters is the result of defining the parameters with respect to the reference configuration. If the naive approach is used anyway, and all other parameters are left unchanged as in [5], [13], it must be realized that when comparing results which have only one of these coupled parameters changed (e.g., front wheel radius), one is actually comparing bicycles with different frame and fork geometry, center of mass location, and mass distribution since these must be changed (on a real bicycle) in order to keep the Meijaard parameters constant. Stated simply, one is not comparing effect of only front wheel radius, but the effect of changing the front wheel radius and thirteen other parameters. The practical utility of such a comparison is dubious at best. While this issue *can* be addressed by carefully changing the thirteen coupled parameters, to our knowledge, no parameter study to date has done this. This is a rarely stated, but critical, drawback of the Meijaard parameters.

The easiest way to remedy the issue of parameter coupling is to choose physical parameters which can be defined independent of configuration. This is standard practice in robotics: robot link lengths are defined relative to the previous link in the chain, and mass and inertia properties of each link are defined with respect to a link-fixed coordinate system which is independent of overall robot configuration. The thirteen Meijaard parameters which are defined relative to the reference configuration can be arranged into three groups and addressed separately. First, the wheelbase w , trail c , and steer axis tilt λ can be replaced with three distances which are

independent of configuration, as in [1]. Two of these measure the perpendicular distance between the wheel centers and the steer axis while a third measures the distance parallel to the steer axis between the first two lines. Second, the bicycle frame and fork center of mass locations relative to the rear wheel ground contact (x_B, z_B, x_H, z_H in [3]) can be replaced with parameters which are defined relative to the rear and front wheel centers, respectively. Finally, the central inertia scalars of the bicycle frame and fork ($I_{Bxx}, I_{Bzz}, I_{Bxz}, I_{Hxx}, I_{Hzz}, I_{Hxz}$ in [3]) can be replaced with inertia scalars defined relative to body-fixed coordinate systems which are aligned with features fixed in the body (such as the steer and wheel axes).

2.2.2 Cylindrical gyrostat

A gyrostat is a mechanical system of one or more bodies which has the rigid body property that its inertia scalars are time independent constants [14]. The most common example of such a system is a motor with a rotor which is inertially symmetric about the spin axis. As the rotor rotates relative to the frame of the motor, neither the location of the mass center nor the combined inertia of the system changes. This type of gyrostat is referred to as a cylindrical gyrostat [4]. The two-body cylindrical gyrostat is typically described as being composed of a carrier and a rotor. A bicycle wheel together with a frame or fork is a cylindrical gyrostat.

To aid the discussion which follows, we present the relationship between the parameters which fully describe a cylindrical gyrostat and the parameters which fully describe the two composing bodies. We present the results for a gyrostat which has the same assumptions about geometric and inertial symmetry that are found in the Whipple bicycle model. Specifically, we assume the frame and fork are inertially symmetric about their respective XZ planes, that the respective wheel centers lie in that plane, and that the wheels are inertially symmetric about an axis of rotation which is perpendicular to that plane.

Consider a carrier A with a set of mutually perpendicular axes X, Y, Z intersecting at the mass center A^* of A . A has mass m_A , and let $\hat{\mathbf{a}}_x, \hat{\mathbf{a}}_y, \hat{\mathbf{a}}_z$ be unit vectors parallel to X, Y, Z , respectively, and express the inertia dyadic of A for A^* as $\mathbf{I}^{A/A^*} = I_{Axx}\hat{\mathbf{a}}_x\hat{\mathbf{a}}_x + I_{Ayy}\hat{\mathbf{a}}_y\hat{\mathbf{a}}_y + I_{Azz}\hat{\mathbf{a}}_z\hat{\mathbf{a}}_z + I_{Axz}(\hat{\mathbf{a}}_x\hat{\mathbf{a}}_z + \hat{\mathbf{a}}_z\hat{\mathbf{a}}_x)$. A is inertially symmetric about the XZ plane. Attached to A with a revolute joint is rotor B with mass m_B . Fixed to B is a set of mutually perpendicular axes X', Y', Z' , intersecting at the mass center B^* of B . Let the revolute joint axis be $Y' \parallel Y$, assume that B^* lies in the XY plane, and express the inertia dyadic of B for B^* as $\mathbf{I}^{B/B^*} = I\hat{\mathbf{a}}_x\hat{\mathbf{a}}_x + J\hat{\mathbf{a}}_y\hat{\mathbf{a}}_y + I\hat{\mathbf{a}}_z\hat{\mathbf{a}}_z$ (i.e., B is inertially symmetric about the Y' axis). Let $\mathbf{r}^{B^*A^*} = a\hat{\mathbf{a}}_x + b\hat{\mathbf{a}}_z$

be the position vector from the mass center of rotor B to the mass center of carrier A . Let G^* denote the center of mass of A and B and refer to the cylindrical gyrostat simply as G . The cylindrical gyrostat has mass

$$m_G \triangleq m_A + m_B \quad (2.1)$$

and the position vector from B^* to G^* is

$$\mathbf{r}^{B^*G^*} = \frac{m_A}{m_A + m_B} (a\hat{\mathbf{a}}_x + b\hat{\mathbf{a}}_z) \quad (2.2)$$

The inertia dyadic of G for G^* is

$$\begin{aligned} \mathbf{I}^{G/G^*} &= \mathbf{I}^{A/G^*} + \mathbf{I}^{B/G^*} \\ &= \mathbf{I}^{A/A^*} + \mathbf{I}^{A^*/G^*} + \mathbf{I}^{B/B^*} + \mathbf{I}^{B^*/G^*} \\ &= \underbrace{\left(I + I_{Axx} + \frac{m_A m_B}{m_A + m_B} b^2 \right)}_{\triangleq I_{Gxx}} \hat{\mathbf{a}}_x \hat{\mathbf{a}}_x \\ &\quad + \underbrace{\left(J + I_{Ayy} + \frac{m_A m_B}{m_A + m_B} (a^2 + b^2) \right)}_{\triangleq I_{Gyy}} \hat{\mathbf{a}}_y \hat{\mathbf{a}}_y \\ &\quad + \underbrace{\left(I + I_{Azz} + \frac{m_A m_B}{m_A + m_B} a^2 \right)}_{\triangleq I_{Gzz}} \hat{\mathbf{a}}_z \hat{\mathbf{a}}_z \\ &\quad + \underbrace{\left(I_{Axz} - \frac{m_A m_B}{m_A + m_B} ab \right)}_{\triangleq I_{Gxz}} (\hat{\mathbf{a}}_x \hat{\mathbf{a}}_z + \hat{\mathbf{a}}_z \hat{\mathbf{a}}_x) \end{aligned} \quad (2.3)$$

where \mathbf{I}^{A^*/G^*} denotes the inertia dyadic relative to G^* of a (fictitious) particle situated at A^* and having mass m_A (similarly for \mathbf{I}^{B^*/G^*}) [2]. Describing the dynamics of a gyrostat in terms of $m_G, I_{Gxx}, I_{Gyy}, I_{Gzz}, I_{Gxz}$, and J (six parameters) is substantially simpler than describing the dynamics in terms of the parameters fundamental to each individual rigid body $m_A, m_B, I, J, I_{Axx}, I_{Ayy}, I_{Azz}, I_{Axz}$, and I_{Axz} (eight parameters). This is partly due to the simple fact that there are two fewer parameters to consider, but also due to significant simplifications that occur when forming generalized inertia forces [4].

2.2.3 Bicycle gyrostat parameters

We assume the bicycle rear frame and rear wheel form a cylindrical gyrostat of the same type as that presented in subsection 2.2.2, as do the fork and front wheel. The bicycle may be

considered to be composed of two such cylindrical gyrostats whose carriers (the frame and fork) are connected by a revolute joint along the steer axis. It is natural to consider how many parameters are needed to describe a bicycle using this formulation. In addition to the six inertial parameters (five inertia scalars and one mass), each gyrostat requires five more parameters: two torus radii, two distances defining the mass center location relative to the wheel center, and the distance from the wheel center to the steer axis along a perpendicular to the steer axis. Thus eleven parameters are needed to describe each cylindrical gyrostat. One more parameter is necessary to define the distance along the steer axis, between the line segments from each wheel center to the steer axis. Thus, a total of twenty three parameters fully characterize the bicycle model. These parameters are tabulated in [Table 2.1](#). Some of the parameters are illustrated in [Figure 2.1](#).

Symbol	Description
$I_{xx}, I_{yy}, I_{zz}, I_{xz}$	gyrostat central inertia scalars
J	wheel spin moment of inertia
m	gyrostat mass
R, r	wheel major and minor radii
a, b	distances from wheel center to gyrostat mass center
c	distance from wheel center steer axis in X direction
l_s	steer axis separation

Table 2.1: Bicycle gyrostat parameters. To distinguish whether they are a property of the rear or the front gyrostat, the first 11 parameters are subscripted with r or f . For example, m_r denotes the rear gyrostat mass, while J_f denotes the front wheel spin moment of inertia. The steer axis separation l_s is a property of how the two gyrostats are connected and hence is not subscripted in this fashion.

Measuring the 23 (21 if the wheels are assumed to be knife edged) parameters for a real bicycle is very similar to the procedure described in [6] except fewer measurements are needed. Instead of determining the mass, mass center locations, and mass distribution of four individual bodies, equivalent properties of only two cylindrical gyrostats need to be measured. Practically speaking, this means only two masses need to be measured (instead of four), the mass center locations and gyrostat inertia measurements should be performed with the rear wheel and frame (front wheel and fork) rigidly connected (i.e., wheel unable to spin), and the wheel moment

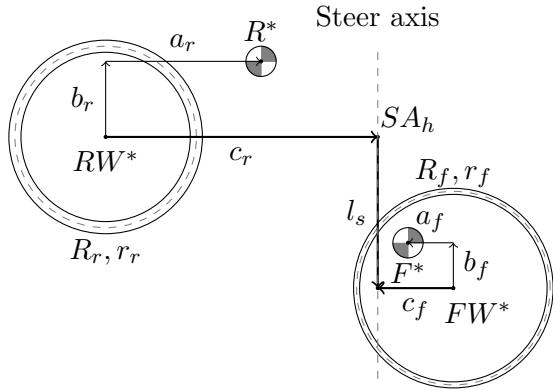


Figure 2.1: Bicycle gyrostat parameters. The X and Z axes of the gyrostat carriers point to the right and down, respectively (not pictured). For a typical bicycle such as the one pictured above, b_r , a_f , b_f , and c_f are all negative. Note that each parameter is defined without reference to the configuration of the bicycle. The rear and front wheel centers are RW^* and FW^* , respectively, and the rear and front mass centers are R^* and F^* , respectively. The head of the steer axis SA_h is the point on top of the steer axis and the foot of the steer axis SA_f (not pictured) is the point on the bottom of the steer axis, both of which are fixed with respect to the gyrostat carriers.

of inertia about an axis in the wheel plane needn't be measured (the wheel spin inertia still does need to be measured, however). The time and energy savings of the experimenter is fairly minor, however, the measurement of the fork inertia can be problematic if the torsional pendulum stiffness is such that the natural frequencies are relatively high; by measuring the inertia of the fork and wheel together, this issue is mitigated to some degree. An alternative method is to measure the exact parameters as described in [6], and convert them to the gyrostat parameters as described in subsection 2.2.4.

2.2.4 Parameter conversion

If the bicycle model in [3] is extended to use toroidal wheels (as opposed to knife edged), and all Meijaard parameters remain otherwise unchanged, a total of 27 parameters describe the bicycle model. The two extra parameters are the rear and front wheel minor radii, which we denote with t_R and t_F , respectively. The mapping from this 27 dimensional parameter space to the 23 dimensional gyrostat parameter space is surjective – distinct choices of Meijaard parameters can yield identical gyrostat parameters (and hence identical dynamics). The practical implication of this is that given a set of gyrostat parameters, it is not generally possible to determine a unique set of the Meijaard parameters. This should not be surprising given that it can be shown with dimensional analysis that the minimal parameter space (assuming $t_R = t_F = 0$) is only 9 dimensional [11] (presumably it would be 11 dimensional if t_R and t_F are included, though this

has not been verified).

In the equations that follow, the symbols used for the parameters presented in [3] are on the right side of the equality, while the symbols used to describe the gyrostat parameters are on the left of the equality. Some parameters have very similar symbols but should not be confused as being the same (i.e., $m_f \neq m_R$). Of the 23 gyrostat parameters, the following 6 have identical counterparts in the Meijaard parameters

$$J_r = I_{Ryy} \quad (2.4)$$

$$R_r = r_R \quad (2.5)$$

$$r_r = t_R \quad (2.6)$$

$$J_f = I_{Fyy} \quad (2.7)$$

$$R_f = r_F \quad (2.8)$$

$$r_f = t_F \quad (2.9)$$

The rear and front gyrostat masses are trivially related

$$m_r = m_B + m_R \quad (2.10)$$

$$m_f = m_H + m_F \quad (2.11)$$

With $s_\lambda = \sin \lambda$, $c_\lambda = \cos \lambda$, the mass center locations are related as

$$a_r = \frac{m_B}{m_B + m_R} (c_\lambda x_B - s_\lambda (r_R + t_R + z_B)) \quad (2.12)$$

$$b_r = \frac{m_B}{m_B + m_R} (c_\lambda (r_R + t_R + z_B) + s_\lambda x_B) \quad (2.13)$$

$$a_f = -\frac{m_H}{m_F + m_H} (c_\lambda (w - x_H) + s_\lambda (r_F + t_F + z_H)) \quad (2.14)$$

$$b_f = \frac{m_H}{m_F + m_H} (c_\lambda (r_F + t_F + z_H) - s_\lambda (w - x_H)) \quad (2.15)$$

The three parameters which describe the perpendicular distance of the wheel centers from the steer axis, and the distance between these perpendicular lines are

$$c_r = c_\lambda (c + w) - s_\lambda (r_R + t_R) \quad (2.16)$$

$$c_f = c_\lambda c - s_\lambda (r_F + t_F) \quad (2.17)$$

$$l_s = -c_\lambda (r_F - r_R + t_F - t_R) + s_\lambda w \quad (2.18)$$

The central inertia scalars of the two gyrostats in terms of the Meijaard parameters are

$$\begin{aligned} I_{rxx} &= I_{Rxx} + c_\lambda^2 I_{Bxx} - 2s_\lambda c_\lambda I_{Bxz} + s_\lambda^2 I_{Bzz} \\ &+ \frac{m_R m_B}{(m_R + m_B)} (s_\lambda x_B + c_\lambda (r_R + t_R + z_B))^2 \end{aligned} \quad (2.19)$$

$$\begin{aligned} I_{ryy} &= I_{Ryy} + I_{Byy} \\ &+ \frac{m_R m_B}{(m_R + m_B)} (x_B^2 + (r_R + t_R + z_B)^2) \end{aligned} \quad (2.20)$$

$$\begin{aligned} I_{rzz} &= I_{Rxx} + s_\lambda^2 I_{Bxx} + 2s_\lambda c_\lambda I_{Bxz} + c_\lambda^2 I_{Bzz} \\ &+ \frac{m_R m_B}{(m_R + m_B)} (c_\lambda x_B - s_\lambda (r_R + t_R + z_B))^2 \end{aligned} \quad (2.21)$$

$$\begin{aligned} I_{rxz} &= (c_\lambda^2 - s_\lambda^2) I_{Bxz} + s_\lambda c_\lambda (I_{Bxx} - I_{Bzz}) \\ &- \frac{m_B m_R}{(m_B + m_R)} (c_\lambda x_B - s_\lambda (r_R + t_R + z_B)) (s_\lambda x_B + c_\lambda (r_R + t_R + z_B)) \end{aligned} \quad (2.22)$$

$$\begin{aligned} I_{fxx} &= I_{Fxx} + c_\lambda^2 I_{Hxx} - 2s_\lambda c_\lambda I_{Hxz} + s_\lambda^2 I_{Hzz} \\ &+ \frac{m_H m_F}{(m_F + m_H)} (-s_\lambda (w - x_H) + c_\lambda (r_F + t_F + z_H))^2 \end{aligned} \quad (2.23)$$

$$\begin{aligned} I_{fyy} &= I_{Fyy} + I_{Hyy} \\ &+ \frac{m_F m_H}{(m_F + m_H)} ((w - x_H)^2 + (r_F + t_F + z_H)^2) \end{aligned} \quad (2.24)$$

$$\begin{aligned} I_{fzz} &= I_{Fxx} + s_\lambda^2 I_{Hxx} + 2s_\lambda c_\lambda I_{Hxz} + c_\lambda^2 I_{Hzz} \\ &+ \frac{m_F m_H}{(m_F + m_H)} (c_\lambda (w - x_H) + s_\lambda (r_F + t_F + z_H))^2 \end{aligned} \quad (2.25)$$

$$\begin{aligned} I_{fxz} &= (c_\lambda^2 - s_\lambda^2) I_{Hxz} + s_\lambda c_\lambda (I_{Hxx} - I_{Hzz}) \\ &+ \frac{m_F m_H}{(m_F + m_H)} (c_\lambda (w - x_H) + s_\lambda (r_F + t_F + z_H)) (-s_\lambda (w - x_H) + c_\lambda (r_F + t_F + z_H)) \end{aligned} \quad (2.26)$$

The numerical values of the benchmark parameter set presented in [3] convert to the gyrostat parameters shown in [Table 2.2](#).

2.3 Kinematics

Were the four bodies of the bicycle model to be disconnected and free to move in an arbitrary manner, the system would have 24 degrees of freedom (4 bodies, 6 degrees of freedom per body). Accounting for the 3 revolute joints (wheel axes and steer axis) reduces the number of degrees of freedom from 24 to 9 (each revolute joint removes 5 degrees of freedom). Requiring the lowest point of the rear wheel and front wheel to touch the horizontal ground plane removes two

	Rear gyrostat	Front gyrostat	Units
I_{xx}	7.684799791449106	0.4335379755311007	kg m^2
I_{yy}	11.99931034482759	0.5746857142857142	kg m^2
I_{zz}	5.315110553378478	0.1481477387546135	kg m^2
I_{xz}	4.262158094617231	0.005332503757935524	kg m^2
J	0.12	0.28	kg m^2
m	87	7	kg
R	0.3	0.35	m
r	0	0	m
a	0.4599058376856177	-0.003411905099535333	m
b	-0.4669419422355365	-0.2114010400161699	m
c	0.9534570696121847	-0.0320714267276193	m
l_s	0.2676445084476887		m

Table 2.2: Benchmark bicycle parameters converted to gyrostat parameters. Parameters shown to 15 or more decimal places are not exact.

further degrees of freedom, resulting in 7 configuration degrees of freedom. A minimal choice of configuration variables is not practical (nor generally possible for all configurations) due to the complexity of the holonomic constraint (it is nonlinear and has multiple roots) [9].

The revolute joints and the requirement that the wheels make point contact with the ground plane are configuration (holonomic) constraints. Without any further restrictions regarding the wheel contact with the ground, the system has 7 velocity degrees of freedom as well. If simulation or stability analysis of such a model is desired, a model for the in ground-plane force acting on each wheel-ground contact point is needed. If it is instead assumed that the wheels roll without slip (said another way, that the wheel contact forces can provide *any* force necessary to prevent slip), 4 more degrees of freedom are removed (2 from each no-slip assumption), yielding a system with 3 velocity degrees of freedom. The no-slip assumptions do not constrain the configuration in any way, so the accessible configuration space remains 7 dimensional, but the accessible velocity space is only 3 dimensional. The remainder of this chapter assumes that the wheels do not slip relative to the ground (and hence the system has 3 velocity degrees of freedom).

A wise choice of generalized coordinates can result in a reduction of the number of config-

uration constraints that need to be solved. A familiar example is the pendulum: using x and y to locate the pendulum mass and deriving the equations of motion in terms of x and y (and their derivatives) requires enforcing the constraint $l - \sqrt{x^2 + y^2} = 0$. This choice is ill advised and best avoided in favor of choosing a single generalized coordinate (i.e., the pendulum angle θ) which implicitly satisfies the length constraint. It is generally in the interest of the analyst to introduce the minimal number of coordinates possible so that needless configuration constraint equations are avoided.

Similarly, a wise choice of generalized speeds can reduce or eliminate the number of velocity constraints that need to be solved. A familiar example of this is the rolling disk: only three generalized speeds are needed to fully describe the angular velocity of the disk *and* the velocity of the disk mass center, therefore no velocity constraint equations need to be solved explicitly to obtain the dynamic equations. If instead one introduces six generalized speeds (e.g., three for the disk angular velocity and three for the velocity of the mass center), three scalar constraint equations must be satisfied and hence only three of the generalized speeds can be chosen independently. Which three to choose as independent in the case of the rolling disk is fairly obvious, but for more complicated systems this is not generally the case and care must be taken to avoid singularities in the constraint equations [10]. Just as in the case for generalized coordinates, it is generally in the interest of the analyst to introduce the minimal number of generalized speeds as possible so that needless velocity constraint equations are avoided.

With this wisdom in mind, we describe the configuration and velocity of the bicycle with 8 generalized coordinates and 6 speeds. This choice results in 1 configuration constraint and 3 velocity constraints. With the exception of frame pitch θ , the other seven coordinates are identical to the coordinates presented in [3].

Let $X_N Y_N Z_N$ be a set of mutually perpendicular axes fixed in a Newtonian frame N , intersecting at the inertial origin N^* , with the positive Z axis pointing down into the ground plane and parallel to the local gravitational field. Let $\hat{n}_x, \hat{n}_y, \hat{n}_z$ be a set of dextral unit vectors aligned with X, Y, Z , respectively. Similarly let the rear gyrostat carrier R and the front gyrostat carrier F be equipped with axes $X_R Y_R Z_R$ and $X_F Y_F Z_F$, respectively, each intersecting at their respective gyrostat mass centers R^* and F^* . Let $Z_R \parallel Z_F$ also be parallel to the steer axis SA . Further, let the wheel spin axis of each carrier be parallel to Y_R and Y_F , respectively. Let $\hat{r}_x, \hat{r}_y, \hat{r}_z$ be a set of dextral unit vectors aligned with X_R, Y_R, Z_R , respectively, and similarly let

$\hat{\mathbf{f}}_x, \hat{\mathbf{f}}_y, \hat{\mathbf{f}}_z$ be a set of dextral unit vectors aligned with X_F, Y_F, Z_F , respectively.

To orient R relative to N , first align $X_R Y_R Z_R$ with $X_N Y_N Z_N$ then apply a sequence of body fixed ZXY rotations: yaw ψ , lean ϕ , and pitch θ . The two intermediate frames in the sequence of rotations are the instantaneous yaw frame Y and the instantaneous lean frame L . To orient F relative to R , first align $X_F Y_F Z_F$ with $X_R Y_R Z_R$ (and note that $Z_R \parallel SA \parallel Z_F$), then apply a right handed rotation to F about SA by steer angle δ . Finally, to orient the wheels relative to their respective carrier, apply simple right handed rotations of the wheel relative to the carrier by angles θ_r and θ_f , respectively. These six rotations completely define the orientation of all frames relative to each other and to the inertial frame N .

The position of the gyrostat mass centers relative to the wheel centers is defined by the choice of parameters a_r, b_r, a_f, b_f (see [Figure 2.1](#)). The position from the rear wheel center RW^* to the rear gyrostat mass center R^* is $\mathbf{r}^{RW^*R^*} = a_r \hat{\mathbf{r}}_x + b_r \hat{\mathbf{r}}_z$; similarly, the position from the front wheel center FW^* to the front gyrostat mass center F^* is $\mathbf{r}^{FW^*F^*} = a_f \hat{\mathbf{f}}_x + b_f \hat{\mathbf{f}}_z$. The position from the rear wheel center to the front wheel center is $\mathbf{r}^{RW^*FW^*} = c_r \hat{\mathbf{r}}_x + l_s \hat{\mathbf{r}}_z - c_f \hat{\mathbf{f}}_x$. To locate the rear wheel ground contact P we introduce generalized coordinates x and y . The position from the inertial origin N^* to the rear wheel ground contact P is $\mathbf{r}^{N^*RW^*} = x \hat{\mathbf{n}}_x + y \hat{\mathbf{n}}_y$. The position from P to the rear wheel center RW^* is $\mathbf{r}^{PRW^*} = -r_r \hat{\mathbf{y}}_z - R_r \hat{\mathbf{l}}_z$ ($\hat{\mathbf{y}}_z$ is the downwards vertical unit vector of the instantaneous yaw frame Y , and $\hat{\mathbf{l}}_z$ is the result of rotating $\hat{\mathbf{y}}_z$ by lean angle ϕ about the instantaneous heading axis of the rear wheel). To define the position from the front wheel center FW^* to the front wheel contact Q we introduce the unit vector

$$\hat{\mathbf{g}}_z = \frac{\hat{\mathbf{y}}_z - (\hat{\mathbf{f}}_y \cdot \hat{\mathbf{y}}_z) \hat{\mathbf{f}}_y}{|\hat{\mathbf{y}}_z - (\hat{\mathbf{f}}_y \cdot \hat{\mathbf{y}}_z) \hat{\mathbf{f}}_y|} \quad (2.27)$$

which is the projection of the downwards vertical vector $\hat{\mathbf{y}}_z = \hat{\mathbf{n}}_z$ onto the front wheel plane ($\hat{\mathbf{f}}_y$ is normal to the front wheel plane). Using this unit vector, the position from the front wheel center FW^* to the front wheel contact Q is $\mathbf{r}^{FW^*Q} = R_f \hat{\mathbf{g}}_z + r_f \hat{\mathbf{y}}_z$. The eight generalized coordinates used to fully orient the four bodies and locate the mass centers, wheel centers, and wheel-ground contact points are tabulated in [Table 2.3](#).

In order to maintain contact with the ground plane the following configuration constraint must be satisfied

$$\mathbf{r}^{PQ} \cdot \hat{\mathbf{y}}_z = 0 \quad (2.28)$$

which is the mathematical statement that the lowest point of the front wheel must lie in the

Coordinate	Description	Units
ψ	Bicycle frame yaw angle	rad
ϕ	Bicycle frame lean angle	rad
θ	Bicycle frame pitch angle	rad
δ	Steer angle	rad
θ_r	Rear wheel angle	rad
θ_f	Front wheel angle	rad
x	Rear wheel contact P X_N measure number	m
y	Rear wheel contact P Y_N measure number	m

Table 2.3: Bicycle generalized coordinates. In the reference configuration ($\phi = \delta = 0$), the pitch angle θ is equal to the steer axis tilt λ of the Meijaard parameter set. This is in contrast to the pitch θ_B defined in [3], which is zero in the reference configuration. The two pitch coordinates are related as $\theta = \theta_B + \lambda$; the other 7 coordinates are identical.

ground plane. The dot product on the left hand side depends on the three coordinates lean ϕ , pitch θ , steer δ , and seven geometric parameters (the wheel radii R_r, r_r, R_f, r_f , and the distances c_r, c_f , and l_s). We introduce a vector $q \in \mathbb{R}^3$ for the three coordinates and a vector $p \in \mathbb{R}^7$ for the parameters involved in this constraint

$$q \triangleq [\phi, \theta, \delta] \quad (2.29)$$

$$p \triangleq [R_r, r_r, R_f, r_f, c_r, c_f, l_s] \quad (2.30)$$

which permit the constraint to be written concisely as

$$f_c(q, p) = 0 \quad (2.31)$$

where $f_c : \mathbb{R}^3 \times \mathbb{R}^7 \mapsto \mathbb{R}$. The three elements of q cannot be varied independently and most often the pitch θ is selected to be a dependent coordinate [9].

The angular velocity of the four bodies and the velocity of the gyrostat mass centers can be defined with only six generalized speeds. These six generalized speeds are defined as follows

$$u_1 \triangleq \dot{\psi} \quad (2.32)$$

$$u_2 \triangleq \dot{\phi} \quad (2.33)$$

$$u_3 \triangleq \dot{\theta} \quad (2.34)$$

$$u_4 \triangleq \dot{\delta} \quad (2.35)$$

$$u_5 \triangleq \dot{\theta}_r \quad (2.36)$$

$$u_6 \triangleq \dot{\theta}_f \quad (2.37)$$

The angular velocity of the four bodies relative to N is fully established by the following relations

$${}^N\boldsymbol{\omega}^R = u_1\hat{\mathbf{y}}_z + u_2\hat{\mathbf{l}}_x + u_3\hat{\mathbf{r}}_y \quad (2.38)$$

$${}^R\boldsymbol{\omega}^F = u_4\hat{\mathbf{f}}_z \quad (2.39)$$

$${}^R\boldsymbol{\omega}^{RW} = u_5\hat{\mathbf{r}}_y \quad (2.40)$$

$${}^F\boldsymbol{\omega}^{FW} = u_6\hat{\mathbf{f}}_y \quad (2.41)$$

The velocities of the gyrostat mass centers R^* and F^* , relative to N , are obtained by assuming the wheel contacts P and Q do not slip relative to the ground plane (i.e., ${}^N\mathbf{v}^P = {}^N\mathbf{v}^Q = \mathbf{0}$) and applying the two point velocity theorem to form the velocity of the wheel centers as follows

$${}^N\mathbf{v}^{RW*} = {}^N\boldsymbol{\omega}^{RW} \times \mathbf{r}^{PRW*} \quad (2.42)$$

$${}^N\mathbf{v}^{R*} = {}^N\mathbf{v}^{RW*} + {}^N\boldsymbol{\omega}^R \times \mathbf{r}^{RW*R*} \quad (2.43)$$

$${}^N\mathbf{v}^{FW*} = {}^N\boldsymbol{\omega}^{FW} \times \mathbf{r}^{QFW*} \quad (2.44)$$

$${}^N\mathbf{v}^{F*} = {}^N\mathbf{v}^{FW*} + {}^N\boldsymbol{\omega}^F \times \mathbf{r}^{FW*F*} \quad (2.45)$$

The velocity constraints are obtained by equating the velocity of the head of the steer axis SA_h (see [Figure 2.1](#)) in two separate but equally valid ways. The velocity of SA_h obtained by working from the rear contact P through the rear wheel RW and bicycle frame R is

$${}^N\mathbf{v}_r^{SA_h} = {}^N\mathbf{v}^{RW*} + {}^N\boldsymbol{\omega}^R \times \mathbf{r}^{RW*SA_h} \quad (2.46)$$

while working from the front contact point Q through the front wheel FW and fork F gives

$${}^N\mathbf{v}_f^{SA_h} = {}^N\mathbf{v}^{FW*} + {}^N\boldsymbol{\omega}^F \times \mathbf{r}^{FW*SA_h} \quad (2.47)$$

where the subscripts r and f denote which set of bodies were used to when obtaining the velocity of SA_h . Since this point can only have one velocity, these two vectors must be identical. The velocity constraints in vector form are then

$${}^N\mathbf{v}_r^{SA_h} - {}^N\mathbf{v}_f^{SA_h} = \mathbf{0} \quad (2.48)$$

Taking the dot product of [Equation 2.48](#) with any three perpendicular vectors (i.e., $\hat{\mathbf{r}}_x, \hat{\mathbf{r}}_y, \hat{\mathbf{r}}_z$), yields three scalar equations which are linear in the 6 generalized speeds $u_1 \dots u_6$. These three scalar equations can be written in matrix form as

$$B(q, p)u = 0 \quad (2.49)$$

where $B : \mathbb{R}^3 \times \mathbb{R}^7 \mapsto \mathbb{R}^{3 \times 6}$, $u \in \mathbb{R}^{6 \times 1}$ and p and q are defined in [Equation 2.29](#) and [Equation 2.30](#), respectively. It is worth noting that the same coordinates and parameters involved in the configuration constraint are also involved in the velocity constraints. By choosing three of the six generalized speeds as independent, it is possible to solve [Equation 2.49](#) for the other three (dependent) speeds. Typically, lean rate $\dot{\phi} = u_2$, steer rate $\dot{\delta} = u_4$, and one of the wheel rates ($\dot{\theta}_r = u_5$ or $\dot{\theta}_f = u_6$) are chosen to be the independent speeds.

The angular acceleration of the four bodies and the acceleration of the two gyrostat mass centers are obtained by taking the time derivative in N of the appropriate angular velocity and velocity expressions described above. We omit the details of these calculations but note that they are generated in symbolic form (i.e., human readable text and L^AT_EX code) as well as output in the form of C++ source code by the symbolic computer algebra system (CAS) SymPy [12] and are freely available on the internet [8].

It is important to note that all of velocity and accelerations described above in general depend on both independent and dependent speeds, and their time derivatives (in the case of acceleration and angular acceleration). This can be dealt with in several ways and is further discussed in [section 2.4](#).

2.4 Dynamics

The contributing active forces and torques acting on the bicycle come from terrestrial gravitational forces acting at the mass centers R^* and F^* , and internal torques applied at the wheel axes and the steer axis. These active forces and torques are

$$G_{R^*} = m_r g \hat{\mathbf{y}}_z \quad (2.50)$$

$$G_{F^*} = m_f g \hat{\mathbf{y}}_z \quad (2.51)$$

$$T^{RW/R} = \tau_{rw} \hat{\mathbf{r}}_y \quad (2.52)$$

$$T^{F/R} = \tau_\delta \hat{\mathbf{r}}_z \quad (2.53)$$

$$T^{FW/F} = \tau_{fw} \hat{\mathbf{r}}_y \quad (2.54)$$

The forces acting at the wheel-ground contact points P and Q can be ignored for the purposes of obtaining the dynamic equations because these points have no velocity in N , and therefore the partial velocity of these points with respect to all generalized speeds will be zero.

A large portion of the complexity of the bicycle dynamic equations is due to the velocity constraints. However, careful accounting of the kinematic quantities using a symbolic CAS makes forming the dynamic equations much less tedious. SymPy [12] is an open source software library, written in the Python programming language, specifically designed to assist in performing symbolic mathematics on a computer. It includes most common symbolic operations such as expanding expressions, substitution, differentiation, as well as useful functionality for generating C/C++/Fortran code and for outputting symbolic expressions in L^AT_EX form. Because it is implemented in the general purpose programming language Python, programs that derive equations of motion with it can utilize the functionality of a feature rich, high level programming language (control flow, file I/O, user interaction, graphical user interfaces, plotting, etc.) without having to transition to a separate software tool. This greatly streamlines the workflow when deriving and analyzing the equations of motion.

SymPy has a submodule called `sympy.physics.mechanics` which provides functions and classes to help form equations of motion using Kane's method. I originally implemented this submodule as a separate project called PyDy [7]. In 2011, Gilbert Gede integrated PyDy into the main SymPy project (using the `sympy.physics.mechanics` name) and substantially extended its functionality; in 2012 it was further extended by Angadh Nanjangud to allow for the use of Lagrange's method for forming equations of motion. Since its inception in 2009, it has been funded by three Google Summer of Code projects, and it has received contributions from several other community members, including nearly every exercise of [2] worked out explicitly.

The kinematics as presented in section 2.3 along with the active forces mentioned above were programmed into a Python script which derives the dynamic equations of motion using Kane's method [2]. The equations of motion were linearized using the techniques described in Chapter 3. The symbolic equations were exported to C++ files which were compiled into a software library [8] to provide efficient and convenient functionality to compute eigenvalues, calculate ground reaction forces, form linearized system dynamics matrices, and perform numerical simulation. This library was used to compute all the system dynamics matrices used in the design of the control system presented in Chapter 4.

Configuration and velocity constraints require the analyst to make several choices in the derivation of the equations of motion. The presence of dependent speeds and their time derivatives in the velocity and acceleration expressions implies that they will be present in the dynamic equations unless they are otherwise eliminated. There are at least three ways this can be handled.

The first way involves making a choice of which speeds to be considered independent (i.e., $u_i = [u_2, u_4, u_5]^T$), symbolically solving [Equation 2.49](#) for the dependent speeds (i.e., $u_d = [u_1, u_3, u_6]^T$), symbolically computing time derivatives of the dependent speeds, and eliminating the dependent speeds and their time derivatives from the expressions needed to compute nonholonomic generalized active and inertia forces (i.e., partial angular velocities, partial velocities, inertia forces, and inertia torques). The resulting symbolic equations are then valid so long as that choice of generalized speeds is valid (i.e. [Equation 2.49](#) can be solved for that choice of dependent generalized speeds). This approach is sufficient for most “normal” choices of bicycle parameters and “normal” operating conditions.

The second way of handling dependent quantities is essentially equivalent, except that the choice of independent speeds is not determined until the matrix $B(q, p)$ ([Equation 2.49](#)) is evaluated numerically (e.g., during a numerical integration of the equations of motion). Once $B(q, p)$ is formed numerically, its singular value decomposition is computed ($B(q, p) = U\Sigma U^*$). The row-wise Euclidean norm of V is computed to yield a column vector of the same size as u , and indices of the dependent speeds are chosen to be the indices of the largest entries of this vector (i.e., if there are o dependent generalized speeds, the indices of the o largest entries of the row-wise Euclidean norm of V are selected as the indices of the dependent speeds). Assuming $B(q, p)$ has full row rank (i.e., none of the constraint equations are redundant), this choice guarantees that constraint singularities will be avoided [[10](#)]. Instead of symbolically forming (and generating code for) nonholonomic generalized inertia forces and nonholonomic generalized active forces specific to a particular choice of independent generalized speeds, one instead outputs code for the holonomic generalized active forces and holonomic generalized inertia forces, and solves [Equation 2.49](#) numerically for the matrix A_{rs} which relates the independent speeds to the dependent speeds

$$B(q, p)u = B_i(q, p)u_i + B_d(q, p)u_d \quad (2.55)$$

$$= 0 \quad (2.56)$$

$$\implies u_d = \underbrace{-B_d(q,p)^{-1} B_i(q,p)}_A u_i \quad (2.57)$$

where u_d, u_i are the dependent and independent speeds, and $B_d(q,p), B_i(q,p)$, are the columns of $B(q,p)$ associated with dependent and independent speeds (A is the matrix representation of A_{rs} in equation 2.13.1 of [2]). The nonholonomic generalized active forces and nonholonomic generalized inertia forces can be computed at the time of simulation (as opposed to the time of symbolic derivation) using Equations 4.4.3 and Equations 4.11.4 of [2], respectively. The primary advantage of this approach is that the generated equations can be utilized for more than a single choice of generalized speeds. This technique is implemented in [8].

A final approach (suggested in conversation by David A. Levinson), is to apply the first approach to generate equations of motion for every possible choice of independent speeds and to select, during numerical integration, the appropriate set of equations to integrate based upon the configuration and parameters (assumed constant) at each time step. In the case of the present system, this implies outputting code for $\frac{6!}{(6-3)!} = 120$ distinct sets of equations. The code storage size and the compilation times for this approach quickly become unwieldy for large systems, but this approach but may be fine for small systems.

The general form of the equations of motion and the associated constraint equations is presented in [Table 3.1 of Chapter 3](#). To follow the notation presented there, for the model of the bicycle presented in this chapter, there are $n = 8$ generalized coordinates, $l = 1$ configuration constraints, $o = 6$ generalized speeds, $m = 3$ velocity constraint equations, and $s = 3$ exogenous inputs (one torque applied at each revolute joint).

References for Chapter 2

- [1] G. Franke, W. Suhr, and F. Riess. “An advanced model of bicycle dynamics”. In: *European Journal of Physics* 11.2 (Mar. 1990), pages 116–121. ISSN: 0143-0807. DOI: [10.1088/0143-0807/11/2/010](https://doi.org/10.1088/0143-0807/11/2/010). URL: <http://stacks.iop.org/0143-0807/11/i=2/a=010?key=crossref.a87799f912a157a8669ab1426468bd7a> (cited on page 7).
- [2] Thomas R. Kane and David A. Levinson. *Dynamics: Theory and Applications*. McGraw-Hill, Inc, 1985. ISBN: 978-0070378469 (cited on pages 8, 19, 21).
- [3] Jaap P. Meijaard, Jeremy M. Papadopoulos, Andy Ruina, and Arend L. Schwab. “Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463.2084 (Aug. 2007), pages 1955–1982. ISSN: 1364-5021. DOI: [10.1098/rspa.2007.1857](https://doi.org/10.1098/rspa.2007.1857). URL: <http://rspa.royalsocietypublishing.org/cgi/doi/10.1098/rspa.2007.1857> (cited on pages 5, 7, 10–12, 14, 16).
- [4] Paul C. Mitiguy and Keith Jerome Reckdahl. “Efficient Dynamical Equations for Gyrostats”. In: *Journal of Guidance, Control, and Dynamics* 24.6 (Nov. 2001), pages 1144–1156. ISSN: 0731-5090. DOI: [10.2514/2.4849](https://doi.org/10.2514/2.4849). URL: <http://arc.aiaa.org/doi/abs/10.2514/2.4849> (cited on pages 4, 7, 8).
- [5] Jason Keith Moore and Mont Hubbard. “Parametric Study of Bicycle Stability”. In: *The Engineering of Sport 7*. Edited by Margaret Estivalet and Pierre Brisson. Biarritz, France: Springer Paris, 2008, pages 311–318. DOI: [10.1007/978-2-287-99056-4_39](https://doi.org/10.1007/978-2-287-99056-4_39). URL: http://link.springer.com/chapter/10.1007/978-2-287-99056-4_39 (cited on page 6).
- [6] Jason Keith Moore, Mont Hubbard, Arend L. Schwab, and Jodi D. G. Kooijman. “Accurate Measurement of Bicycle Parameters”. In: *Proceedings of the 1st Bicycle and Motorcycle Dynamics Conference*. Edited by Arend L. Schwab. October. Delft, The Netherlands, 2010, page 23 (cited on pages 5, 9, 10).
- [7] Dale Lukas Peterson. *PyDy*. URL: <http://pydy.org/> (cited on page 19).
- [8] Dale Lukas Peterson. *libbicycle*. 2013. URL: <https://github.com/hazelnusse/libbicycle> (cited on pages 18, 19, 21).

- [9] Dale Lukas Peterson and Mont Hubbard. “Analysis of the Holonomic Constraint in the Whipple Bicycle Model”. In: *The Engineering of Sport 7*. Whipple 1899. Springer Paris, 2008, pages 623–631. DOI: [10 . 1007 / 978 - 2 - 287 - 09413 - 2 _ 75](https://doi.org/10.1007/978-2-287-09413-2_75). URL: [http : // www . springerlink . com / content / u711012816534258 /](http://www.springerlink.com/content/u711012816534258/) (cited on pages 13, 16).
- [10] Keith Jerome Reckdahl. “Dynamics and control of mechanical systems containing closed kinematic chains”. Ph.D. Stanford University, 1996, page 231 (cited on pages 14, 20).
- [11] Robin S. Sharp. “On the Stability and Control of the Bicycle”. In: *Applied Mechanics Reviews* 61.6 (2008), page 060803. ISSN: 00036900. DOI: [10 . 1115 / 1 . 2983014](https://doi.org/10.1115/1.2983014). URL: [http : // link . aip . org / link / AMREAD / v61 / i6 / p060803 / s1 & Agg = doi](http://link.aip.org/link/AMREAD/v61/i6/p060803/s1&Agg=doi) (cited on pages 5, 10).
- [12] SymPy Development Team. *Sympy: Python library for symbolic mathematics*. 2013. URL: [http : // www . sympy . org](http://www.sympy.org) (cited on pages 18, 19).
- [13] Tae-Oh Tak, Jong-Sung Won, and Gwang-Yeol Baek. “Design Sensitivity Analysis of Bicycle Stability and Experimental Validation”. In: *Proceedings of the 1st Bicycle and Motorcycle Dynamics Conference*. October. Delft, The Netherlands, 2010, pages 20–22 (cited on page 6).
- [14] Jens Wittenburg. *Dynamics of Multibody Systems*. Springer, 2008, page 223. ISBN: 978-3-540-73913-5. URL: [http : // www . springer . com / materials / mechanics / book / 978 - 3 - 540 - 73913 - 5](http://www.springer.com/materials/mechanics/book/978-3-540-73913-5) (cited on pages 4, 7).

Chapter 3

Linearization of equations of motion of constrained multibody systems

3.1 Introduction

Many common dynamic mechanical systems have configuration or velocity constraints. Analyses of such systems often require linearized forms of the motion equations. To address this issue, we developed a procedure for organizing the constraint and motion equations and their subsequent linearization. This procedure was specifically developed for equations of motion generated by Kane's method, but is compatible with any method as long as the following are true: only ordinary differential equations are required to describe the system, and independent and dependent states remain in the final equations (i.e. no DAE's, no substitution of variables). Following a brief review of Kane's method and the structure of the equations it generates, we present the procedure to symbolically linearize the nonlinear motion equations of constrained multibody systems, and illustrate it with the familiar example of the rolling disk.

Design of linear state feedback controllers and first order stability analysis of equilibria are two of the most common uses of multibody system dynamics models. Both of these uses require linearized equations of motion. For unconstrained systems, obtaining the linearized equations of motion is as simple as arranging the equations of motion in first order form ($\dot{x} = f(x, t)$) and evaluating the Jacobian of the right hand side function, with respect to the states, evaluated at the equilibria point of interest. However, when the system in question has constraints, i.e., not all the states are independent, this process is not as straightforward.

For a system with constraints, Kane's dynamical equations [4] are first order ODE's (in the generalized speeds) which include both the independent and dependent states (assuming

the dependents states have not been algebraically eliminated). The kinematical and dynamical differential equations can be solved for the time derivatives of the generalized coordinates and generalized speeds, respectively. Taken together, it is tempting to think that this system of first order differential equations could be linearized in the same fashion as an unconstrained system. Unfortunately, this is not the case. Linearizing these equations in the same manner as one would for an unconstrained system (by directly computing the Jacobian) is incorrect and will yield incorrect linearized equations (and subsequently incorrect eigenvalues, in the case of a stability analysis). We illustrate this by means of a familiar example, the rolling disk, formulated with a non-traditional set of coordinates and generalized speeds which are not all independent. We provide a systematic method to correctly generate the linear equations of motion for constrained systems which have been found using Kane's method (or any other method which results in equations with the same form). We apply this method to the rolling disk example to illustrate that it yields eigenvalues that match previously published results.

We consider three types of constraints: configuration, velocity, and acceleration. Configuration constraints limit the location or orientation of parts of the system, relative to the external world or other parts of the system. Velocity constraints limit the speeds at which the configuration can change, either from configuration constraints (through time differentiation) or independent application. Velocity constraints most often appear in systems where there is rolling without slip or there are closed kinematic loops. In the context of this chapter, acceleration constraints arise either from time differentiated velocity constraints or similar kinematic considerations.

Bottema appears to be the first author to have shown that special considerations are needed for linearizing nonholonomic systems [3]. While other authors have examined linearization of nonholonomic systems, we have found that most techniques are not directly applicable to equations formed using Kane's method, and while we do not doubt that they *could* be applied correctly, the details of how to do so, as far as the authors are aware, have not been explicitly presented. Kang et al. [5] and Negrut and Ortiz [11] have both explored linearization, but these methods have been developed for equations of motion found using Lagrange's method, which contain Lagrange multipliers (which are not present in equations of motion generated from Kane's method). Further, both of these papers lack a discussion of basic concepts such as how many quantities are independent, and in the case of eigenvalue computation, how many

eigenvalues should be present. Minaker and Rieveley [9] and Schwab and Meijaard [17] both have developed techniques for generating equations of motion (and linear forms thereof) using Lagrange's method; the work of Schwab and Meijaard can be applied symbolically while Minaker and Reiveley's approach is numeric in nature. Neither discuss the choice of generalized speeds as in Kane's method: both form the equations of motion in terms of second time derivatives of the coordinates. Further, neither of these works address constraints which cannot be eliminated (i.e., nonlinear configuration constraints), nor do they address what approach should be taken when a single choice of dependent states may not suffice. While Neuman presents a symbolic (as opposed to numeric) formulation for linearization of the Q-matrix formulation of the Lagrangian [13], his work makes no mention of constrained systems and attempts to contact the author regarding the software (algebraic robot modeler – Arm) as well as internet searches for the software have been fruitless.

The goal of this chapter is to definitively establish how the first time derivative of all coordinates and speeds (independent and dependent) depend, to first order, upon a selection of independent coordinates and independent speeds, for an arbitrary point of linearization. We demonstrate how a reduced portion of these linearized equations can be used to perform standard stability analysis (i.e., for control system design or linear stability analysis). The formalism presented here is generic enough to cover most examples of time-varying constrained multibody systems with arbitrary external inputs and arbitrary specified quantities. The procedure was created for systems whose equations of motion have been derived with Kane's method. Although also applicable to dynamic system equations formulated using other methods, it is restricted to systems which can be completely described by a set of ODE's (i.e., DAE's needn't be solved).

We begin with a review of Kane's method for generating equations of motion is reviewed first, in order to properly orient the reader to the format and some of the qualities of the generated equations. Next, we apply Kane's method to generate the equations of motion for the rolling disk, but we purposefully introduce a dependent coordinate and three dependent speeds in the formulation. The need for a different linearization technique is demonstrated with this example. In response to this need, we present the newly developed linearization procedure and its derivation. This new linearization procedure is applied the same rolling disk example to illustrate its validity and an example of its application. Finally, we discuss other details and nuances of the procedure and its use, as well as directions for future work.

3.2 Kane's method, briefly

To familiarize the reader with Kane's method, some concepts relating to its use will be described. The first is the use of generalized speeds in addition to generalized coordinates. The second concept is in how these generalized speeds can be used to project permissible motions of the system, and how this removes the need to consider non-contributing (internal constraint) forces. Then, the mathematical steps required to use Kane's method are shown and described.

When using Lagrange's method, generalized coordinates are used to describe the configuration of a system while the time derivatives of the generalized coordinates describe the velocities of a system. Kane's method allows for the velocities of a system to be written in terms of generalized speeds, which are not required to be the time derivatives of coordinates (although such a definition is permitted).

The benefit of choosing generalized speeds to be other than simply the derivatives of the corresponding generalized coordinates can be seen in the following example. If the orientation of a rigid body were to be described using Euler parameters or quaternions as coordinates, its angular velocity in terms of time derivatives of those coordinates is relatively complicated; it also requires a velocity constraint. Using generalized speeds allows for the angular velocity to be expressed more simply as

$${}^N\boldsymbol{\omega}^B = \omega_1 \hat{\mathbf{b}}_1 + \omega_2 \hat{\mathbf{u}}_2 + \omega_3 \hat{\mathbf{b}}_3 \quad (3.1)$$

or, the angular velocity of rigid body B in the reference frame N is defined as the sum of three generalized-speed basis-vector products. The derivatives of these generalized speeds will appear in the equations of motion, rather than the twice time differentiated generalized coordinates (\ddot{q} 's).

A drawback of allowing such definitions can be seen when formulating the kinematic differential equations, which relate the rate of change of the generalized coordinates to the generalized speeds. These equations become more complicated (in comparison to $\dot{q}_i = u_i$), but this is usually offset by the significantly simpler dynamical equations of motion [10].

Use and understanding of Kane's method requires the use and understanding of partial velocities and partial angular velocities. These are simply the partial derivatives of the (angular) velocity vectors with respect to the generalized speeds. For example, for the angular velocity in

equation (3.1), the partial angular velocities of body B in reference frame N are

$${}^N\boldsymbol{\omega}_{\omega_1}^B = \hat{\mathbf{b}}_1 \quad {}^N\boldsymbol{\omega}_{\omega_2}^B = \hat{\mathbf{b}}_2 \quad {}^N\boldsymbol{\omega}_{\omega_3}^B = \hat{\mathbf{b}}_3$$

and it can be seen that the partial velocities represent the direction of motion associated with each generalized speed. The geometric interpretations of these partial (angular) velocities are discussed at length by Lesser [7].

Taking the dot product of each partial velocity and both sides of Newton's second law, $\mathbf{F} = m\mathbf{a}$ (or the rotational equivalent), only terms that are parallel to the partial velocities remain; one scalar equation is generated for each generalized speed. An important byproduct is that internal constraint forces (non-contributing forces) imposed by objects such as pin or sliding joints no longer need to be considered. The reason for this is that partial velocities are *by construction* perpendicular to forces of constraint. Constraint forces can be included in the formulation, but will not appear in the final equations of motion because of this very desirable property.

In summary, Kane's method allows for velocities to be defined in terms of generalized speeds, allowing for simpler velocity expressions. The dot product of each partial velocity (the direction of motion associated with each generalized speed) with Newton's second law (or Euler's equations) removes all terms which are not related to the rate of change of each generalized speed. This generally results in simple equations of motion, which are easier to form.

The mathematical formalism for applying Kane's method to multibody systems follows. Consider a system composed of particles P_1, \dots, P_h , rigid bodies B_1, \dots, B_g , with points of force application Q_1, \dots, Q_k , and reference frames of torque application E_1, \dots, E_c , all defined relative to the inertial reference frame N . This system has generalized coordinates q_1, \dots, q_n and generalized speeds u_1, \dots, u_o . Additionally, there are l configuration constraints and m velocity constraints.

In order to properly apply Kane's method, the velocities of each component in the system need to be written as linear combinations of the generalized speeds. The translational velocity of the particles are

$${}^N\mathbf{v}^{P_i} = \sum_{j=1}^o {}^N\mathbf{v}_{u_j}^{P_i} u_j + {}^N\mathbf{v}_t^{P_i} \quad i = 1, \dots, g \quad (3.2)$$

while the translational velocity of rigid body mass centers are

$${}^N\mathbf{v}^{B_i^*} = \sum_{j=1}^o {}^N\mathbf{v}_{u_j}^{B_i^*} u_j + {}^N\mathbf{v}_t^{B_i^*} \quad i = 1, \dots, h \quad (3.3)$$

and finally the translational velocity of points which have applied forces are

$${}^N\boldsymbol{v}^{Q_i} = \sum_{j=1}^o {}^N\boldsymbol{v}_{u_j}^{Q_i} u_j + {}^N\boldsymbol{v}_t^{Q_i} \quad i = 1, \dots, k \quad (3.4)$$

Next, the angular velocity of rigid bodies are

$${}^N\boldsymbol{\omega}^{B_i} = \sum_{j=1}^o {}^N\boldsymbol{\omega}_{u_j}^{B_i} u_j + {}^N\boldsymbol{\omega}_t^{B_i} \quad i = 1, \dots, h \quad (3.5)$$

and the angular velocity of reference frames which have applied torques are

$${}^N\boldsymbol{\omega}^{E_i} = \sum_{j=1}^o {}^N\boldsymbol{\omega}_{u_j}^{E_i} u_j + {}^N\boldsymbol{\omega}_t^{E_i} \quad i = 1, \dots, c \quad (3.6)$$

In these equations, each term \boldsymbol{v}_t or $\boldsymbol{\omega}_t$ represents a velocity or angular velocity component which is a prescribed function of time, and has no dependence on generalized speeds (e.g., a motor or crank driven at a specified rate).

For this system, we consider configuration constraints of the form

$$f_c(q, t) = 0 \quad (3.7)$$

where $q \in \mathbb{R}^n$, $t \in \mathbb{R}$, and $f_c : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^l$. The velocity constraints are taken to have the form

$$f_v(q, u, t) = 0 \quad (3.8)$$

where $u \in \mathbb{R}^o$, $f_v : \mathbb{R}^n \times \mathbb{R}^o \times \mathbb{R} \mapsto \mathbb{R}^m$, and f_v is linear in u . It is important that the velocity constraints include the time-differentiated configuration constraints or equivalent constraints which produce the same behavior. Furthermore, velocity constraints should be written such that generalized speeds and coordinates appear, but \dot{q} terms do not (i.e., they should be eliminated by appealing to the kinematic differential equations). Finally, acceleration constraints are typically time-differentiated velocity constraints (or similarly formed from kinematic considerations) and are assumed to have the form

$$f_a(q, \dot{q}, u, \dot{u}, t) = 0 \quad (3.9)$$

where $f_a : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^o \times \mathbb{R}^o \times \mathbb{R} \mapsto \mathbb{R}^m$. In contrast to the velocity constraints, our formulation permits the acceleration constraints to involve \ddot{q} terms, although eliminating them using by appealing to the kinematic differential equations presents no problems. By assumption, the velocity constraints are linear with respect to u , so we introduce

$$B \triangleq \nabla_u f_v(q, u, t) \quad (3.10)$$

which implies that $B : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^m \times \mathbb{R}^o$, i.e., B is an $m \times o$ matrix whose entries depend on q and t . The velocity constraint equations may now be written as

$$Bu + f_{vt}(q, t) = 0 \quad (3.11)$$

where $f_{vt} : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^m$ arises if there are explicit time varying terms in the velocity constraints. These typically arise from specified quantities whose dynamics are being ignored (e.g., a motor spinning at constant speed).

With these definitions established, Kane's method can be applied. In vector form, Kane's holonomic dynamical equations are

$$F + F^* = 0 \quad (3.12)$$

where $F \in \mathbb{R}^o$ are the generalized active forces and $F^* \in \mathbb{R}^o$ are the generalized inertia forces. The i -th element of F and F^* are each associated with the i -th generalized speed. If Newton's second law is written as $f - ma = 0$, F is the multibody generalization of f and F^* is the multibody generalization of $-ma$. As will be shown, these forces are obtained by taking the dot product of the partial velocities and partial angular velocities, as defined in (3.2) - (3.6), with the resultant forces and torques (\mathbf{R} and \mathbf{T}) and with inertial forces/torques (\mathbf{R}^* and \mathbf{T}^*).

The resultant force at a point is defined as the sum of all force vectors acting directly on that point. The resultant torque on a reference frame is defined similarly – it is the sum of all torque vectors acting on that reference frame. Distance forces do not need to be transformed into a resultant force/torque pair for any components of the system; they only need to be considered at the point of application (e.g., if a force \mathbf{f} is the only force applied to body B at point Q , then $\mathbf{R}_{B^*} = \mathbf{0}$ and $\mathbf{T}_B = \mathbf{0}$ but $\mathbf{R}_Q = \mathbf{f}$). The generalized active forces are

$$F = \begin{bmatrix} \sum_{i=1}^g \mathbf{R}_{B_i^*} \cdot {}^N \mathbf{v}_1^{B_i^*} + \sum_{i=1}^h \mathbf{R}_{P_i} \cdot {}^N \mathbf{v}_1^{P_i} + \sum_{i=1}^k \mathbf{R}_{Q_i} \cdot {}^N \mathbf{v}_1^{Q_i} + \sum_{i=1}^g \mathbf{T}_{B_i} \cdot {}^N \boldsymbol{\omega}_1^{B_i} + \sum_{i=1}^c \mathbf{T}_{E_i} \cdot {}^N \boldsymbol{\omega}_1^{E_i} \\ \vdots \\ \sum_{i=1}^g \mathbf{R}_{B_i^*} \cdot {}^N \mathbf{v}_2^{B_i^*} + \sum_{i=1}^h \mathbf{R}_{P_i} \cdot {}^N \mathbf{v}_2^{P_i} + \sum_{i=1}^k \mathbf{R}_{Q_i} \cdot {}^N \mathbf{v}_2^{Q_i} + \sum_{i=1}^g \mathbf{T}_{B_i} \cdot {}^N \boldsymbol{\omega}_2^{B_i} + \sum_{i=1}^c \mathbf{T}_{E_i} \cdot {}^N \boldsymbol{\omega}_2^{E_i} \\ \vdots \\ \sum_{i=1}^g \mathbf{R}_{B_i^*} \cdot {}^N \mathbf{v}_o^{B_i^*} + \sum_{i=1}^h \mathbf{R}_{P_i} \cdot {}^N \mathbf{v}_o^{P_i} + \sum_{i=1}^k \mathbf{R}_{Q_i} \cdot {}^N \mathbf{v}_o^{Q_i} + \sum_{i=1}^g \mathbf{T}_{B_i} \cdot {}^N \boldsymbol{\omega}_o^{B_i} + \sum_{i=1}^c \mathbf{T}_{E_i} \cdot {}^N \boldsymbol{\omega}_o^{E_i} \end{bmatrix} \quad (3.13)$$

We now define the inertia forces and torques, \mathbf{R}^* and \mathbf{T}^* . We leave out the inertial frame N from the expressions; it is assumed to be N . The inertia force for particle P is

$$\mathbf{R}_P^* \triangleq -m_P {}^N \mathbf{a}^P \quad (3.14)$$

while the inertia force and inertia torque for rigid body B is

$$\mathbf{R}_{B^*}^* \triangleq -m_B {}^N \mathbf{a}^{B^*} \quad (3.15)$$

$$\mathbf{T}_B^* \triangleq -{}^N \boldsymbol{\alpha}^B \cdot \mathbf{I}^{B/B^*} - {}^N \boldsymbol{\omega}^B \times \mathbf{I}^{B/B^*} \cdot {}^N \boldsymbol{\omega}^B \quad (3.16)$$

Where \mathbf{I}^{B/B^*} is the central inertia dyadic of the rigid body B about its mass center B^* [4]. The generalized inertia forces are

$$\mathbf{F}^* = \begin{bmatrix} \sum_{i=1}^g \mathbf{R}_{B_i^*}^* \cdot {}^N \mathbf{v}_1^{B_i^*} + \sum_{i=1}^h \mathbf{R}_{P_i}^* \cdot {}^N \mathbf{v}_1^{P_i} + \sum_{i=1}^g \mathbf{T}_{B_i}^* \cdot {}^N \boldsymbol{\omega}_1^{B_i} \\ \vdots \\ \sum_{i=1}^g \mathbf{R}_{B_i^*}^* \cdot {}^N \mathbf{v}_2^{B_i^*} + \sum_{i=1}^h \mathbf{R}_{P_i}^* \cdot {}^N \mathbf{v}_2^{P_i} + \sum_{i=1}^g \mathbf{T}_{B_i}^* \cdot {}^N \boldsymbol{\omega}_2^{B_i} \\ \vdots \\ \sum_{i=1}^g \mathbf{R}_{B_i^*}^* \cdot {}^N \mathbf{v}_o^{B_i^*} + \sum_{i=1}^h \mathbf{R}_{P_i}^* \cdot {}^N \mathbf{v}_o^{P_i} + \sum_{i=1}^g \mathbf{T}_{B_i}^* \cdot {}^N \boldsymbol{\omega}_o^{B_i} \end{bmatrix} \quad (3.17)$$

All the terms in (3.12) are now defined, but this equation is valid only if there are no velocity constraints. If there are velocity constraints, the equations must be transformed. As we present them, Kane's nonholonomic dynamical equations will be of dimension $o - m$, where o is the total number of generalized speeds and m is the number of velocity constraints. However, these $o - m$ equations generally involve all o speeds (we assume the dependent speeds and their time derivatives have *not* been algebraically eliminated). In order to obtain the nonholonomic dynamical equations, however, a choice must be made as to which speeds will be dependent. The methods for doing this are described extensively in [16] and described briefly in section 2.4.

We assume the o generalized speeds are ordered $u = [u_i, u_d]^T$, where $u_i \in \mathbb{R}^{o-m}$, $u_d \in \mathbb{R}^m$ are the independent and dependent speeds, respectively. The velocity constraint Jacobian matrix B can be similarly rearranged as $B = [B_i, B_d]$, where $B_i \in \mathbb{R}^{m \times o-m}$ and $B_d \in \mathbb{R}^{m \times m}$. With this in mind, equation (3.11) can be written as

$$B_i u_i + B_d u_d + f_{vt}(q, t) = 0 \quad (3.18)$$

$$\implies u_d = -B_d^{-1} B_i u_i - f_{vt}(q, t) \quad (3.19)$$

For convenience, we define

$$A \triangleq -B_d^{-1} B_i \quad (3.20)$$

and note that $A \in \mathbb{R}^{m \times (o-m)}$. To constrain the generalized active and inertia forces, we partitioned them in a similar manner

$$F = \begin{bmatrix} F_i \\ F_d \end{bmatrix} \quad F^* = \begin{bmatrix} F_i^* \\ F_d^* \end{bmatrix} \quad (3.21)$$

where $F_i, F_i^* \in \mathbb{R}^{o-m}$, $F_d, F_d^* \in \mathbb{R}^m$, allowing the nonholonomic generalized forces to be written as

$$\tilde{F} = F_i + A^T F_d \quad \tilde{F}^* = F_i^* + A^T F_d^* \quad (3.22)$$

where $\tilde{F}, \tilde{F}^* \in \mathbb{R}^{o-m}$ are the nonholonomic generalized active and nonholonomic generalized inertia forces, respectively. This allows Kane's nonholonomic dynamical equations to be written as

$$\tilde{F} + \tilde{F}^* = 0 \quad (3.23)$$

This process of constraining the equations of motion reduces the number of equations from o to $o-m$. However, the (3.23) still contain o unknowns ($\dot{u}_1, \dots, \dot{u}_o$). By augmenting (3.23) with the acceleration constraint equations (3.9), we obtain o equations in o unknowns. Note that equations (3.23) and (3.9) are linear in the \dot{u}_i 's. The augmented system of equations is

$$\begin{bmatrix} \tilde{F} + \tilde{F}^* \\ f_a(q, \dot{q}, u, \dot{u}, t) \end{bmatrix} = 0 \quad (3.24)$$

For convenience, we summarize the critical equations in Table 3.1.

The only new terms are f_0 , f_1 , f_2 , and f_3 . These terms are introduced for notational convenience when deriving the linearization procedure. The n kinematic differential equations are arranged so all terms appear to the left of the equality and arranged into $f_0 : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$ and $f_0 : \mathbb{R}^n \times \mathbb{R}^o \times \mathbb{R} \mapsto \mathbb{R}^n$, depending on whether each term has a u or \dot{q} present. The same organizational scheme is applied to the $o-m$ dynamic differential equations, where $f_2 : \mathbb{R}^n \times \mathbb{R}^o \times \mathbb{R} \mapsto \mathbb{R}^{o-m}$ and $f_3 : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^o \times \mathbb{R}^s \times \mathbb{R} \mapsto \mathbb{R}^{o-m}$; if a term contains a \dot{u} it is placed into f_2 and if not it is placed into f_3 . The motivation for this organizational scheme will be made apparent in Section 3.4.

Quantity	Space	Description
q, \dot{q}	\mathbb{R}^n	Coordinates and their time derivatives
u, \dot{u}	\mathbb{R}^o	Speeds and their time derivatives
r	\mathbb{R}^s	Exogenous inputs
Equation	Description	
$f_c(q, t) = 0$	Configuration constraints	
$f_v(q, u, t) = 0$	Velocity constraints	
$f_a(q, \dot{q}, u, \dot{u}, t) = 0$	Acceleration constraints	
$f_0(q, \dot{q}, t) + f_1(q, u, t) = 0$	Kinematic differential equations	
$f_2(q, \dot{u}, t) + f_3(q, \dot{q}, u, r, t) = 0$	Dynamic differential equations	

Table 3.1: Constrained multibody system governing definitions and equations

3.3 Rolling disk

To illustrate Kane’s method in the presence of configuration and velocity constraints, we consider a thin disk C rolling without slip on a horizontal plane. The dextral unit vectors $\hat{c}_x, \hat{c}_y, \hat{c}_z$ are fixed to C with \hat{c}_y perpendicular to the disk plane. Let r and m be the disk radius and mass, and assume a thin disk mass distribution so that $I_{xx} = I_{zz} = mr^2/4$ and $I_{yy} = mr^2/2$. Take the inertial (Newtonian) frame N to be similarly equipped with dextral unit vectors $\hat{n}_x, \hat{n}_y, \hat{n}_z$, with \hat{n}_z perpendicular to the ground plane (downwards) and aligned with the local gravitational field. To orient the disk, first align C with N , then apply successive body fixed ZXY rotations: yaw q_1 , roll q_2 , and spin q_3 . Denote by A and B the two frames associated with the first two intermediate rotations in the sequence; we refer to these as the instantaneous yaw and lean frames, respectively. The unit vector, \hat{b}_z fixed in the B frame, is parallel to the disk plane and aligned with the position vector from the disk center C^* to the lowest point of the disk P^* .

Traditionally, to locate the center of the disk C^* relative to the inertial origin N^* , two coordinates are used to locate the contact point in the ground plane; these two coordinates and the lean angle and yaw angles determine the location of the center of the disk. The advantage of this approach is that the choice of coordinates guarantees that the disk remain in contact with the ground plane.

To illustrate how dependent coordinates must be accounted for when linearizing the equations of motion, we purposefully locate the center of the disk C^* relative to the inertial origin N^* with a non-minimal choice of coordinates

$$\mathbf{r}^{C^*/N^*} = q_4 \hat{\mathbf{n}}_x + q_5 \hat{\mathbf{n}}_y + q_6 \hat{\mathbf{n}}_z n$$

which results in the configuration constraint \mathbf{f}_c

$$r \cos q_2 + q_6 = 0 \quad (3.25)$$

which must be satisfied for the disk to remain in contact with the ground. While this constraint is easily avoided by appropriate choice of coordinates, it serves to illustrate considerations which must be made when analyzing systems in which there is no obvious way to eliminate configuration constraints.

Again to demonstrate how velocity constraints must be accounted for when obtaining linearized equations of motion, we define the velocity of C^* in N with a non-minimal choice of speeds, defined in the body-fixed (C) frame. The angular velocity of C in N , however, is defined in the intermediate lean frame B .

$${}^N \mathbf{v}^{C^*} = u_4 \hat{\mathbf{c}}_x + u_5 \hat{\mathbf{c}}_y + u_6 \hat{\mathbf{c}}_z \quad (3.26)$$

$${}^N \boldsymbol{\omega}^C = u_1 \hat{\mathbf{b}}_x + u_2 \hat{\mathbf{b}}_y + u_3 \hat{\mathbf{b}}_z \quad (3.27)$$

The unconstrained partial angular velocity and partial velocity of C^* , respectively, are

$${}^N \mathbf{v}_u^{C^*} = [\mathbf{0}, \mathbf{0}, \mathbf{0}, \hat{\mathbf{c}}_x, \hat{\mathbf{c}}_y, \hat{\mathbf{c}}_z] \quad (3.28)$$

$${}^N \boldsymbol{\omega}_u^C = [\hat{\mathbf{b}}_x, \hat{\mathbf{b}}_y, \hat{\mathbf{b}}_z, \mathbf{0}, \mathbf{0}, \mathbf{0}] \quad (3.29)$$

where we present the partial velocities as a 1×6 matrix whose entries are populated with standard 3-vectors. Given these definitions, the velocity of the lowest point of the disk P^* is

$$\begin{aligned} {}^N \mathbf{v}^{P^*} &= \mathbf{v}^{C^*} + \boldsymbol{\omega}^C \times \mathbf{r}^{P^*/C^*} \\ &= u_4 \hat{\mathbf{c}}_x + u_5 \hat{\mathbf{c}}_y + u_6 \hat{\mathbf{c}}_z + r u_2 \hat{\mathbf{b}}_x - r u_1 \hat{\mathbf{b}}_y \\ &= (r u_2 \cos(q_3) + u_4) \hat{\mathbf{c}}_x + (-r u_1 + u_5) \hat{\mathbf{c}}_y + (r u_2 \sin(q_3) + u_6) \hat{\mathbf{c}}_z \end{aligned}$$

where we have made use of the fact that $\mathbf{r}^{P^*/C^*} = r \mathbf{b}_z$. Under the assumption of pure rolling

this immediately yields three velocity constraint equations which make up $f_v(q, u, t) = 0$

$$\begin{bmatrix} ru_2 \cos(q_3) + u_4 \\ -ru_1 + u_5 \\ ru_2 \sin(q_3) + u_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.30a)$$

These equations are linear in the u_i terms, nonlinear in the q_i terms, involve geometric system parameters (but not mass or inertial parameters), and do not explicitly involve \dot{q}_i terms. This structure will be taken advantage of in Section 3.4.

At this point, the reader familiar with the rolling disk problem is certainly wondering why we have chosen such a diabolical set of coordinates and speeds. In this simple example, there is an ***obvious and minimal*** choice of coordinates and speeds (q_6, u_4, u_5, u_6 , should be dependent; this choice will be valid for all parameters and configurations; further, the derivation of the equations of motion can be done without ever introducing dependent quantities), for some systems this is not the case (e.g., Stewart's platform, bicycle). The numeric values of system parameters and the configurations in which the system will operate will determine which coordinates and speeds can be taken to be dependent (and hence potentially eliminated by algebraic manipulations). However, it may not be possible *a priori* to determine easily which coordinates and speeds should be taken as dependent; further, some systems may operate in distinct enough regions of the configuration space that one *must* use different choices of dependent coordinates and/or speeds as the system moves from one region to another. The purpose of our complex choice of coordinates and speeds is to illustrate, in the context of a familiar example, how constraints must be taken into account when linearizing equations of motion. It is our hope that by doing this for a well known and relatively simple example, readers can apply the same techniques to more complicated systems where its use is more appropriate or necessary.

Time differentiating the velocity constraint equations yields the acceleration constraint equations, $f_a(q, \dot{q}, u, \dot{u}, t) = 0$

$$\begin{bmatrix} -ru_2 \sin(q_3)\dot{q}_3 + r \cos(q_3)\dot{u}_2 + \dot{u}_4 \\ -r\dot{u}_1 + \dot{u}_5 \\ ru_2 \cos(q_3)\dot{q}_3 + r \sin(q_3)\dot{u}_2 + \dot{u}_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.31a)$$

which must also be satisfied during general motions. There are two types of terms in these equations; 1) terms involving \dot{u}_i 's and q_i 's, and 2) terms involving \dot{q}_i 's, u_i 's, and q_i 's. This structure will be taken advantage of subsequently.

The kinematic differential equations relate the time derivatives of the coordinates and the generalized speeds. They are obtained by equating velocities expressed in terms of time-differentiated generalized coordinates with velocities expressed in terms of generalized speeds. From the addition theorem for angular velocity, the angular velocity of C is

$${}^N\boldsymbol{\omega}^C = \dot{q}_1 \hat{\mathbf{n}}_z + \dot{q}_2 \hat{\mathbf{a}}_x + \dot{q}_3 \hat{\mathbf{b}}_y \quad (3.32)$$

while time differentiating r^{C^*} in N yields

$$\mathbf{v}^{C^*} = \dot{q}_4 \hat{\mathbf{n}}_x + \dot{q}_5 \hat{\mathbf{n}}_y + \dot{q}_6 \hat{\mathbf{n}}_z \quad (3.33)$$

Equating (3.27) with (3.32) and (3.26) with (3.33), and resolving these vector equations into the B frame, and rearranging so that all terms appear to the left of the equality sign, we obtain

$$\underbrace{\begin{bmatrix} \dot{q}_2 \\ \sin(q_2)\dot{q}_1 + \dot{q}_3 \\ \cos(q_2)\dot{q}_1 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}}_{f_0} + \underbrace{\begin{bmatrix} -u_1 \\ -u_2 \\ -u_3 \\ ru_1 \sin(q_1) \cos(q_2) + ru_2 \cos(q_1) \\ -ru_1 \cos(q_1) \cos(q_2) + ru_2 \sin(q_1) \\ -ru_1 \sin(q_2) \end{bmatrix}}_{f_1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.34)$$

The translational acceleration of C^* and the angular acceleration of C , relative to N , are

$$\begin{aligned} {}^N\mathbf{a}^{C^*} = & -(u_1 \sin(q_3) + u_3 \cos(q_3))u_5 + u_2u_6 + \dot{u}_4)\hat{\mathbf{c}}_x \\ & + ((u_1 \sin(q_3) + u_3 \cos(q_3))u_4 - (u_1 \cos(q_3) - u_3 \sin(q_3))u_6 + \dot{u}_5)\hat{\mathbf{c}}_y \\ & + ((u_1 \cos(q_3) - u_3 \sin(q_3))u_5 - u_2u_4 + \dot{u}_6)\hat{\mathbf{c}}_z \end{aligned} \quad (3.35)$$

$${}^N\boldsymbol{\alpha}^C = -u_2u_3 + u_3^2 \tan(q_2) + \dot{u}_1)\hat{\mathbf{b}}_x + \dot{u}_2 \hat{\mathbf{b}}_y + (u_1u_2 - u_1u_3 \tan(q_2) + \dot{u}_3)\hat{\mathbf{b}}_z \quad (3.36)$$

The mass of the disk is m , and the inertia dyadic of the disk can is

$$\mathbf{I}^{C/C^*} = \frac{mr^2}{4}\hat{\mathbf{c}}_x\hat{\mathbf{c}}_x + \frac{mr^2}{2}\hat{\mathbf{c}}_y\hat{\mathbf{c}}_y + \frac{mr^2}{4}\hat{\mathbf{c}}_z\hat{\mathbf{c}}_z \quad (3.37)$$

Using equations (3.15) and (3.16), $\mathbf{R}_{C^*}^*$ and \mathbf{T}_C^* can be written.

$$\begin{aligned}\mathbf{R}_{C^*}^* = & -m(-(u_1 \sin(q_3) + u_3 \cos(q_3))u_5 + u_2u_6 + \dot{u}_4)\hat{\mathbf{c}}_x \\ & -m((u_1 \sin(q_3) + u_3 \cos(q_3))u_4 - (u_1 \cos(q_3) - u_3 \sin(q_3))u_6 + \dot{u}_5)\hat{\mathbf{c}}_y \\ & -m((u_1 \cos(q_3) - u_3 \sin(q_3))u_5 - u_2u_4 + \dot{u}_6)\hat{\mathbf{c}}_z\end{aligned}\quad (3.38)$$

$$\begin{aligned}\mathbf{T}_C^* = & \frac{mr^2}{4}(2u_1u_2 \sin(q_3) - u_1u_3 \sin(q_3) \tan(q_2) + 2u_2u_3 \cos(q_3) \\ & - u_3^2 \cos(q_3) \tan(q_2) + \sin(q_3)\dot{u}_3 - \cos(q_3)\dot{u}_1)\hat{\mathbf{c}}_x \\ & - \frac{mr^2}{2}\dot{u}_2\hat{\mathbf{c}}_y \\ & + \frac{mr^2}{4}(-2u_1u_2 \cos(q_3) + u_1u_3 \cos(q_3) \tan(q_2) + 2u_2u_3 \sin(q_3) \\ & - u_3^2 \sin(q_3) \tan(q_2) - \sin(q_3)\dot{u}_1 - \cos(q_3)\dot{u}_3)\hat{\mathbf{c}}_z\end{aligned}\quad (3.39)$$

The active forces and active torques are

$$\mathbf{R}_{C^*} = mg\hat{\mathbf{n}}_z \quad (3.40)$$

$$\mathbf{T}_C = \mathbf{0} \quad (3.41)$$

Equations (3.13), (3.17), and (3.12) can be used to construct the equations of motion. For brevity, we omit the results of each step involved. The resulting three dynamic equations are

$$\begin{aligned}-\frac{mr}{4}(r\dot{u}_1 + 4\dot{u}_5) = & -mr(g \sin(q_2) - u_1u_4 \sin(q_3) + \\ & u_1u_6 \cos(q_3) + (\frac{ru_2}{2} - \frac{ru_3}{4} \tan(q_2) - \\ & u_4 \cos(q_3) - u_6 \sin(q_3))u_3)\end{aligned}\quad (3.42)$$

$$\begin{aligned}\frac{mr}{2}(-r\dot{u}_2 + 2 \sin(q_3)\dot{u}_6 + 2 \cos(q_3)\dot{u}_4) = & mr(u_2u_4 \sin(q_3) - u_2u_6 \cos(q_3) + \\ & u_3u_5)\end{aligned}\quad (3.43)$$

$$-\frac{mr^2}{4}\dot{u}_3 = \frac{mr^2}{4}(2u_2 - u_3 \tan(q_2))u_1 \quad (3.44)$$

The terms on the left of the equality are f_2 , the terms on the right of the equality are $-f_3$ (they are written as $f_2 = -f_3$ here purely for formatting reasons).

The twelve unknown quantities appearing in the nine kinematic and dynamic differential equations are \dot{q}_{1-6} and \dot{u}_{1-6} . The three acceleration constraint equations provide the final three equations which allow for all twelve unknowns to be solved. Combining (3.34), (3.42-3.44), and

(3.31), the complete second order differential equations can be written.

$$\begin{bmatrix} \mathbf{f}_0 + \mathbf{f}_1 \\ \mathbf{f}_2 + \mathbf{f}_3 \\ \vdots \\ \mathbf{f}_a \end{bmatrix} = \mathbf{0} \quad (3.45)$$

The naive approach to linearizing these equations of motion would be to solve for the (\dot{q}_i, \dot{u}_i) terms to construct the right-hand side of

$$\begin{bmatrix} \dot{q} \\ \dot{u} \end{bmatrix} = \mathbf{f}(q, u, t) \quad (3.46)$$

and take the Jacobian of \mathbf{f} with respect to $[q \ u]^T$. However, this will give incorrect results. As discussed in [17], when m, r , and g are taken to be unity, and the system is linearized about the upright steady rolling condition, the critical speed is $v \triangleq -r\dot{q}_3 = \pm \frac{1}{\sqrt{3}}$. When following the naive approach, and evaluating the Jacobian at the same operating conditions and parameters, eight of the twelve eigenvalues are identically zero while the remaining four are

$$\lambda_{1,2} = \pm \frac{\sqrt{6}}{3} \sqrt{-v^2}, \quad \lambda_{3,4} = \pm \sqrt{\frac{4}{5} - \frac{12}{5}v^2} \quad (3.47)$$

which demonstrates the incorrectness of this naive approach (since $v = \pm \frac{1}{\sqrt{3}}$ is not a critical point of these eigenvalues). Further, the fact that twelve eigenvalues are obtained should be an alert that something is incorrect since the number of independent quantities is only eight (five coordinates and three speeds). We now outline a linearization procedure which addresses this issue; we revisit this example in Section 3.5.

3.4 Derivation of linearization procedure

In response to the need we have demonstrated, we present a linearization procedure that properly accounts for system constraints. Taking the Jacobian of the right hand side of the ODE's as we did at the end of the previous section is incorrect because it fails to apply the chain rule and thereby properly account for the relationships imposed by configuration, velocity, and acceleration constraints. That it isn't immediately obvious that the chain rule needs to be applied is a byproduct of the commonly used notation which doesn't make it explicitly clear that dependent coordinates and dependent speeds are not only functions of time, but also functions of the independent coordinates and independent speeds. These dependent terms should in fact

be written as

$$q_d(t) \rightarrow q_d(q_i, t) \quad (3.48)$$

$$u_d(t) \rightarrow u_d(q_i, u_i, t) \quad (3.49)$$

While this might seem obvious, no previous author appears to make this explicitly clear when presenting techniques for symbolically linearizing equations of motion which are subject to constraints. While the concept is simple in principle, correctly accounting for all quantities is tedious and error prone. Having a high level, systematic procedure that can be implemented reliably in software is therefore a strong argument for the detailed and explicit procedure we present.

We begin with a first order Taylor series expansion of the equations in Table 3.1 about $q = q^*$, $\dot{q} = \dot{q}^*$, $u = u^*$, $\dot{u} = \dot{u}^*$, $r = r^*$; it is assumed that all of the equations in Table 3.1 are satisfied by these quantities (i.e., the system is satisfies the constraint equations and Newton's 2nd law). In the interest of brevity, we omit writing this point of linearization in each gradient in the calculations below; all are evaluated at this point. Expansion of the three constraint equations, keeping only first order terms, yields

$$f_c(q, t) \approx \underbrace{f_c(q^*, t)}_0 + \nabla_q f_c \delta q \quad (3.50)$$

$$f_v(q, u, t) \approx \underbrace{f_v(q^*, u^*, t)}_0 + \nabla_q f_v \delta q + \nabla_u f_v \delta u \quad (3.51)$$

$$\begin{aligned} f_a(q, \dot{q}, u, \dot{u}, t) &\approx \underbrace{f_a(q^*, \dot{q}^*, u^*, \dot{u}^*, t)}_0 + \nabla_q f_a \delta q + \nabla_{\dot{q}} f_a \delta \dot{q} \\ &\quad + \nabla_u f_a \delta u + \nabla_{\dot{u}} f_a \delta \dot{u} \end{aligned} \quad (3.52)$$

The first terms are identically zero because of the assumption that the point of linearization satisfies the constraints. The Taylor series expansion of the kinematic differential equations is

$$f_0(q, \dot{q}, t) \approx f_0(q^*, \dot{q}^*, t) + \nabla_q f_0 \delta q + \nabla_{\dot{q}} f_0 \delta \dot{q} \quad (3.53)$$

$$f_1(q, u, t) \approx f_1(q^*, u^*, t) + \nabla_q f_1 \delta q + \nabla_u f_1 \delta u \quad (3.54)$$

Summing (3.53) and (3.54) and recognizing that the sum of the first term on the right hand side of each equation must equal zero, we obtain

$$f_0(q, \dot{q}, t) + f_1(q, u, t) \approx \nabla_q(f_0 + f_1) \delta q + \nabla_{\dot{q}} f_0 \delta \dot{q} + \nabla_u f_1 \delta u \quad (3.55)$$

Similarly, a Taylor series expansion of the dynamic differential equations, we obtain

$$f_2(q, \dot{u}, t) \approx f_2(q^*, \dot{u}^*, t) + \nabla_q f_2 \delta q + \nabla_{\dot{u}} f_2 \delta \dot{u} \quad (3.56)$$

$$\begin{aligned} f_3(q, \dot{q}, u, r, t) &\approx f_3(q^*, \dot{q}^*, u^*, r^*, t) + \nabla_q f_3 \delta q \\ &+ \nabla_{\dot{q}} f_3 \delta \dot{q} + \nabla_u f_3 \delta u + \nabla_r f_3 \delta r \end{aligned} \quad (3.57)$$

Summing (3.56) and (3.57) and recognizing that the sum of the first term on the right hand sides of these equations must equal zero, we obtain

$$\begin{aligned} f_2(q, \dot{u}, t) + f_3(q, \dot{q}, u, r, t) &\approx \nabla_q(f_2 + f_3) \delta q + \nabla_{\dot{q}} f_3 \delta \dot{q} \\ &+ \nabla_u f_3 \delta u + \nabla_{\dot{u}} f_2 \delta \dot{u} + \nabla_r f_3 \delta r \end{aligned} \quad (3.58)$$

Equating the right hand sides of equations (3.55), (3.52), and (3.58) to zero (as in Table 3.1), and introducing the following definitions

$$\begin{aligned} \tilde{M}_{qq} &\triangleq \nabla_{\dot{q}} f_0 & \tilde{M}_{uqc} &\triangleq \nabla_{\dot{q}} f_a \\ \tilde{M}_{uuc} &\triangleq \nabla_{\dot{u}} f_a & \tilde{M}_{uqd} &\triangleq \nabla_{\dot{q}} f_3 \\ \tilde{M}_{uud} &\triangleq \nabla_{\dot{u}} f_2 & \tilde{A}_{qq} &\triangleq -\nabla_q(f_0 + f_1) \\ \tilde{A}_{qu} &\triangleq -\nabla_u f_1 & \tilde{A}_{uqc} &\triangleq -\nabla_q f_a \\ \tilde{A}_{uuc} &\triangleq -\nabla_u f_a & \tilde{A}_{uqd} &\triangleq -\nabla_q(f_2 + f_3) \\ \tilde{A}_{uud} &\triangleq -\nabla_u f_3 & \tilde{B}_u &\triangleq -\nabla_r f_3 \end{aligned} \quad (3.59)$$

enables the unconstrained linear state space equations to be written as

$$\begin{bmatrix} \tilde{M}_{qq} & 0_{n \times o} \\ \tilde{M}_{uqc} & \tilde{M}_{uuc} \\ \tilde{M}_{uqd} & \tilde{M}_{uud} \end{bmatrix} \begin{bmatrix} \delta \dot{q} \\ \delta \dot{u} \end{bmatrix} = \begin{bmatrix} \tilde{A}_{qq} & \tilde{A}_{qu} \\ \tilde{A}_{uqc} & \tilde{A}_{uuc} \\ \tilde{A}_{uqd} & \tilde{A}_{uud} \end{bmatrix} \begin{bmatrix} \delta q \\ \delta u \end{bmatrix} + \begin{bmatrix} 0_{(n+m) \times s} \\ \tilde{B}_u \end{bmatrix} \delta r \quad (3.60)$$

Equation (3.60) has a state space of dimension $n + o$, yet only $p = n - l + o - m$ of these quantities are independent. To address this issue, a smaller set of independent coordinates and speeds must be selected. To this end, we partition the generalized coordinates and generalized speeds as

$$\tilde{q} \triangleq \begin{bmatrix} q_i & q_d \end{bmatrix}^T = P_q^{-1} q \quad \tilde{u} \triangleq \begin{bmatrix} u_i & u_d \end{bmatrix}^T = P_u^{-1} u$$

where $P_q \in \mathbb{R}^{n \times n}$ and $P_u \in \mathbb{R}^{o \times o}$ are invertible permutation matrices which map an ordering which has the independent quantities ($q_i \in \mathbb{R}^{n-l}$, $u_i \in \mathbb{R}^{o-m}$) first, followed by the dependent quantities, ($q_d \in \mathbb{R}^l$, $u_d \in \mathbb{R}^m$) to the original ordering of the coordinates and speeds. We use

the notation P_{qi} and P_{qd} to denote the first $n - l$ and last l columns of P_q , respectively; similarly, P_{ui} is the first $o - m$ columns of P_u while P_{ud} is the last m columns of P_u .

Making use of P_q and the assumption that equation (3.50) is zero gives

$$\begin{aligned} \nabla_q f_c \delta q &= \nabla_q f_c P_q \delta \tilde{q} \\ &= \nabla_q f_c P_{qi} \delta q_i + \nabla_q f_c P_{qd} \delta q_d \\ \implies \delta q_d &= -(\nabla_q f_c P_{qd})^{-1} (\nabla_q f_c P_{qi}) \delta q_i \\ \implies \delta q &= \underbrace{[I_{n \times n} - P_{qd} (\nabla_q f_c P_{qd})^{-1} \nabla_q f_c] P_{qi}}_{\triangleq C_0} \delta q_i \end{aligned} \quad (3.61)$$

Making use of P_q , P_u , the assumption that equation (3.51) is zero, and taking equation (3.61) into account gives

$$\begin{aligned} \nabla_q f_v \delta q + \nabla_u f_v \delta u &= \nabla_q f_v C_0 \delta q_i + \nabla_u f_v \delta P_u \tilde{u} \\ &= \nabla_q f_v C_0 \delta q_i + \nabla_u f_v P_{ui} \delta u_i + \nabla_u f_v P_{ud} \delta u_d \\ \implies \delta u_d &= -(\nabla_u f_v P_{ud})^{-1} [\nabla_q f_v C_0 \delta q_i + \nabla_u f_v P_{ui} \delta u_i] \\ \implies \delta u &= \underbrace{-P_{ud} (\nabla_u f_v P_{ud})^{-1} \nabla_q f_v}_{\triangleq C_1} C_0 \delta q_i \\ &\quad + \underbrace{[I - P_{ud} (\nabla_u f_v P_{ud})^{-1} \nabla_u f_v] P_{ui}}_{\triangleq C_2} \delta u_i \end{aligned} \quad (3.62)$$

Making use of equations (3.61) and (3.62), we can rewrite equation (3.60) as

$$\begin{bmatrix} \tilde{M}_{qq} & 0_{n \times o} \\ \tilde{M}_{uqc} & \tilde{M}_{uuc} \\ \tilde{M}_{uqd} & \tilde{M}_{uud} \end{bmatrix} \begin{bmatrix} \delta \dot{q} \\ \delta \dot{u} \end{bmatrix} = \begin{bmatrix} (\tilde{A}_{qq} + \tilde{A}_{qu} C_1) C_0 & \tilde{A}_{qu} C_2 \\ (\tilde{A}_{uqc} + \tilde{A}_{uuc} C_1) C_0 & \tilde{A}_{uuc} C_2 \\ (\tilde{A}_{uqd} + \tilde{A}_{uud} C_1) C_0 & \tilde{A}_{uud} C_2 \end{bmatrix} \begin{bmatrix} \delta q_i \\ \delta u_i \end{bmatrix} + \begin{bmatrix} 0_{(n+m) \times s} \\ \tilde{B}_u \end{bmatrix} \delta r \quad (3.63)$$

This definitively establishes how the first time derivatives of coordinates and speeds (independent *and* dependent) depend, to first order, upon a selection of independent coordinates and independent speeds, for an arbitrary point of linearization. Note that the only requirement on this point of linearization is that it satisfy all the equations in Table 3.1; it may or may not be an equilibrium point.

3.5 Rolling disk, revisited

As shown at the end of Section 3.3, obtaining correct linearized equations in the presence of constraints requires a technique other than straightforward calculation of the Jacobian. In this section, we demonstrate that our linearization procedure yields eigenvalues which match published results [4], [12], [17]. The first step in the procedure is to form the matrices in Equation (3.59). Once obtained, these matrices can be evaluated at the equilibrium conditions and parameters of interest. We follow the standard approach to finding the equilibrium conditions by *choosing* the independent quantities and solving the constraint equations for the dependent quantities.

We first consider the case of upright steady cruise and begin by choosing the independent coordinates to be zero ($q_i^* = 0$, $i = 1, \dots, 5$), which implies (by appealing to the configuration constraint) that $q_6^* = -r \cos q_2 = -r$. This corresponds to the disk upright, the lowest point of the disc in contact with the ground, and the disk heading aligned with the \hat{n}_x unit vector. Next, we solve the velocity constraint equations for the dependent speeds in terms of the independent ones, and substitute these into the kinematic differential equations. This yields six equations with nine unknowns: u_i ($i = 1, 2, 3$) and \dot{q}_i ($i = 1, \dots, 6$). We *choose* yaw rate $\dot{q}_1^* = 0$, lean rate $\dot{q}_2^* = 0$, let spin rate \dot{q}_3^* be a free variable, and solve for the remaining six unknowns. This yields $u_1^* = u_3^* = 0$, $u_2^* = \dot{q}_3^*$, $\dot{q}_4^* = -r\dot{q}_3^*$, $\dot{q}_5^* = \dot{q}_6^* = 0$. Back substituting the first three of these results into the velocity constraint equations yields $u_4^* = -r\dot{q}_3^*$, $u_5^* = u_6^* = 0$. Finally, by evaluating the acceleration constraints f_a and the dynamic equations $f_2 + f_3$ at these conditions, we can solve for \dot{u}_i^* , $i = 1, \dots, 6$. We obtain $\dot{u}_1^* = \dot{u}_2^* = \dot{u}_3^* = \dot{u}_4^* = \dot{u}_5^* = 0$, and $\dot{u}_6^* = -r\dot{q}_3^{*2}$. The upright equilibrium conditions are thus parameterized in terms of the disc spin rate \dot{q}_3 ; for this upright configuration ($q_2 = 0$) it is convenient to introduce the forward speed $v \triangleq -r\dot{q}_3^*$.

Evaluating the matrices in equations (3.63) at these equilibrium conditions and subsequently solving these equations for $\delta\dot{q}$ and $\delta\dot{u}$ yields the linearized relationship between the time derivatives of all state variables and the independent state variables. By taking only the rows associated with the independent states, the linear relationship between the independent states and their time derivatives are formed. The eigenvalues of this coefficient matrix may be computed symbolically. Six of the eight eigenvalues of this matrix are zero, and the remaining two are

$$\lambda_{1,2} = \pm \frac{2\sqrt{1-3v^2}}{\sqrt{5}} \quad (3.64)$$

which has critical points at $v^* = \pm \frac{1}{\sqrt{3}}$ and matches previously published results [4], [12], [17]. For $|v| < v^*$, the disk is unstable, but for $|v| \geq v^*$, the eigenvalues are purely imaginary and are hence marginally stable. We omit the details of these calculations and direct the reader to the electronic supplementary material.

A more general analysis of the stability of the rolling disk equilibria considers the case where the disk lean q_2 is constant but not necessarily zero. To satisfy the dynamics, the yaw rate \dot{q}_1 and spin rate \dot{q}_3 must be chosen to satisfy

$$\frac{g}{r} \sin(q_2) + \frac{3}{2} \cos(q_2) \dot{q}_3 \dot{q}_1 + \frac{5}{4} \cos(q_2) \sin(q_2) \dot{q}_1^2 = 0 \quad (3.65)$$

which is a quadratic equation in the yaw rate \dot{q}_1 with roots

$$\dot{q}_1 = \frac{r}{2g \sin(q_2)} \left(-\frac{3}{2} \dot{q}_3 \cos(q_2) \pm \sqrt{-\frac{5g}{r} \sin^2(q_2) \cos(q_2) + \frac{9}{4} \dot{q}_3^2 \cos^2(q_2)} \right) \quad (3.66)$$

To be physically meaningful, these roots must be real, which gives rise to the following additional requirement

$$\dot{q}_3^2 \geq \frac{20g}{9r} \sin(q_2) \tan(q_2) \quad (3.67)$$

For a given set of parameters, the eigenvalues of the linearized dynamics can be parameterized by three quantities: lean q_2 , yaw rate \dot{q}_1 , and spin rate \dot{q}_3 (though only two can be independent because of the above restrictions). For this more general equilibrium condition, there are still six zero eigenvalues, while the two non-zero eigenvalues are

$$\lambda_{1,2} = \pm \sqrt{\frac{4}{5} \cos(q_2) - \dot{q}_1^2 - \frac{14}{5} \sin(q_2) \dot{q}_1 \dot{q}_3 - \frac{12}{5} \dot{q}_3^2} \quad (3.68)$$

which can be easily shown to reduce to Equation (3.64) when $q_2 = \dot{q}_1 = 0$. For a more detailed stability of steadily rolling disks, see [6], [12], [14].

3.6 Discussion

Most modern control techniques assume a system can be written as

$$\dot{x} = Ax + Br \quad (3.69)$$

(traditionally u is used instead of r , but Kane's method reserves u for generalize speeds, so we use r to denote what is traditionally written as u in control-focused literature). Equation 3.63

isn't square, so it doesn't fit into the standard linear system framework. To obtain a square system we define

$$\tilde{A} = \begin{bmatrix} (\tilde{A}_{qq} + \tilde{A}_{qu}C_1)C_0 & \tilde{A}_{qu}C_2 \\ (\tilde{A}_{uqc} + \tilde{A}_{uuc}C_1)C_0 & \tilde{A}_{uuc}C_2 \\ (\tilde{A}_{uqd} + \tilde{A}_{uud}C_1)C_0 & \tilde{A}_{uud}C_2 \end{bmatrix} \quad (3.70)$$

$$\tilde{B} = \begin{bmatrix} 0_{(n+m) \times s} \\ \tilde{B}_u \end{bmatrix} \quad (3.71)$$

from (3.63) and

$$A' \triangleq \tilde{M}^{-1}\tilde{A} \quad (3.72)$$

$$B' \triangleq \tilde{M}^{-1}\tilde{B} \quad (3.73)$$

where $A' \in \mathbb{R}^{(o+n) \times (o-m+n-l)}$, $B' \in \mathbb{R}^{(o+n) \times s}$. We can extract the rows corresponding to the independent states by defining

$$P' \triangleq \begin{bmatrix} P_{qi} & O_{n \times (o-m)} \\ O_{o \times (n-l)} & P_{ui} \end{bmatrix} \quad (3.74)$$

$$A \triangleq P'^T A' \quad (3.75)$$

$$B \triangleq P'^T B' \quad (3.76)$$

where $P' \in \mathbb{R}^{(o+n) \times (o-m+n-l)}$. Defining $x_i = [\delta q_i, \delta u_i]^T$ yields the square state space system $\dot{x}_i = Ax_i + Br$ to which standard linear systems analyses may be applied. It is worth noting that the rows of A' and B' which correspond to dependent states can be used in output or measurement equations of a linear state space model (e.g., measurements from an accelerometer, or measurements of dependent speeds).

Kane's dynamical equations are typically formulated for one particular choice of dependent speeds, and computer code is generated for that single choice of dependent speeds. It is possible, however, to output computer code for the unconstrained equations of motion (equation (3.12)), along with the velocity constraint coefficient matrix (equation (3.10)), such that the independent choice of speeds and the constrained equations of motion (equation (3.23)) are formed at the time of numerical evaluation (as opposed to choosing a particular set of dependent speeds during symbolic derivation). There are several reasons why this can be desirable: 1) the configuration

of the system in question changes significantly enough during simulation that no single choice of independent speeds will work for all regions of the configuration space; 2) the choice of parameters greatly affects which states should be selected to be dependent; and 3) to choose the dependent states which minimize the effect of numerical round off. This approach can also be applied to computation of the linearized dynamics. A task where the ability to switch the choice of dependent states “on the fly” is extremely helpful is in the computation of Lyapunov characteristic exponents (LCE’s). When computing LCE’s, time simulation of the nonlinear dynamics *and* linearization at each point along the trajectory a required [1], [2], [19]; having a computer code that makes it easy to switch choice of dependent states at the time of numerical evaluation is of great benefit in this case. The matrices which must be used to determine the best choice of coordinates and speeds are $\nabla_q f_c P_{qd}$ and $\nabla_u f_v P_{ud}$; these matrices appear in equations (3.61) and (3.62). If for some choice of dependent coordinates or dependent speeds either of these matrices are singular or nearly singular, a different set of dependent coordinates and dependent speeds should be chosen. These matrices generally depend only upon parameters and coordinates (not speeds). While some systems may permit a choice of independent state variables which are valid for all configurations of interest, others may not. Methods for automatically selecting the “best” choice of independent state variables are discussed in [16]; they involve computing the singular value decomposition $\nabla_q f_c$ and $\nabla_u f_v$ to determine a set of independent states which will ensure the non-singularity of the aforementioned matrices.

3.7 Conclusions

A procedure for forming linearized equations of motion for constrained multi-body systems has been presented. This procedure can be implemented symbolically or numerically, and handles configuration, velocity, and acceleration level constraints. The coefficient matrices in equation (3.63) can generally be computed symbolically and output as efficient C/C++/Fortran routines which may be compiled into highly efficient machine code. This permits library routines which are both highly efficient (no finite differences) and very general (arbitrary system parameters and linearization point).

The procedure has been implemented symbolically in the `sympy.physics.mechanics` sub-module of the open source symbolic manipulator SymPy [18]. In addition to the rolling disk example, we have also applied it to the extended Whipple bicycle model described in Chapter 2 [15], and obtained results that matched (to at least 14 digits) the eigenvalues published for a

benchmark set of parameters [8]. This latter example provides a challenging and rigorous test of the procedure which strongly indicates both its utility and its correctness.

References for Chapter 3

- [1] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-marie Strelcyn. “Lyapunov Characteristic Exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 1: Theory”. In: *Meccanica* 15.1 (Mar. 1980), pages 9–20. ISSN: 0025-6455. DOI: [10.1007/BF02128236](https://doi.org/10.1007/BF02128236). URL: <http://link.springer.com/10.1007/BF02128236> (cited on page 45).
- [2] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-marie Strelcyn. “Lyapunov Characteristic Exponents for smooth dynamical systems and for Hamiltonian systems; A method for computing all of them. Part 2: Numerical application”. In: *Meccanica* 15.1 (Mar. 1980), pages 21–30. ISSN: 0025-6455. DOI: [10.1007/BF02128237](https://doi.org/10.1007/BF02128237). URL: <http://link.springer.com/10.1007/BF02128237> (cited on page 45).
- [3] Oene Bottema. “On the small vibrations of non-holonomic systems”. In: *Proceedings Koninklijke Nederlandse Akademie van Wetenschappen*. 1949, pages 848–850 (cited on page 25).
- [4] Thomas R. Kane and David A. Levinson. *Dynamics: Theory and Applications*. McGraw-Hill, Inc, 1985. ISBN: 978-0070378469 (cited on pages 24, 31, 42, 43).
- [5] Ju Seok Kang, Sangwoo Bae, Jang Moo Lee, and Tae Oh Tak. “Force Equilibrium Approach for Linearization of Constrained Mechanical System Dynamics”. In: *Journal of Mechanical Design* 125.1 (2003), page 143. ISSN: 10500472. DOI: [10.1115/1.1541631](https://doi.org/10.1115/1.1541631). URL: <http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=1447404> (cited on page 25).
- [6] Alexander S. Kuleshov. “The steady rolling of a disc on a rough plane”. In: *Journal of Applied Mathematics and Mechanics* 65.1 (Feb. 2001), pages 171–173. ISSN: 00218928. DOI: [10.1016/S0021-8928\(01\)00020-X](https://doi.org/10.1016/S0021-8928(01)00020-X). URL: <http://linkinghub.elsevier.com/retrieve/pii/S002189280100020X> (cited on page 43).
- [7] Martin Lesser. “A Geometrical Interpretation of Kane’s Equations”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 436.1896 (Jan. 1992), pages 69–87. ISSN: 1364-5021. DOI: [10.1098/rspa.1992.0005](https://doi.org/10.1098/rspa.1992.0005). URL: <http://rspa.royalsocietypublishing.org/cgi/doi/10.1098/rspa.1992.0005> (cited on page 28).

- [8] Jaap P. Meijaard, Jeremy M. Papadopoulos, Andy Ruina, and Arend L. Schwab. “Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463.2084 (Aug. 2007), pages 1955–1982. ISSN: 1364-5021. DOI: [10.1098/rspa.2007.1857](https://doi.org/10.1098/rspa.2007.1857). URL: <http://rspa.royalsocietypublishing.org/cgi/doi/10.1098/rspa.2007.1857> (cited on page 46).
- [9] Bruce P. Minaker and Robert J. Rieveley. “Automatic generation of the non-holonomic equations of motion for vehicle stability analysis”. In: *Vehicle System Dynamics* 48.9 (Sept. 2010), pages 1043–1063. ISSN: 0042-3114. DOI: [10.1080/00423110903248702](https://doi.org/10.1080/00423110903248702). URL: <http://www.tandfonline.com/doi/abs/10.1080/00423110903248702> (cited on page 26).
- [10] Paul C. Mitiguy and Thomas R. Kane. “Motion Variables Leading to Efficient Equations of Motion”. In: *The International Journal of Robotics Research* 15.5 (Oct. 1996), pages 522–532. ISSN: 0278-3649. DOI: [10.1177/027836499601500507](https://doi.org/10.1177/027836499601500507). URL: <http://ijr.sagepub.com/cgi/doi/10.1177/027836499601500507> (cited on page 27).
- [11] Dan Negrut and Jose L. Ortiz. “A Practical Approach for the Linearization of the Constrained Multibody Dynamics Equations”. In: *Journal of Computational and Nonlinear Dynamics* 1.3 (2006), page 230. ISSN: 15551423. DOI: [10.1115/1.2198876](https://doi.org/10.1115/1.2198876). URL: <http://computationalnonlinear.asmedigitalcollection.asme.org/article.aspx?articleid=1393955> (cited on page 25).
- [12] Ju. I. Neimark and N. A. Fufaev. *Dynamics of Nonholonomic Systems*. American Mathematical Society, 1972, page 518. ISBN: 978-0-8218-3617-0 (cited on pages 42, 43).
- [13] Charles P. Neuman and John J. Murray. “Linearization and sensitivity functions of dynamic robot models”. In: *IEEE Transactions on Systems, Man, and Cybernetics SMC-14.6* (Nov. 1984), pages 805–818. ISSN: 0018-9472. DOI: [10.1109/TSMC.1984.6313309](https://doi.org/10.1109/TSMC.1984.6313309). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6313309> (cited on page 26).
- [14] Oliver M. O'Reilly. “The dynamics of rolling disks and sliding disks”. In: *Nonlinear Dynamics* 10.3 (July 1996), pages 287–305. ISSN: 0924-090X. DOI: [10.1007/BF00045108](https://doi.org/10.1007/BF00045108). URL: <http://link.springer.com/10.1007/BF00045108> (cited on page 43).

- [15] Dale Lukas Peterson. *libbicycle*. 2013. URL: <https://github.com/hazelnusse/libbicycle> (cited on page 45).
- [16] Keith Jerome Reckdahl. “Dynamics and control of mechanical systems containing closed kinematic chains”. Ph.D. Stanford University, 1996, page 231 (cited on pages 31, 45).
- [17] Arend L. Schwab and Jaap P. Meijaard. “Dynamics of Flexible Multibody Systems with Non-Holonomic Constraints: A Finite Element Approach”. In: *Multibody System Dynamics* 10.1 (2003), pages 107–123. DOI: [10.1023/A:1024575707338](https://doi.org/10.1023/A:1024575707338) (cited on pages 26, 38, 42, 43).
- [18] SymPy Development Team. *Sympy: Python library for symbolic mathematics*. 2013. URL: <http://www.sympy.org> (cited on page 45).
- [19] Firdaus E. Udwadia and Hubertus F. von Bremen. “An efficient and stable approach for computation of Lyapunov characteristic exponents of continuous dynamical systems”. In: *Applied Mathematics and Computation* 121.2-3 (June 2001), pages 219–259. ISSN: 00963003. DOI: [10.1016/S0096-3003\(99\)00292-1](https://doi.org/10.1016/S0096-3003(99)00292-1). URL: <http://linkinghub.elsevier.com/retrieve/pii/S0096300399002921> (cited on page 45).

Chapter 4

Robot bicycle

4.1 Introduction and motivation

The Whipple bicycle model qualitatively predicts behavior commonly observed in physical bicycles such as a stable speed range and counter steer. To quantitatively assess the accuracy of these predictions, in as controlled a setting as possible, a robot bicycle was constructed. By *robot bicycle*, we simply mean a bicycle that can balance without a human rider; this key feature permits experiments to be performed where dynamic quantities can be measured without the influence of undesirable and difficult to measure human inputs such as limb motion. Measurement of the bicycle motion that results from the application of precisely known bicycle inputs permits direct comparisons to predictions of the Whipple model without the need to model the human. Experiments such as those conducted by Kooijman [13], wherein a riderless bicycle was pushed and “ghost ridden”, do not maintain a constant speed (the bicycle decelerates as soon as it is released by the pusher), nor can a specific initial speed be set easily or precisely. Additionally, in [13], only the uncontrolled bicycle dynamics were examined; as such, it was limited to speeds for which the uncontrolled bicycle is stable. This work extends the work of [13] by accurately maintaining a constant speed, and stabilizing the bicycle for speeds below the weave speed.

This chapter describes the design and construction (mechanical, electrical, and software) of the robot bicycle and presents data collected in a set of experiments. Physical parameter measurement, sensor calibration, and all mechanical and electrical components (purchased and fabricated) are documented here. The mechanical and electrical sections are somewhat low level and tedious but are included for completeness. Readers interested in the high level control system design may consider skipping to [section 4.4](#). Finally, a description and analysis of a set

of experiments is presented.

4.2 Mechanical construction

4.2.1 Design

The robot bicycle is shown in Figures 4.1-4.5. Where possible, readily available bicycle parts were used. A number of components were fabricated to interface standard bicycle parts (frame, fork steer tube, disc brake tabs) with equipment not typically found on a bicycle (batteries, fork motor, safety casters, and electrical equipment). This section provides a high level overview of the construction; more details of each subsystem can be found in the subsequent subsections pertaining to each piece of equipment.



Figure 4.1: Robot bicycle viewed from the right side.

A Surly 4130 cromoly steel bicycle frame and fork were used [15]. The frame and fork permit the use of 26" or 700c wheels, cantilever or disc brakes, and were chosen because its rugged steel construction permitted modification without fear of compromising its structural integrity. The



Figure 4.2: Robot bicycle viewed from the rear right side. The wireless antenna is visible behind the batteries, and the attachment of the training wheel struts to the bicycle frame can also be seen.



Figure 4.3: Robot bicycle viewed from the rear.



Figure 4.4: Robot bicycle viewed from the rear left. The motor wiring exits the left side of the rear wheel axle. The rear wheel optical encoder is mounted to the rear disc brake tabs and is driven by the rear wheel with a toothed kevlar belt combined with 100 tooth and 25 tooth pulleys.

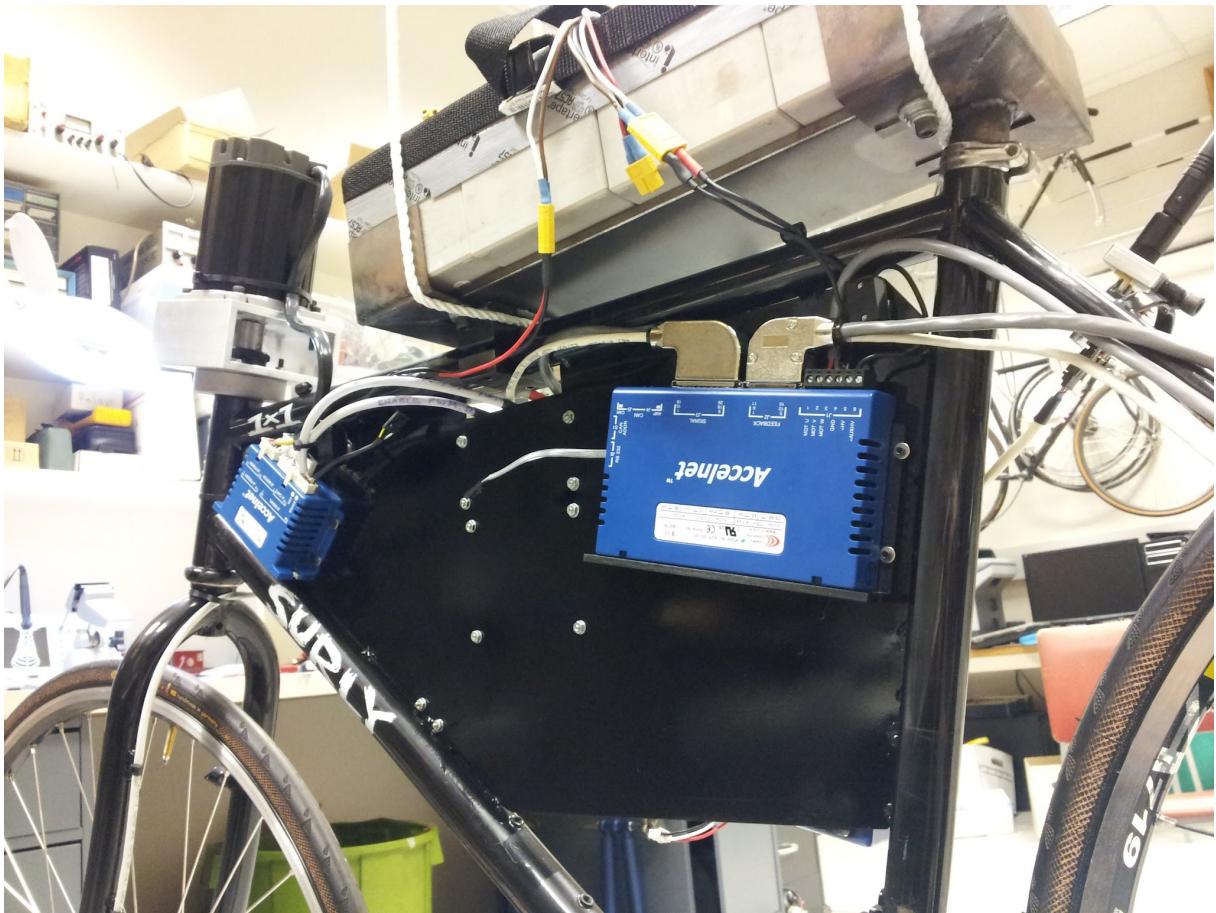


Figure 4.5: Close up of battery plate, fork motor mount, and digital motor drives. The cylindrical plug interfacing the fork motor spindle and the fork steer tube is visible inside the aluminum box section directly beneath the steer motor.

frame was modified by welding an 18 gauge mild steel sheet to the inside of the front triangle to provide a surface on which to mount the digital motor drives, microcontroller development board and batteries.

To prevent damage to the robot bicycle in the case of a fall, custom training wheels were designed and built. The training wheels were composed of casters positioned approximately 18" from the frame plane and roughly even with the bottom bracket. The height of the casters permitted the bicycle to lean approximately 20° before they touched the ground. Each caster wheel was bolted to a 2" long 2"x4" aluminum box section which was in turn bolted to one side of a small vertical steel plate. On the other side of the steel plate, three 1/2"x0.0625" 4130 cromoly steel round tubes were welded – each tube connected the steel plate (and hence the casters) to the bicycle frame and provided sufficient strength and rigidity to handle the loads

transmitted during a fall. Welded to the inboard ends of the three 1/2" tubes were brackets which interfaced with the seat tube, down tube, and bottom bracket. The brackets on either side of the frame were bolted together, thereby sandwiching the bicycle frame.

Other custom-fabricated mechanical components were the wheel optical encoder mounts, fork motor mounting hardware, and battery support plate. The details of each of these components are discussed in [4.3.2](#), [4.3.3](#), and [4.3.4](#), respectively.

4.2.2 Physical parameter measurement

The twenty three physical parameters (described in [Chapter 2](#)) of the robot bicycle were estimated by measuring first the benchmark parameter set, and then converting those parameters to the gyrostat parameter set using the equations presented in [subsection 2.2.4](#). It was assumed that the wheel minor radius was zero (i.e., that the wheels were knife edged). The resulting parameters are presented in [Table 4.1](#).

	Rear gyrostat	Front gyrostat	Units
I_{xx}	1.542	0.183	kg m^2
I_{yy}	3.557	0.226	kg m^2
I_{zz}	3.014	0.069	kg m^2
I_{xz}	0.839	-0.010	kg m^2
J	0.114	0.092	kg m^2
m	34.1	2.95	kg
R	0.336	0.336	m
r	0	0	m
a	0.514	-0.021	m
b	-0.219	-0.152	m
c	0.963	-0.048	m
l_s	0.343		m

Table 4.1: Robot bicycle physical parameters.

4.3 Electrical system description

4.3.1 Microcontroller

All functionality related to control, measurement, and user interaction was implemented by programming an Olimex STM32-H407 [14] development board. A summary of the board functionality utilized is shown in Table 4.2. The C++11 [11] programming language was used to

CPU	ST Microelectronics STM32F407ZGT6 @ 168 MHz 1MiB flash memory, 192KiB ram, 32-bit memory address space Thumb-2 instruction set, single precision floating point
Timers	3 x 16-bit timers in quadrature counting mode 32-bit count up timer @ 4 MHz PWM generation @ 2.563 kHz; $2^{16} - 1$ distinct duty cycles
Communication	UART peripheral (115,200 baud 8N1) to XBee Pro radio I2C peripheral @ 400 kHz to communicate with IMU SDIO peripheral to log data to micro SD flash memory card JTAG peripheral for flashing and debugging programs
General Purpose	Input: momentary switch, motor faults, fork encoder index Outputs: motor enable and direction, lean and steer LEDs

Table 4.2: Microcontroller development board functionality.

implement all functionality executed on the development board CPU. Certain language features such as dynamic memory allocation, runtime type information, and exceptions were not used due to associated overhead (generated code size or runtime efficiency). The compiler Version 4.7 (update 2) of the GCC ARM Embedded [2] tool chain was used to compile and link source code to machine executable code. This ARM maintained version of GCC is customized to generate efficient machine instructions for ARM embedded processors.

The real time operating system ChibiOS/RT [8] provided functionality for concurrent execution, thread synchronization primitives, a file system for data logging, an extensible interactive serial shell, and a high level interface to the development board hardware peripherals (UART, I2C, SDIO, and GPIOs). ChibiOS/RT directly supports a large number of development boards, including the Olimex STM32-H407, which made building and running test code convenient and relatively painless. Additionally, the very active user community and excellent documentation of

ChibiOS/RT made it easy to troubleshoot problems, ask questions, and get help while developing the firmware.

The development board was attached to the right side of the sheet in the front triangle of the frame. Slightly above the development board on the electrical sheet was a small electrical prototype board which held several small integrated circuits, Molex connector housings, and two LEDs. The prototype board connections were soldered directly to the development board so they effectively acted as a single unit. The Molex connector housings connected several units to the microcontroller: 1) the optical encoders from both wheels and fork, 2) the XBee wireless radio, and 3) the enable, fault, direction, and signal ground pins from the digital motor drives. Wired directly to the microcontroller development board was a small 7.2V battery ([subsection 4.3.4](#)) and the inertial measurement unit ([subsection 4.3.2](#)).

4.3.2 Sensors

The robot bicycle was equipped with three optical encoders which measured the wheel angles and the steer angle. All three optical encoders were differentially signalled for noise robustness. The front wheel encoder was mounted to the fork but was left disconnected during all experiments due to risk of damage when the front fork spun more than 180 degrees from straight ahead (which happened several times in the testing phase of the robot bicycle design). The steer angle encoder was integrated into the steer motor [16] with the optical disc fixed to the motor shaft and provided 20000 quadrature counts per revolution ($\pi \times 10^{-4}$ rad / quadrature count) as well as an index signal once per revolution. The wheel optical encoders [17] provided an effective 800 quadrature counts per revolution ($0.25\pi \times 10^{-2}$ rad / quadrature count) without an index channel. The wheel encoders were mounted with a custom aluminum adapter to the frame and fork disc brake mounting tabs. The rear wheel optical encoder is visible in [Figure 4.4](#); the front wheel optical encoder was mounted similarly. Since the wheels were symmetric about their axis of rotation, no calibration was needed for wheel encoders.

The steer encoder was calibrated whenever the cylindrical plug interfacing the fork motor spindle to the fork steer tube was moved (this was only moved twice, so only two calibrations were done; the cylindrical plug is visible in [Figure 4.5](#)). A square T fixture was built to hold the front wheel axle parallel with the rear wheel axle; this was defined to be the zero steer configuration. The fixture can be seen in [Figure 4.6](#). The steer angle calibration involved the following steps:

- Suspend the bicycle frame from the ceiling and remove both wheels.
- Rigidly mount the steer calibration fixture into the wheel dropouts, thereby setting the steer angle $\delta = 0$.
- Turn the microcontroller on and issue the `calibrate` (see subsection 4.5.3 command. This calls a function which zeros the steer encoder count and waits for the steer encoder index.
- Remove the fixture from the front dropouts, and rotate the fork left to right 16 times. This causes the `calibrate` command to record the steer encoder count each time the steer index signal transitions from low to high or high to low.
- Record the steer offset presented by the `calibrate` function. The offset is the signed integer number of quadrature counts the index is away from steer $\delta = 0$.

On microcontroller reset, the steer encoder count is set to zero regardless of the position of the fork. By permanently recording the number of quadrature counts the steer index is from the zero steer configuration, this offset can be set during a fork homing procedure when the steer index is triggered. This functionality was implemented in the `homefork` function and was run every time the microcontroller was reset.

A combined rate gyroscope and accelerometer MEMS sensor [12] was fixed to the underside of the battery pack plate as shown in Figure 4.7. The gyroscope and accelerometer sensor axes were assumed to be aligned with each other since they are manufactured on the same piece of silicon. The sensor $\hat{s}_x - \hat{s}_y$ plane was approximately parallel to the plane of the battery plate, with \hat{s}_y pointed approximately forward, \hat{s}_y to the left, and \hat{s}_z down.

As described in Chapter 1, the bicycle model introduces set of dextral unit vectors fixed to the rear frame R of the bicycle, with \hat{r}_z parallel to the steer axis and down, \hat{r}_y perpendicular to the frame plane and to the right, and $\hat{r}_x = \hat{r}_y \times \hat{r}_z$ (\hat{r}_x points forward and slightly up when the bicycle is in the reference configuration). Fixed to the sensor S are a set of dextral unit vectors with \hat{s}_x and \hat{s}_y in the plane of the integrated circuit, and \hat{s}_z normal to the plane of the integrated circuit. To orient the sensor relative to the bicycle, first align S with R , then apply the following successive body-fixed ZXY rotations: $\alpha - \pi/2, \beta, \gamma$. The first rotation was offset by $-\pi/2$ so that all three angles were near zero.



Figure 4.6: Calibration of steer angle encoder, accelerometer, and rate gyroscope ($\phi = -\pi/2$ configuration). The steer calibration fixture ensured $\delta = 0$ and provided surfaces to rest the bubble levels (visible on far side of bicycle frame). Two turnbuckles were used to make minor orientation adjustments to level the frame (visible on left side of image).

To determine the three angles α , β , and γ , the static acceleration was measured in three sensor directions (\hat{s}_x , \hat{s}_y , \hat{s}_z) in six unique orientations. The six orientations are those that would be obtained if R were aligned with and fixed to the edges of a perfect cube and the cube was laid on a level surface on each of its six sides. These six configurations correspond to the following bicycle orientations: $\phi = \pm\pi/2$ (frame plane horizontal), $\phi = 0$ and $\theta_r = \{-\pi/2, 0, \pi/2, \pi\}$ (frame plane vertical, two horizontal steer axis orientations and two vertical steer axis orientations). The sensed acceleration in the three sensor directions was recorded for approximately sixty seconds while the bicycle was in each orientation.

The six orientations were obtained by suspending the bicycle by ropes from three points of attachment and using a turnbuckle on two of the ropes to level the desired surface in two directions. Two precision horizontal bubble levels were attached to the frame 90 degrees apart to permit levelling in both directions. Once levelled, the bicycle was left to rest for approximately 5 minutes to allow for small swinging and twisting oscillations to die out; this was also aided by suspending a weight from the frame with fishing line and hanging it into a bucket of water. The bucket of water provided some dissipation which helped damp out vibrations. Once stationary

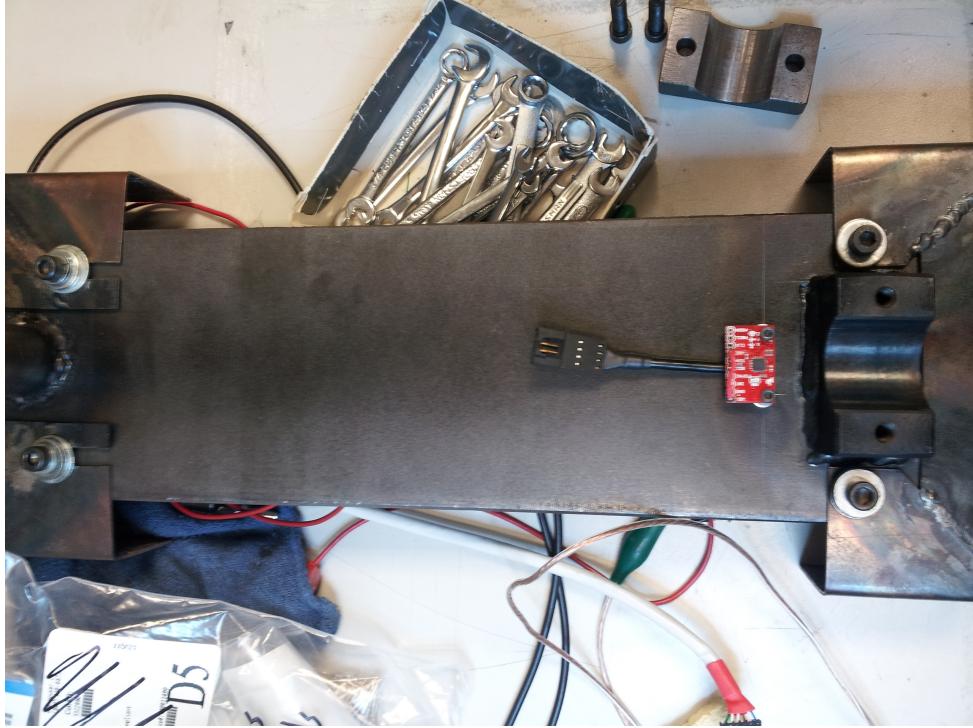


Figure 4.7: Rate gyroscope and accelerometer sensor placement on bottom of battery support plate. The top tube clamshell bracket is visible on the right side of the image (forward end of battery plate). On the left side of the image the seat post is partially visible. The mounting hardware for the battery box is also visible.

(within the limits of the vibrating building), data collection was initiated.

When at rest, the accelerometer senses the gravitational field as if it were accelerating *upwards* at g , this corresponds to $-g\hat{\mathbf{n}}_z$. Resolving $-g\hat{\mathbf{n}}_z$ into the three sensor measurement directions yields

$$-g\hat{\mathbf{n}}_z \cdot \hat{\mathbf{s}}_x = g(-s_\alpha s_\beta s_\gamma s_\phi + s_\alpha s_\theta c_\gamma c_\phi + s_\beta s_\gamma s_\theta c_\alpha c_\phi + s_\gamma c_\beta c_\phi c_\theta + s_\phi c_\alpha c_\gamma) \quad (4.1)$$

$$-g\hat{\mathbf{n}}_z \cdot \hat{\mathbf{s}}_y = g(-s_\alpha s_\phi c_\beta - s_\beta c_\phi c_\theta + s_\theta c_\alpha c_\beta c_\phi) \quad (4.2)$$

$$-g\hat{\mathbf{n}}_z \cdot \hat{\mathbf{s}}_z = g(s_\alpha s_\beta s_\phi c_\gamma + s_\alpha s_\gamma s_\theta c_\phi - s_\beta s_\theta c_\alpha c_\gamma c_\phi + s_\gamma s_\phi c_\alpha - c_\beta c_\gamma c_\phi c_\theta) \quad (4.3)$$

where s_x and c_x , $x = \alpha, \beta, \gamma, \phi, \theta$ are abbreviations of $\sin x$ and $\cos x$. The goal of the calibration was to convert the measured acceleration to the true acceleration, and for this reason the following slightly non traditional (yet equally representative) form of the acceleration sensor model

was used

$$\begin{bmatrix} -g\hat{\mathbf{n}}_z \cdot \hat{\mathbf{s}}_x \\ -g\hat{\mathbf{n}}_z \cdot \hat{\mathbf{s}}_y \\ -g\hat{\mathbf{n}}_z \cdot \hat{\mathbf{s}}_z \end{bmatrix} = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{xy} & S_{yy} & S_{yz} \\ S_{xz} & S_{yz} & S_{zz} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (4.4)$$

where a_x, a_y, a_z are the raw sensor measurements (in units of bits), b_x, b_y, b_z are biases (in units of m/s^2), S_{xx}, S_{yy}, S_{zz} are the sensitivities, and S_{xy}, S_{yz}, S_{xz} are the cross axis sensitivities, both of which have units of $\text{m/s}^2/\text{bit}$. Cross axis sensor symmetry was assumed, i.e., $S_{xy} = S_{yx}$.

Equating the right side of Equations 4.1-4.3 with the right side of Equation 4.4, evaluating at the value of lean ϕ and pitch θ corresponding with a particular configuration, and taking the time series mean of each of the raw measurements a_x, a_y, a_z , we obtain three equations with twelve unknowns. Repeating for each of the six configurations, we obtain an overdetermined system of eighteen equations in twelve unknowns. The twelve unknowns are the six sensitivities, the three biases, and the three orientation angles relating the sensor frame to the bicycle frame. This overdetermined system of equations was solved by the method of least squares to obtain the following sensitivities, biases, and orientation angles

$$\begin{bmatrix} S_{xx} \\ S_{yy} \\ S_{zz} \\ S_{xy} \\ S_{yz} \\ S_{xz} \end{bmatrix} = \begin{bmatrix} 5.9898 \times 10^{-4} \\ 5.9534 \times 10^{-4} \\ 5.8288 \times 10^{-4} \\ -5.4766 \times 10^{-7} \\ -1.6455 \times 10^{-6} \\ 1.9272 \times 10^{-6} \end{bmatrix} \text{ m/s}^2/\text{bit} \quad (4.5)$$

$$\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \begin{bmatrix} -0.5700 \\ 0.0514 \\ 1.1690 \end{bmatrix} \text{ m/s}^2 \quad (4.6)$$

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} 0.0031 \\ 0.3230 \\ -0.0182 \end{bmatrix} \text{ rad} \quad (4.7)$$

The diagonal entries of S were very close to the manufacturer specified sensitivity of 5.9876×10^{-4} $\text{m/s}^2/\text{bit}$. The diagonal entries were greater than the off-diagonal entries by more than two orders of magnitude, indicating the cross axis sensitivity was less than 1%, also in line with

the manufacturers specifications. The means of three rate gyroscope measurements axes were computed for each static configuration, and were found to be independent of configuration (as expected). These means were used as biases for the measurement of the bicycle frame angular velocity and were found to be

$$E[\omega_x] = -0.1204 \text{ rad/s} \quad (4.8)$$

$$E[\omega_y] = 0.0316 \text{ rad/s} \quad (4.9)$$

$$E[\omega_z] = 0.0100 \text{ rad/s} \quad (4.10)$$

Since a rate table or other convenient means of calibrating the gyroscope sensitivities was not available, the manufacturer published gyroscope sensitivities were used.

Two LEDs and a momentary switch were mounted on the right side of the bicycle. The green LED was illuminated when the lateral acceleration was relatively small compared to the total acceleration: $\frac{|\mathbf{a} \cdot \hat{\mathbf{r}}_y|}{\|\mathbf{a}\|} < \frac{\pi}{180}$; this visually indicated when the bicycle frame plane was nearly vertical. Similarly, the yellow LED was illuminated when the steer angle was less than $1^\circ = \frac{\pi}{180}$ rad. Finally, the momentary switch was mounted on the right side of the steer motor mount. When depressed, the switch grounded a microcontroller pin and indicated that the bicycle was being balanced manually; when released, the pin state was pulled up to +3.3V and indicated the bicycle was not being balanced manually. The state of this pin was sampled and recorded along with all other sensor signals; when the switch was closed, a 0 was recorded, when the switch was open, a 1 was recorded (see [Figure 4.10](#) for an example of the signal).

4.3.3 Actuators

The robot bicycle was equipped with a rear wheel hub motor to drive the rear wheel and a fork motor to steer the bicycle. Both motors were brush-less DC motors and were interfaced to the microcontroller with Copley Controls digital motor drives (fork: ACJ-055-18 [\[5\]](#), rear wheel: ADP-090-36 [\[6\]](#)). Both digital motor drives were configured in current control mode and internally implemented a PI current (with gains automatically selected via manufacturer software based upon motor parameters) controller that operated at 30 kHz. The current commands were generated by the microcontroller as 3.3V pulse width modulated (PWM) signals which were converted by the motor drives to the appropriate high voltage, high current PWM signals to the motor windings. The bicycle control system was designed to generate applied joint torques as control signals, which were then scaled by the respective motor torque constants before

generating the current command PWM signal.

The rear wheel was built with an electric hub motor [1]. The manufacturer supplied rim and spokes were of extremely low quality (very weak, untrue, and noticeably inertially non-symmetric), so they were replaced with DT Swiss 2.0 mm stainless steel spokes and Mavic model A719, 700c diameter, 36 hole rim. The rear hub axle served as the motor stator and armature with the three motor power leads exiting the left side of the axle. The motor field magnets were fixed to the inside of the hub shell and rotated along with the spokes, rim, and tire when current was applied to the motor windings. This motor configuration is commonly known as the “outrunner” configuration. No manufacturer specifications were available for this motor, so the motor torque constant was determined experimentally by applying a constant rear wheel current I_R while the wheel was off the ground and measuring the angular response. The spin inertia J_r of the rear wheel was measured (subsection 4.2.2), and the idealized DC motor equation $J_r \ddot{\theta}_r = K_t I_r$ was integrated twice with respect to time (constant current assumed and $\theta_r(0) = \dot{\theta}_r(0) = 0$) to obtain $\theta_r = \frac{K_t I_r}{2J_r} t^2$. A least squares fit was then used to estimate $K_t = 6.6$ Nm/A. Since a PI speed controller was implemented to control rear wheel rate, it was not critical to know K_t precisely (PI controllers perform reasonably well even when system parameters are not well known); for this reason we didn’t expend any effort to more accurately determine K_t .

The fork motor [16] was bolted to an aluminum box section which in turn was bolted to a custom upper headset. The custom upper headset was pressed into the upper end of the bicycle steer tube with green Loctite to ensure it would not twist relative to the frame when motor torques were applied. In contrast to the rear wheel hub motor, the fork motor stator and armature windings were fixed to the outer portion of the motor (fixed to the bicycle frame) and the field magnets were fixed to the motor output shaft. The motor output shaft was equipped with a key-way which was mated to a circular plug fixed to the inside of the steer tube. The circular plug was rigidly attached to the steer tube in the same way a threaded bicycle stem wedge expander attaches to the bicycle steer tube. The upper portion of the circular plug (which mates with the motor shaft) was bored with a hole to match the diameter of the motor output shaft, and an internal key-way was cut with a wire-cut electrical discharge machine. Once the motor was secured, a set screw was threaded into the upper portion of the plug to make contact with the motor key. This design permitted the fork to be driven directly by the motor without the use of a gearbox, and still permitted the motor to be removed easily if

necessary. Most importantly, the design had no backlash between the motor shaft and the fork. A previous design was attempted which made use of a precision gearbox, but it was found to have unacceptable levels of backlash. Since the sign of the steer angle rate frequently changes during normal operation of a bicycle, any backlash between the motor fork is extremely undesirable.

4.3.4 Batteries

Four 12.0V sealed lead acid (SLA) batteries were wired in series and used to power the motors and the digital motor drives. The SLA batteries were arranged in a row of four and tightly bound to each other with duct tape and nylon strapping. All other electronics were powered with a two-cell 7.2V lithium polymer battery [10] which was fastened to the right side of the electrical panel with velcro and a small bungee cord.

The SLA batteries were securely fastened to the bicycle frame with a 1/4"x4"x18" steel plate to support the bottom of the batteries, and a sheet metal box on either end of the plate to maintain the lateral and longitudinal position of the batteries relative to the plate. The plate was rigidly attached to the bicycle frame with a steel seat post on the rear end and a steel top tube bracket on the forward end. The seat post and bracket were welded to the bottom of the plate such that when the seat post was inserted into the frame, the top tube bracket interfaced with the top of the top tube to align the steel plate symmetrically with respect to the bicycle frame. The top tube bracket was a clamshell design, with the top half welded to the bottom of the steel plate and the bottom half placed on the underside of the top tube and then bolted to the top half, thereby securing the plate to the tube (see [Figure 4.7](#)). When the bicycle was in the upright reference configuration, the battery plate was approximately horizontal. In addition to the battery box to maintain the lateral and longitudinal position of the batteries with respect to the frame, nylon straps were used to secure the batteries onto the plate.

4.3.5 Wireless communication

A pair of XBee Pro [7] wireless radios were used as a serial cable replacement between the robot bicycle and a nearby laptop computer. Commands were sent to the robot bicycle by typing them into a serial terminal program [4] which transmitted the text as a simple character stream through the USB port of the computer to the XBee Pro radio, which in turn transmitted the commands wirelessly. A shell thread running on the robot bicycle microcontroller monitored the serial port for received commands along with optional command arguments. When a valid command was received, the corresponding function was executed. A list of available commands

are detailed in [subsection 4.5.3](#).

4.4 Control system design

This section details the theoretical framework as well as the implementation details of the control system for the robot bicycle. Lower level details and user interface considerations are detailed in [subsection 4.4.1](#) and [subsection 4.5.3](#).

4.4.1 Data logging

Data was written at 200 Hz, in binary format, to a single file per “run” on the micro SD card. The data format used Google Protocol Buffers [9], a platform independent data interchange format used by Google. In addition to abstracting away platform dependent issues (byte order, word size, etc.), this data format permitted data fields to be marked as required or optional, and new data fields could be easily added without losing the ability to easily work with old data collected without the new fields; this ensured backwards compatibility for all data collected during the development and refinement of the control system. This feature proved essential for debugging errors and being able to inspect intermediate calculations during an experiment to verify that we had implemented the control algorithms correctly.

The following is a partial list of the data that was recorded during each run; the complete list is viewable in the source code.

System time Units of $0.25\mu\text{s}$. Time elapsed since the beginning of data collection.

Computation time Units of $0.25\mu\text{s}$. Time elapsed from the beginning of each 5ms period until data collection and logging are complete. Used to ensure no calculations exceed the loop time.

System state 32-bit unsigned integer whose bits are set high or low depending on the Boolean state of the following: Rear motor enable, steer motor enable, rear motor fault, steer motor fault, sample buffer encode/initialization/overflow error, momentary switch ([subsection 4.3.2](#)), IMU communication error, filesystem error.

Rear wheel and steer angles and rates Units of rad, rad/s. Rear wheel angular rate was determined by applying a low pass filter ($G(s) = 20\pi/(s + 20\pi)$) to a 100ms moving average of an ideal derivative ($\dot{\theta}_r \approx \Delta\theta_r/\Delta t$). This effectively reduced the sample rate used for computing the rear wheel rate and hence reduced the effect of discretization

error. The moving average approach for obtaining the rear wheel rate was used because the discretization error of the rear wheel encoder was much higher than that of the steer angle encoder (800 vs. 20000 quadrature counts per revolution). Also, commanded forward speeds (and by extension rear wheel rates) were constants, so the delay associated with a moving average did not present any problems. Steer angular rate was obtained by a low pass filtered ideal derivative ($G(s) = 20\pi s/(s + 20\pi)$).

Commanded rear wheel rate and yaw rate Units of rad/s. By default, these both start at 0 rad/s and change once the `speed` or `yaw_rate` commands are issued. Speed was specified in m/s but converted to rear wheel rate by dividing by rear wheel radius.

Motor torques Units of N m. The torque commanded by rear motor controller and the fork motor controller. When generating the command to the digital motor drives, each signal was divided by the respective motor torque constant to calculate the current command in units of Amperes.

Frame angular velocity, sensor acceleration Units of rad/s, m/s². Both quantities were computed by applying the sensitivities, biases, and direction cosine matrix to map raw measurements to appropriately scaled measurements in the bicycle body-fixed frame. (subsection 4.3.2).

State estimates Units of rad, rad / s. The four states of the linearized bicycle model were estimated at each time step and recorded; these state estimates were used in the feedback control law.

4.5 Feedback control system

Two independent control systems were implemented: A rear wheel rate controller and a yaw rate controller. The `speed` command was used to change the commanded rear wheel rate, while the `yaw_rate` command was used to change the commanded yaw rate.

4.5.1 Rear wheel rate controller

The rear wheel rate controller was a discrete time proportional-integral (PI) controller with conditional integration. If the desired rear wheel torque command exceeded the allowable torque, it was saturated and the integrator state was not updated. This technique prevented integrator

windup and was simple to implement. The control law was

$$e_i = \dot{\theta}_{r,i,\text{commanded}} - \dot{\theta}_{r,i} \quad (4.11)$$

$$\tau_{i,\text{desired}} = K_p e_i + x_{i-1} \quad (4.12)$$

$$x_i = \begin{cases} x_{i-1} + \frac{K_p e_i (t_i - t_{i-1})}{T_i} & \text{if } |\tau_{i,\text{desired}}| \leq \tau_{\max} \\ x_{i-1} & \text{if } |\tau_{i,\text{desired}}| > \tau_{\max} \end{cases} \quad (4.13)$$

$$\tau_i = \text{sat}(\tau_{i,\text{desired}}, \tau_{\max}) \quad (4.14)$$

where x_i , $\tau_{i,\text{desired}}$, and τ_i are the integrator state, the desired rear wheel torque, and the saturated rear wheel torque, respectively, all for time step i . The maximum torque was limited by the maximum current of the digital motor drive, $\tau_{\max} = 158\text{Nm}$ (the max motor current was configured to be 24.0A , and a rear motor torque constant was estimated to be 6.6Nm/A). Through repeated experimentation, we found that a gain of $K_p = 50$ and an integration time constant of $T_i = 2000$ provided sufficiently fast response, no steady state tracking error, and no noticeable oscillatory behavior that can be present when integral action is used.

4.5.2 Yaw rate controller

The yaw rate controller was implemented with an inner loop and an outer loop. The inner loop was comprised of a full state estimator for the four bicycle states using measurements of steer angle δ and lean rate $\dot{\phi}$, and a full state feedback control law which used the state estimates instead of the states. The stabilized inner loop was then inserted into an outer yaw rate control loop which computed an additive control action based upon the error between the commanded yaw rate $\dot{\psi}_c$ and the estimated yaw rate $\dot{\psi}$. The block diagram for the complete system is shown in [Figure 4.8](#). An additive sinusoidal disturbance torque with user selectable amplitude a_d and frequency f (see the `disturb` command in [subsection 4.5.3](#)) was also incorporated into the design; by default $a_d = 0$ so that no disturbance was applied unless requested by the user.

The state feedback gain K was computed by discretizing the plant model and solving the discrete algebraic Riccati equation associated with the cost function

$$J = \sum_{i=1}^{\infty} (x_i^T Q x_i + u_i^T R u_i) \quad (4.15)$$

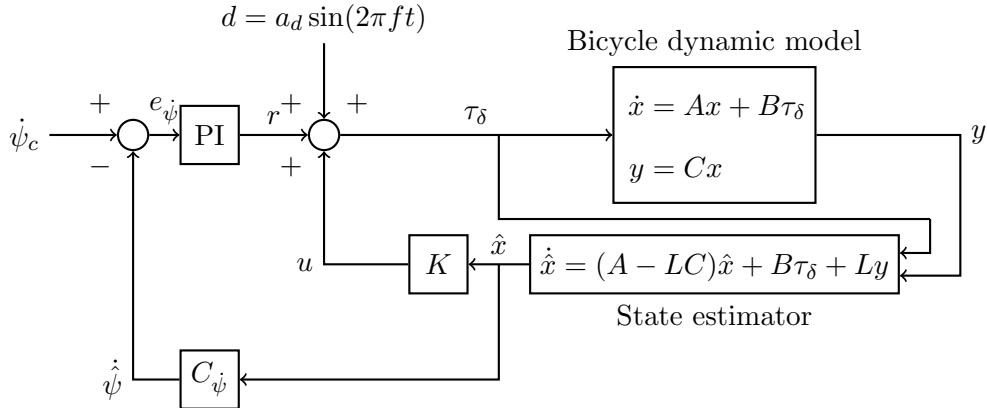


Figure 4.8: Yaw rate control block diagram. The bicycle state is $x = [\phi, \delta, \dot{\phi}, \dot{\delta}]$, \hat{x} is the state estimate, input to the bicycle is steer torque τ_δ , the measurements are $y = [\delta, \dot{\phi}]$.

where Q and R were selected to be

$$Q = \text{diag}\left(\frac{1}{\left(\frac{2\pi}{180}\right)^2}, \frac{1}{\left(\frac{5\pi}{180}\right)^2}, \frac{1}{\left(\frac{2\pi^2}{180}\right)^2}, \frac{1}{\left(\frac{100\pi^2}{180}\right)^2}\right) \quad (4.16)$$

$$R = \frac{1}{0.5^2} \quad (4.17)$$

This choice was selected following Bryson's rule [3]; the terms inside the $(\cdot)^2$'s in the denominator can be viewed as the “maximum allowable” value for the corresponding state or control variable (e.g., the “maximum allowable” value for lean ϕ was $\frac{2\pi}{180} = 2^\circ$).

Once the state feedback gain was computed, the closed loop eigenvalues of the stabilized plant dynamics $A + BK$ were computed, and the estimator gain was selected using pole placement. The poles of the estimator were all placed on the negative real axis, equally spaced by 0.2 rad/s, and with the slowest estimator pole being $3 \min(\text{Re}(\sigma(A+BK)))$. This ensured the convergence of the estimator was substantially faster than the controller dynamics.

With the bicycle dynamics stabilized by the inner loop, an outer loop was added to track yaw rate. Yaw rate was chosen as the variable to track because it is a natural way to describe both straight line motion ($\dot{\psi} = 0$) as well as steady turning motion ($\dot{\psi} = c \neq 0$). The Matlab function `pidtune()` was used to compute the PI gains.

Since the dynamics of the bicycle vary with speed, this control system design was performed at 101 speeds, logarithmically spaced between 0.5m/s and 10m/s. All matrices in Figure 4.8 were output to a C++ file as a sorted array of matrices parameterized by speed. This file was then compiled into the firmware and gain scheduling was used to compute the actual control law based on the measured speed. During a experimental run, the measured speed was used

to perform a binary search into the array to find the matrices associated with speeds bounding the measured speed. A linear interpolation of the state estimate update and control law were performed using these nearest two speeds. An implication of this is that estimation and control was not possible below 0.5m/s. This turned out not to be a problem as long as the bicycle was started from a near the upright reference configuration.

4.5.3 User interface

The following list details the available commands and gives a brief description of each.

collect [filename] Begin main data collection and control loop and store data in optional argument **filename** (if not supplied, data is stored in **samples.dat**).

disable Immediately disable the motors and end data collection.

reset Perform a software reset of the microcontroller.

threads Show the memory address, stack address, priority, number of references to this thread, thread state, thread time in ticks, and the name of all currently running threads.

calibrate Begin the steer angle calibration routine (see [subsection 4.3.2](#))

homefork Wait for the steer encoder index signal to set the steer calibration offset.

e_thresh v_e Set state estimation threshold speed to **v_e** m/s. **v_e** must be greater than or equal to 0.5m/s (the lowest speed for which linearized bicycle dynamics state space matrices were generated).

c_thresh v_c Set the threshold speed to **v_c** m/s. Once the bicycle speed **v** exceeds **v_c** yaw rate control is enabled. Must be larger than estimation threshold speed.

thresh v_e v_c Simultaneously set state estimation and control threshold speeds.

disturb a_d f Set disturbance amplitude to **a_d** 10^{-2} N m and disturbance frequency to **f** Hz (see [Figure 4.8](#)).

speed v Set commanded speed in m/s. The control system will immediately attempt to track this speed until a fault occurs or the reference is changed by the user. On reset, the reference speed is set 0 m/s.

speed_limit v d Set commanded speed to v m/s and distance limit to d meters. This works the same as **speed** except that once the bicycle has travelled d meters, the commanded speed is changed to 1.0 m/s, a speed which was easy to walk next to and catch the bicycle.

l_thresh t Set the lateral acceleration threshold to t . At the beginning of each run, a green LED is illuminated when the bicycle lateral acceleration is below t (with a default of $t=0.01$ m/s²). This enables the bicycle to be initialized with as close to zero lean angle as possible. The sensed lateral acceleration is non-zero in static conditions because the accelerometer senses the gravitational field.

yaw_rate yr Set the commanded yaw rate in rad/s. On reset, reference yaw rate is 0 rad/s (straight line motion).

4.6 Experiments

On July 13, 2013, twenty eight experimental runs were performed on the Dairy Road basketball courts of the UC Davis campus (38.53794° North, 121.759475° West). Oliver Lee and Dr. Mont Hubbard were stationed at the northeast corner of the east court, while I travelled east and west with the robot bicycle each run. The bicycle can be seen balancing in [Figure 4.9](#). To begin each run, Oliver Lee operated the laptop and issued commands to configure the filename (following a naming convention <run number>.dat), speed set point, distance to travel before reducing the speed set point to 1.0 m s⁻¹, and optionally, an additive sinusoidal disturbance steer torque of magnitude a_d (10⁻² N m) and frequency f_d (Hz). Dr. Hubbard took notes about each run and coordinated with Oliver Lee to ensure commands issued to robot were consistent with his notes. Prior to issuing a non-zero speed set point command, I held the bicycle near the reference configuration (as indicated by two LED's which turned on when IMU lateral acceleration and steer angle were less than 0.01 m s⁻² and 1.0 deg, respectively), then ran along side the bicycle as it travelled east to west or west to east. At the end of each run, I would turn off the main power switches to each motor and catch the bicycle to prevent it from falling on its casters. The speed command issued by Oliver took two arguments: speed set point and distance to travel before changing the set point to 1.0 m s⁻¹. At 1.0 m s⁻¹, it was easy to manually switch off the motors and catch the bicycle.



Figure 4.9: Taking the robot bicycle on a jog.

Table 4.3: Runs 000–010 utilized a yaw rate PI controller which explicitly specified the first 0db cross-over frequency to be 0.1 Hz and utilized a disturbance torque $d = a_d \sin(2\pi t)$. The firmware was revised in Runs 011–014 to utilize a disturbance torque $d = a_d \sin(2\pi(t - t_i))$ to ensure the initial disturbance began at 0 N m. For runs 015–028, the firmware was updated to use a yaw rate PI controller which did not explicitly specify the crossover frequency (instead it was selected automatically by Matlab based on the plant dynamics).

Run	Time (PST)	Speed (m s^{-1})	a_d (10^{-2} N m)	f_d (Hz)	x (m)	Direction
000	0641	2.0	—	—	30.0	West
001	0643	2.0	—	—	30.0	West
002	0644	3.0	—	—	30.0	East
003	0646	3.0	—	—	30.0	West
004	0647	4.0	—	—	30.0	East
005	0649	4.0	—	—	30.0	West
006	0650	4.0	—	—	60.0	East
007	0652	4.0	—	—	60.0	East

Table 4.3: (continued)

Run	Time (PST)	Speed (m s^{-1})	a_d (10^{-2} N m)	f_d (Hz)	x (m)	Direction
008	0656	2.0	0.5	1.0	30.0	West
009	0659	2.0	1.0	1.0	30.0	East
010	0701	2.0	1.0	1.0	30.0	East
011	0714	2.0	1.0	1.0	30.0	West
012	0718	2.0	5.0	1.0	30.0	East
013	0721	2.0	5.0	1.0	30.0	West
014	0725	2.0	-5.0	1.0	30.0	East
015	0754	2.0	-	-	30.0	West
016	0756	2.0	-	-	30.0	East
017	0757	4.0	-	-	30.0	West
018	0800	2.0	5.0	1.0	30.0	East
019	0808	2.0	5.0	5.0	30.0	West
020	0813	2.0	10.0	1.0	30.0	West
021	0815	2.0	-	-	30.0	West
022	0816	1.0	-	-	30.0	East
023	0818	1.0	-	-	30.0	East
024	0821	2.0	-	-	5.0	East
025	0824	2.0	-	-	60.0	West
026	0825	3.0	-	-	50.0	East
027	0827	3.0	-	-	50.0	East
028	0829	3.0	-	-	50.0	East

4.6.1 Example results

The speed and steer angle response to a step in commanded speed is shown in [Figure 4.10](#). Prior to the step, the bicycle was held in a nearly stationary upright zero steer configuration. During the transition from 0 m s^{-1} the bicycle was manually guided until a speed of approximately 1.0 m s^{-1} was reached, at which point the bicycle was left to accelerate further and balance

solely via the yaw rate controller. The forward speed was achieved only by the action of the rear wheel motor, no manual pushing of the bicycle was performed. This speed response was typical of all runs.

As described in [subsection 4.5.2](#), controller gains were computed for forward speeds in the range of $0.5\text{--}10\text{ m s}^{-1}$. However, due to the difficulty of balancing at low speeds, we found that enabling the yaw rate control system at speeds below 1.0 m s^{-1} proved problematic and would result in drastic steer torques when accelerating from 0 m s^{-1} to 1.0 m s^{-1} . For this reason, the threshold speed for state estimation was set to 1.0 m s^{-1} , effectively disabling any yaw rate control (i.e., $\tau_\delta = 0$) until the forward speed exceeded 1.0 m s^{-1} . This threshold was a one time trigger – once this speed was reached, the controller remained enabled even if the speed went below 1.0 m s^{-1} (for example, if the commanded speed was less than 1.0 m s^{-1}).

Once the forward speed exceeded the estimation threshold speed, state estimation and control began. This event is visible in the steer angle estimate  of [Figure 4.10](#) at the time when the measured speed  crosses 1.0 m s^{-1} – the steer angle estimate initial condition is set to the steer angle measurement at that time step. Similarly, the other estimator initial conditions were populated using the most recently available measurements, lean rate $\dot{\phi}$ (from rate gyroscope), and steer rate $\dot{\delta}$ (from a low pass filtered numerical time derivative of steer). No direct measurement of lean ϕ was available, so the initial lean estimate $\hat{\phi}$ was zero for each run. This latter initial condition was the motivation for ensuring that the bicycle was as close to $\phi = 0$ at the beginning of each run – doing so minimized the initial estimation error and helped ensure a smooth start.

There is a noticeable difference between the measured steer  and estimated steer . This estimation error can be attributed several things: modelling errors, measurement errors, and gain and phase response of the state estimator. Modelling errors arise from inaccuracies of the measured physical parameters or unaccounted-for dynamics. Measurement errors arise from discretization error, imprecisely known sensor sensitivities and biases, and to a lesser degree, unmodelled sensors dynamics (the frequencies of interest in the bicycle are 2-3 orders of magnitude smaller than the natural frequencies of typical MEMS rate gyroscopes). These modelling errors result in a steady state estimation error, similar to a pure proportional controller without any integral action. Finally, the state estimate is attenuated and delayed by some amount from the estimator inputs (steer torque τ_δ and measurements of steer δ and lean rate $\dot{\phi}$); this is a trade-off that cannot be avoided without introducing undesired amplification of high frequency

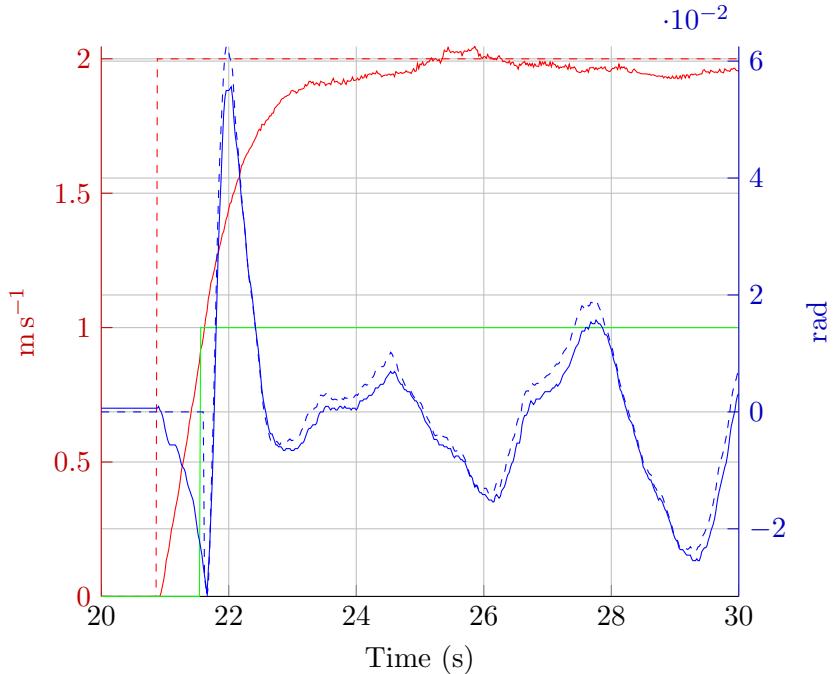


Figure 4.10: Run 000, measured speed — $v = -\dot{\theta}_r R_r$, commanded speed - - - v_c , measured steer angle — δ , and estimated steer angle - - - $\hat{\delta}$. The momentary switch state — — — indicates the bicycle was released at $t = 21.56$ s, shortly before state estimation began at $t = 21.62$ s ($v = 1.0 \text{ m s}^{-1}$). See 4.3.2 for a description of this switch.

sensor noise.

Estimates of steer rate $\dot{\delta}$ and lean rate $\dot{\phi}$ are shown in Figure 4.11, Figure 4.12, respectively. In comparison to the steer and steer estimate signals, these signals are much noisier. Visible in both is the sudden change in the estimator state (at $t = 21.62$ s) when the estimation threshold speed is reached. The time response of rear wheel torque and the steer torque τ_δ is shown in Figure 4.13.

4.6.2 Discussion

The transfer function from the input steer torque τ_δ to an output measurement (i.e., steer δ or lean rate $\dot{\phi}$) can be estimated by computing the ratio of the forward discrete Fourier transform (DFT) of the selected output to the forward DFT of the input steer torque. Because the dynamics are speed dependent, the selected input and output should be time series data during which the speed is nearly constant. For n real data points $x_0 \dots x_{n-1}$, the forward DFT is given by

$$X_k = \sum_{j=0}^{n-1} x_j e^{-2\pi j k \sqrt{-1}/n} \quad k = 0 \dots n-1 \quad (4.18)$$

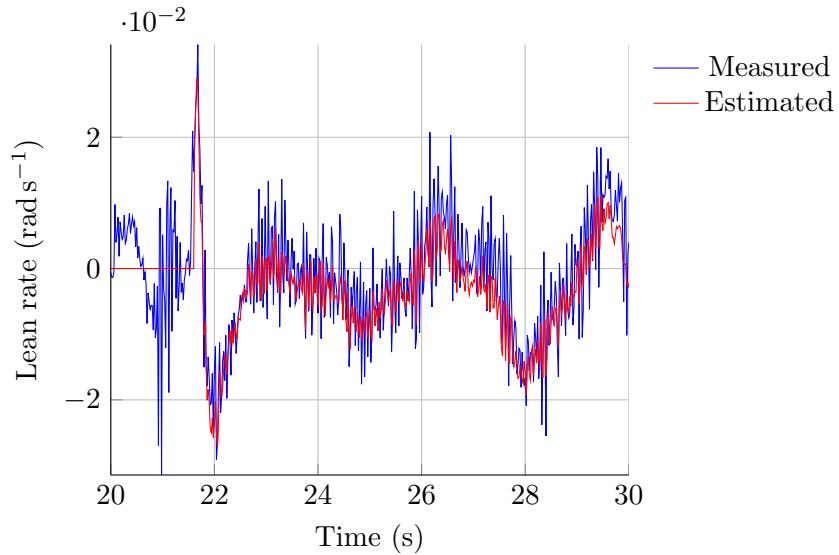


Figure 4.11: Run 000 lean rate measured and estimated. The lean rate measurement was obtained by transforming rate gyroscope measurements from the sensor fixed axes to the lean frame fixed axes (constant frame pitch θ was assumed).

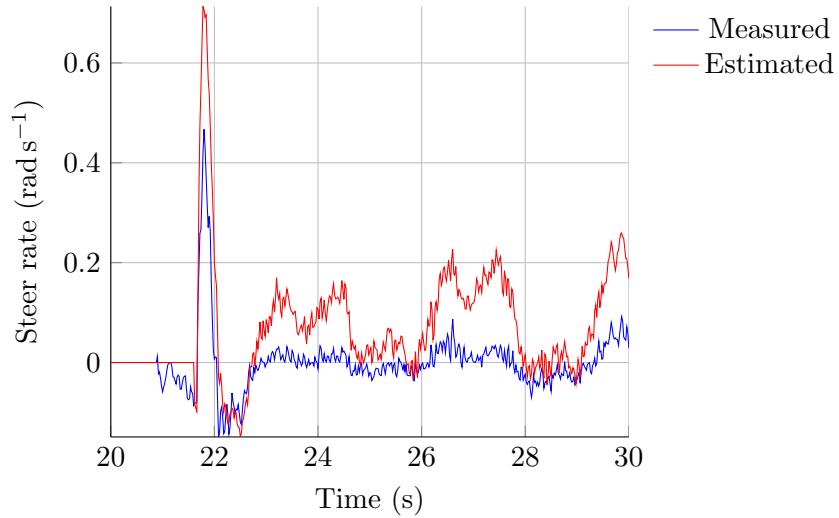


Figure 4.12: Run 000 steer rate measured and estimated. Measured steer rate was the result of applying a low pass filtered ideal derivative $G(s) = 20\pi s/(s + 20\pi)$ to the steer angle measurement. The noise in this signal is directly proportional to steer angle quantization error and inversely proportional to sample period.

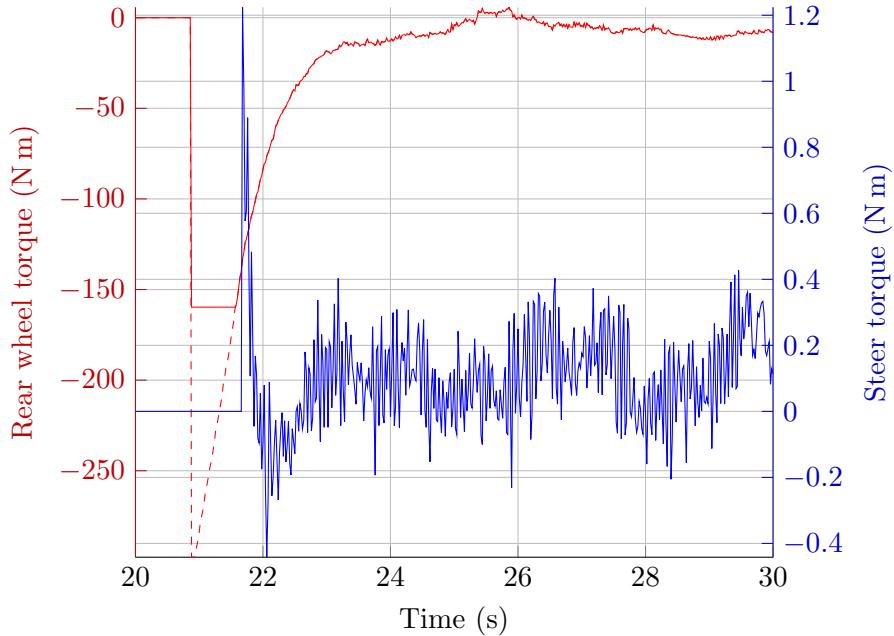


Figure 4.13: Run 000 rear wheel and steer torque response. Rear wheel torque — and desired rear wheel torque - - - (saturation occurred for $t \in [20.88 \text{ s}, 21.57 \text{ s}]$). Steer torque — did not saturate. The effect of the positive initial spike in steer torque is visible in [Figure 4.10](#), [Figure 4.11](#), and [Figure 4.12](#).

The sequence X_k possesses Hermitian symmetry ($X_k = X_{n-k}^*$), so half of the output is redundant (the second half is the complex conjugate of the first half, in reverse order). Therefore, the result of the DFT of a sequence of n real numbers is a sequence of $\frac{n}{2} + 1$ complex numbers, where $\frac{n}{2}$ is rounded down if n is odd. The first element of this output sequence, X_0 is always real and represents n times the mean of the input sequence x_k (i.e., $X_0 = nE\{x_k\}$). The magnitude of each of the remaining $\frac{n}{2}$ complex numbers is $\frac{1}{2}$ the magnitude of the frequency response, at frequencies evenly spaced between 0 and the Nyquist frequency $\frac{f_s}{2}$. In practice, to obtain the magnitude portion of the frequency response, one needs to compute $\frac{2}{n}|X_k|$ for $k = 1 \dots n/2$, and $\frac{1}{n}|X_0|$ for the DC component. However, to estimate a transfer function given the DFT of the input and output of an assumed linear system, only the ratio of the DFTs need to be considered, so this scaling is unnecessary.

Estimates of $|\frac{\delta}{\tau_\delta}(j\omega)|$ and $|\frac{\dot{\phi}}{\tau_\delta}(j\omega)|$ were computed for three different commanded speeds: 1, 2, and 4 m s^{-1} . The time series data was selected over a time interval during which the forward speed of the bicycle was nearly constant. Since the sampling frequency was 200 Hz, all frequency responses are plotted up to 100 Hz (the Nyquist frequency).

Control of a bicycle through steer torque alone is more difficult at low speeds than at high

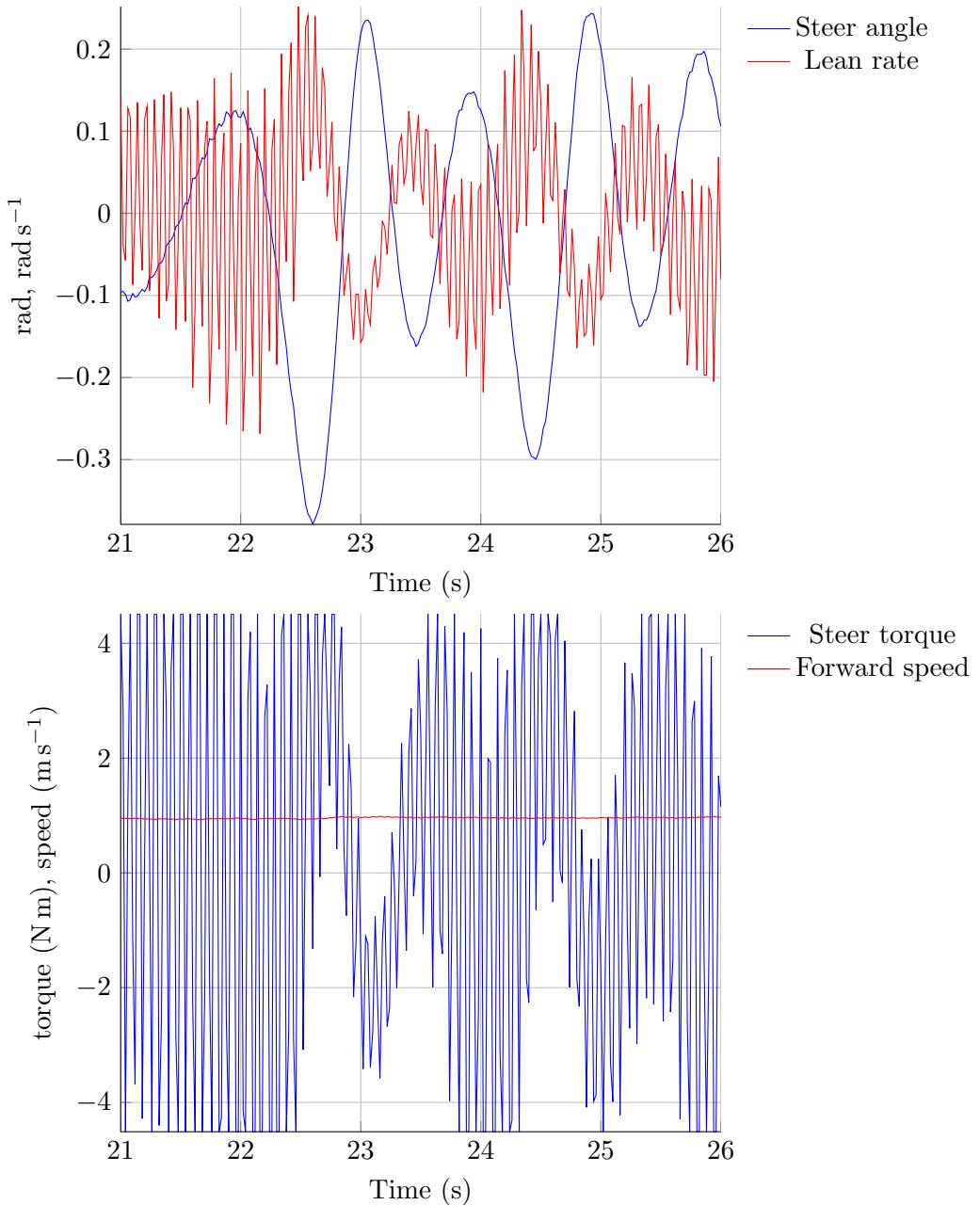


Figure 4.14: Run 024 steer, lean rate, steer torque and speed response, $v_c = 1.0 \text{ m s}^{-1}$. The run continued until $t = 41 \text{ s}$ with very similar response. Steer torque saturated repeatedly at $\tau_{\delta,\max} = \pm 4.511 \text{ N m}$.

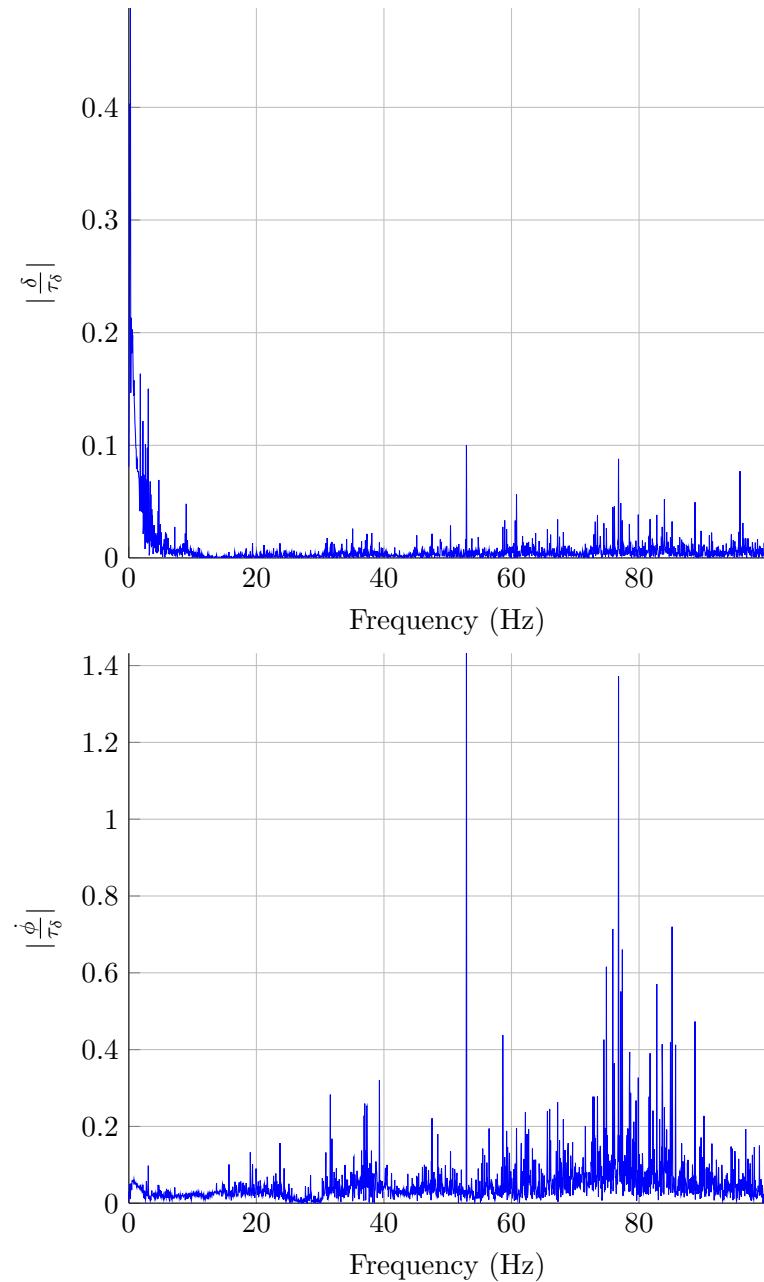


Figure 4.15: Run 024 steer and lean rate frequency response, $v_c = 1.0 \text{ m s}^{-1}$. DFT of steer, lean rate, and steer torque signals was computed for $t \in [21 \text{ s}, 41 \text{ s}]$ (4001 time points).

speeds. This fact manifests itself in the control system as large state feedback gains which increase noise sensitivity. This effect is clearly apparent in [Figure 4.14](#) where high frequency, large magnitude (saturating) steer torque oscillations were observed. The frequency response in [Figure 4.15](#) exhibits sharp peaks at 53 Hz and 77 Hz and generally has much more high frequency content than runs at higher speeds. This indicates that the applied steer torque excited dynamics that are not present in the Whipple bicycle model. The $\frac{\delta}{\tau_\delta}$ transfer function exhibits stronger peaks in the low frequency range than the $\frac{\dot{\phi}}{\tau_\delta}$ transfer function; this may be due to the $\frac{1}{s}$ -like dynamics (i.e., high gain at low frequencies, -20 dB/decade decay at high frequencies) associated with this state being an angle as opposed to a rate.

For a speed command of $v_c = 2.0 \text{ m s}^{-1}$, the steer torque response is much less noisy and does not saturate. There is a clear DC offset visible in the steer torque in [Figure 4.16](#). Despite this, the bicycle travelled nearly in a straight line for approximately 60 m, indicating that the yaw rate controller was driving the yaw rate $\dot{\psi}$ to zero (a straight line). The steer torque bias was likely due to mass imbalance or frame misalignment. Possible sources of mass imbalance include the rear hub motor (it unfortunately was not laterally symmetric, even though the rim and tire were, see [Figure 4.3](#)), the electrical hardware on the frame sheet, the batteries, and the training wheels. Possible sources of frame misalignment include the wheel planes not being coplanar when steer $\delta = 0$ as a result of frame and fork construction. The steer magnitude frequency response $|\frac{\delta}{\tau_\delta}|$ at $v_c = 2.0 \text{ m s}^{-1}$ exhibits similar low frequency characteristics as in the case of $v_c = 1.0 \text{ m s}^{-1}$. However, $|\frac{\delta}{\tau_\delta}|$ is drastically different; this is likely due to the steer torque bias that is more apparent when the steer torque oscillations are less extreme as in the case when $v_c = 1.0 \text{ m s}^{-1}$.

[Table 4.4](#) indicates that the higher speeds result in smaller mean and standard deviation of steer angle. This can be attributed to the fact that the bicycle is closer to the stable speed range where no control action is necessary to balance. The lean rate standard deviation $\sigma_{\dot{\phi}}$ is also substantially decreased as speed increases. The steer torque mean μ_{τ_δ} does not appear to be speed dependent and indicates that mass imbalance is more likely the cause than geometric frame misalignment.

4.6.3 Conclusions

The robot bicycle has a robust control and data acquisition system that can be the basis for more experiments. While the bicycle does balance and record data reliably across a moderate

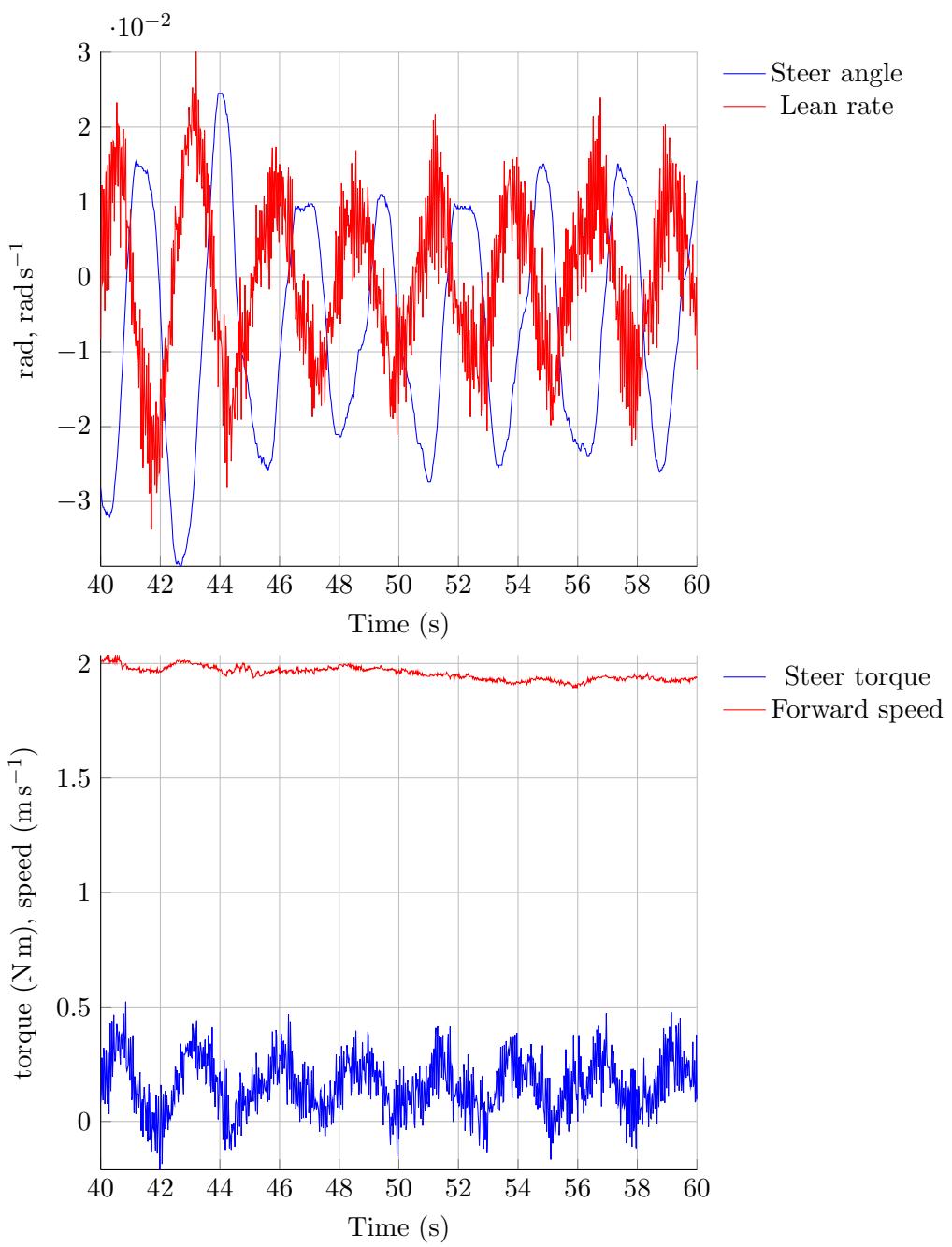


Figure 4.16: Run 025 steer, lean rate, steer torque and speed response, $v_c = 2.0 \text{ m s}^{-1}$.

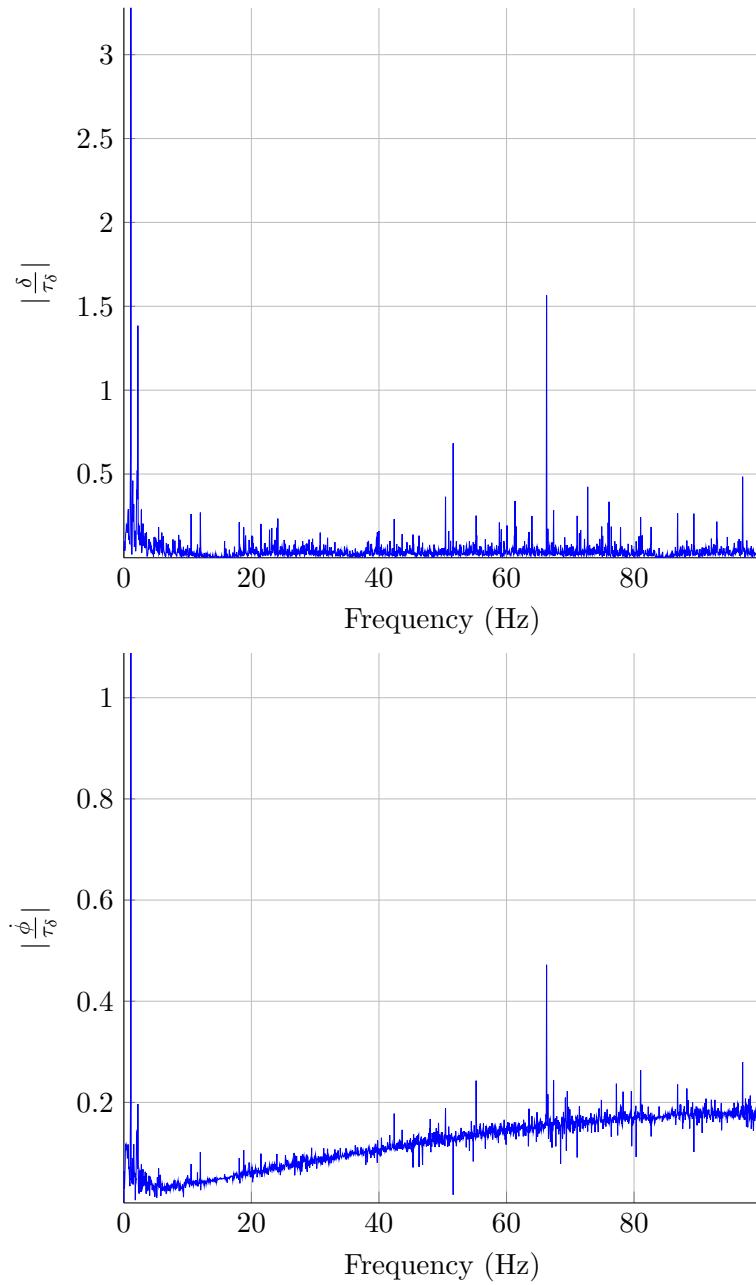


Figure 4.17: Run 025 steer and lean rate frequency response, $v_c = 2.0 \text{ m s}^{-1}$. DFT of steer, lean rate, and steer torque signals was computed for $t \in [40 \text{ s}, 60 \text{ s}]$ (4001 time points).

Run	$v_c \text{ m s}^{-1}$	$\mu_v \pm \sigma_v \text{ m s}^{-1}$	$\mu_\delta \pm \sigma_\delta \text{ rad}$	$\mu_{\dot{\phi}} \pm \sigma_{\dot{\phi}} \text{ rad s}^{-1}$	$\mu_{\tau_\delta} \pm \sigma_{\tau_\delta} \text{ N m}$
024	1.0	0.96 ± 0.02	-0.0128 ± 0.1559	0.0007 ± 0.1045	0.16 ± 3.20
025	2.0	1.96 ± 0.03	-0.0070 ± 0.0153	-0.0002 ± 0.0108	0.16 ± 0.13
007	4.0	3.95 ± 0.02	-0.0034 ± 0.0029	-0.0025 ± 0.0093	0.15 ± 0.12

Table 4.4: Mean and standard deviation of speed, steer, lean rate, and steer torque.

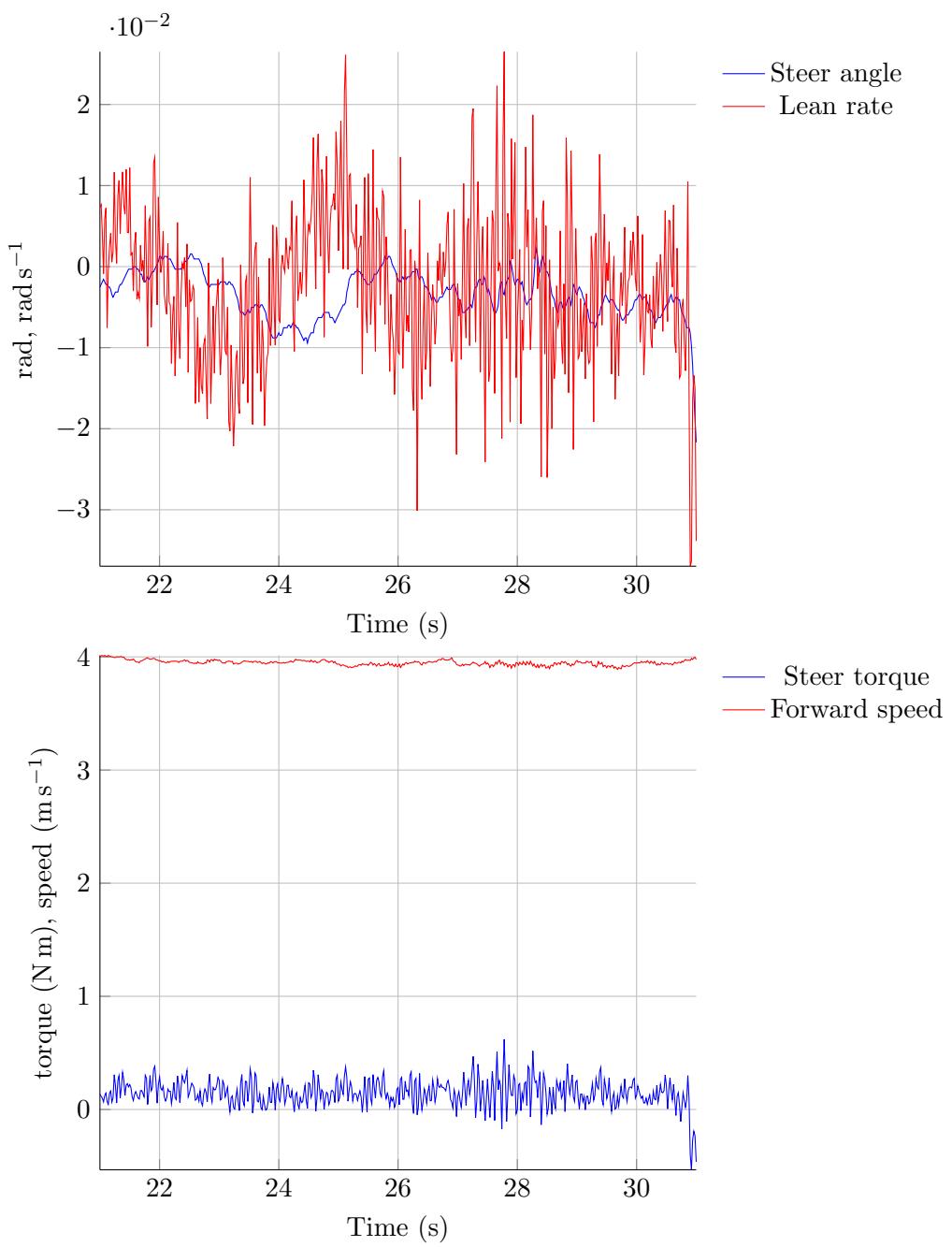


Figure 4.18: Run 007 steer, lean rate, steer torque and speed response, $v_c = 4.0 \text{ m s}^{-1}$.

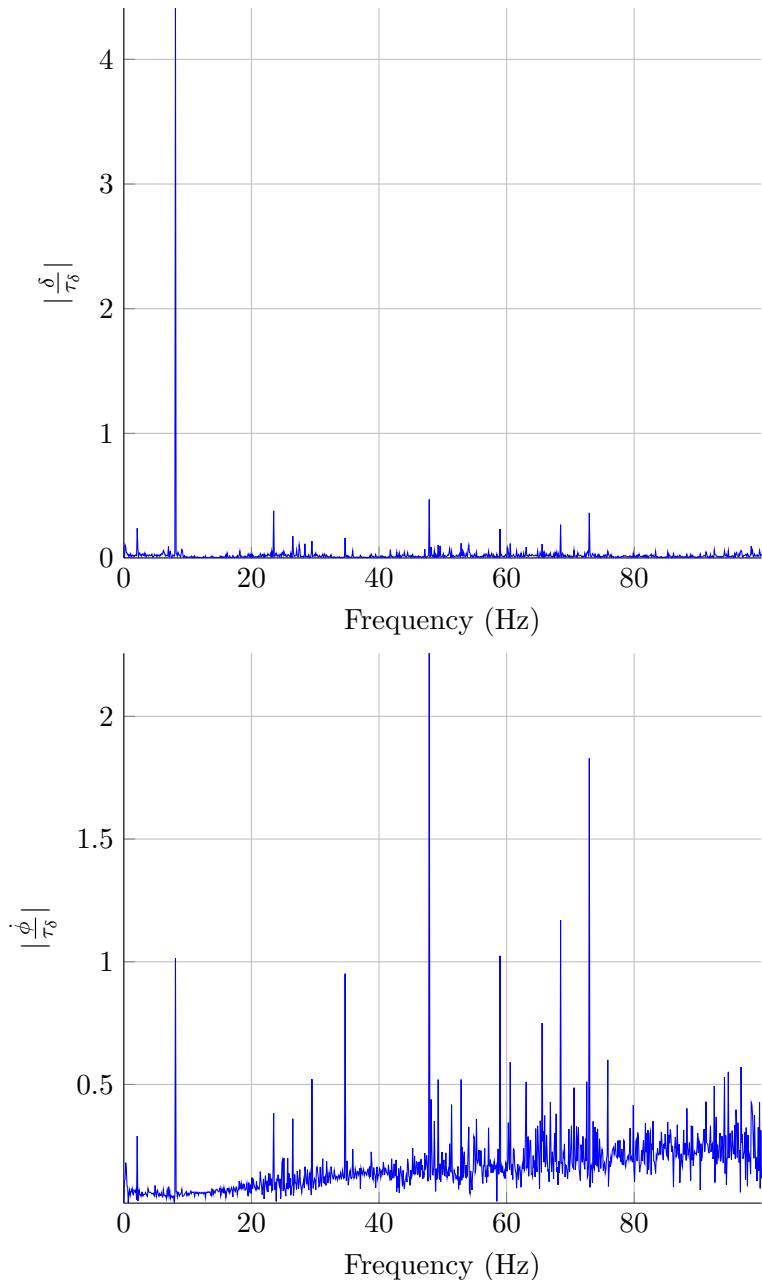


Figure 4.19: Run 007 steer and lean rate frequency response, $v_c = 4.0 \text{ m s}^{-1}$. DFT of steer, lean rate, and steer torque signals was computed for $t \in [21 \text{ s}, 31 \text{ s}]$ (2001 time points).

range of speeds, there are a number of issues that need to be addressed in order for quantitative system identification techniques to be applied with any reasonable hope of success. Of primary concern is the mass imbalance that is apparent by the steer torque bias. Additionally, the rear wheel encoder was relatively low resolution (800 counts per wheel revolution), and this directly affected the balance control because speed measurement noise (due to discretization error) directly impacted the gain scheduling algorithm used for the control of yaw rate and the lateral dynamics. It is possible that some of the noise in the steer torque signal was induced by the gain scheduling. Replacing the rear encoder with a higher resolution version would directly address this issue.

Another implementation issue is the full state estimator. Especially in the case of steer angle, this estimator was redundant. The steer rate estimate was also relatively poor in comparison to a simple low pass filtered derivative of the steer angle. Implementing a reduced order estimator to estimate only lean angle and lean rate, the performance (i.e., bandwidth and accuracy) of the estimator would be improved and likely contribute to overall better performance of the state feedback law.

Finally, it should be mentioned that the conditions of the experiment were not as well controlled as they ideally should have been. The experiments were conducted early in the morning with very little wind, but ideally they would have been conducted indoors on a smoother surface than the surface provided by the outdoor basketball courts. Unfortunately, due to time limitations and scheduling conflicts, we were unable to conduct experiments indoors at the UC Davis Activities and Recreation Center (ARC).

Qualitatively, the frequency response obtained in the experiments indicates the presence of dynamics besides those predicted by the Whipple model. However, the fact that the Whipple model was successfully used as the plant model for the control system does indicate that at least for low bandwidth tasks, the model can be used to reasonably describe the behavior of real bicycles. However, the robustness of the control system was not rigorously tested through disturbance inputs, and it may very well be that relatively minor disturbances would destabilize the bicycle. To know this for sure, refinement of the experimental conditions and hardware are needed. It is my sincere hope that the project can be continued and taken further now that much of the groundwork is in place.

References for Chapter 4

- [1] Amped Bikes. "Brute" Direct Drive Rear Hub Motor. URL: <http://www.ampedbikes.com/description.html> (visited on 08/11/2013) (cited on page 64).
- [2] ARM Holdings plc. *GNU Tools for ARM Embedded Processors*. 2013. URL: <https://launchpad.net/gcc-arm-embedded> (visited on 08/10/2013) (cited on page 57).
- [3] Arthur E. Bryson Jr. and Yu-Chi Ho. *Applied Optimal Control*. Cambridge, MA, USA: Hemisphere Publishing Corporation, 1975. ISBN: 0-89116-228-3 (cited on page 69).
- [4] Michael J. Chudobiak. *moserial*. 2013. URL: <https://wiki.gnome.org/moserial%20http://git.gnome.org/cgit/moserial> (cited on page 65).
- [5] Copley Controls – Analogic Motion Technology. *Copley Controls ACJ-055-18*. URL: <http://www.copleycontrols.com/Motion/Products/Drives/Digital/accelnet.html> (visited on 08/11/2013) (cited on page 63).
- [6] Copley Controls – Analogic Motion Technology. *Copley Controls ADP-090-36*. URL: <http://www.copleycontrols.com/Motion/Products/Drives/Digital/accelnet.html> (visited on 08/11/2013) (cited on page 63).
- [7] Digi International Incorporated. *XBee-PRO 802.15.4 extended-range module w/ RPSMA connector (Part Number: XBP24-ASI-001)*. 2013. URL: <https://www.digi.com/products/model?mid=3270> (visited on 08/12/2013) (cited on page 65).
- [8] Giovanni Disirio. *ChibiOS/RT free embedded RTOS*. 2013. URL: <http://www.chibios.org/> (visited on 08/10/2013) (cited on page 57).
- [9] Google. *Google Protocol Buffers*. 2013. URL: <https://developers.google.com/protocol-buffers/> (cited on page 66).
- [10] HobbyKing. *ZIPPY Flightmax 5000mAh 2S1P 20C*. URL: http://www.hobbyking.com/hobbyking/store/_8578_zippy_flightmax_5000mah_2s1p_20c.html (visited on 08/11/2013) (cited on page 65).
- [11] International Standards Organization. *ISO/IEC 14882:2011 – Information technology – Programming languages – C++*. 2011. URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50372 (visited on 08/10/2013) (cited on page 57).

- [12] Invensense. *Invensense MPU-6050*. URL: <http://www.invensense.com/mems/gyro/mpu6050.html> (visited on 08/10/2013) (cited on page 59).
- [13] Jodi D. G. Kooijman, Arend L. Schwab, and Jaap P. Meijaard. “Experimental validation of a model of an uncontrolled bicycle”. In: *Multibody System Dynamics* 19.1-2 (May 2007), pages 115–132. ISSN: 1384-5640. DOI: [10.1007/s11044-007-9050-x](https://doi.org/10.1007/s11044-007-9050-x). URL: <http://link.springer.com/10.1007/s11044-007-9050-x> (cited on page 50).
- [14] Olimex. *Olimex STM32-H407 Development Board*. URL: <https://www.olimex.com/Products/ARM/ST/STM32-H407/> (visited on 08/10/2013) (cited on page 57).
- [15] Surly. *Surly 1x1, Extra Large*. 2009. URL: http://surlybikes.com/uploads/downloads/Surly_Cat_2009_web.pdf (visited on 08/13/2013) (cited on page 51).
- [16] Teknic. *Teknic M-3441E-LN-20D*. URL: http://teknic.com/files/product_info/N34_Industrial_Grade_Motors_v5.2.pdf (visited on 08/10/2013) (cited on pages 58, 64).
- [17] US Digital. *US Digital H5-50-NE-D*. URL: <http://usdigital.com/products/encoders/incremental/rotary/shaft/H5> (visited on 08/10/2013) (cited on page 58).

Chapter 5

Conclusions and suggestions for future work

The nonlinear equations of motion for the bicycle model described in [Chapter 2](#) were formulated using bicycle gyrostat parameters and subsequently linearized using the linearization procedure described in [Chapter 3](#). These linearized equations were found to match previously published eigenvalues for a set of benchmark parameters, thereby establishing that no mistakes were made in the derivation. However, this does not establish the soundness of the modelling assumptions. The use of the linearized dynamic equations to design a control system that balances the bicycle both in simulation and in practice does however, to some degree, establish that the Whipple bicycle model is at least descriptive for a control system based upon its assumptions to keep an unmanned bicycle from falling over. To what degree the Whipple model is accurate and how exactly to quantify the degree to which it is accurate was not concretely established by this work.

There are several areas where this issue can be addressed. A more careful measurement of all of the physical parameters of the bicycle would be an inexpensive way to improve the knowledge of the assumed plant. At the same time, issues such as inertial asymmetries of the four rigid bodies in the real bicycle should be either eliminated or, if it is not possible to eliminate entirely, quantified. This may necessitate the need to reformulate the model of [Chapter 2](#) to include inertial asymmetries. A change to the model such as allowing for the mass center of the frame and fork to lie outside the plane of the wheel, would be a simple addition, and similar such modifications could be added if necessary. Other improvements in the design of the state estimator (i.e., a reduced order observer) would also be worth testing to see if they improve the performance characteristics of the controller. A final high value, low cost, would be to use a higher resolution optical encoder (a drop in replacement encoder exists that would yield 20000

quadrature counts per wheel revolution). This improvement would reduce discretization jitter in the speed measurement which in turn would reduce jitter in the gain scheduling lookups. It is unclear whether this concern is actually justified, but nevertheless, it is a simple and cheap fix.

The goal of applying additive sinusoidal disturbance steer torques as a means to excite specific frequencies did not work as well as planned. One problem was that a zero mean disturbance steer torque caused a non-zero mean yaw rate, despite the yaw rate command of zero. This may be addressable by refinement of the control system and how the disturbance signal is applied.

Commanding circular motions (i.e., $\dot{\psi}_c \neq 0$) of the robotic bicycle was not attempted due to lack of space and time. The theory of steady turning bicycles is less developed than that of bicycles travelling in a straight line, so there is substantial room for investigations of steady turns, both in terms of numerical studies of the model, as well as experimental validations of the model in operating conditions other than upright steady forward cruise.

With the improvements mentioned, more rigorous system identification experiments would be possible and the validity of the inadequacies of the Whipple model could be more precisely quantified. It is my suspicion that the lack of a tire model is likely the first place to look when making improvements to the model, though there may be other simple additions to the model, such as using the torus model of the wheel instead of the knife edged model that could potentially improve model fidelity. Simple models of tires do exist and can be added to the bicycle model easily to determine how far a simple tire model can be taken. If more sophisticated tire models prove necessary, measurement of tire viscoelastic properties would be required.

This dissertation, as with any work, is never fully “finished”. The L^AT_EX source code, scripts, images, and data used for generating figures is available online [1]. Data collected from the robotic bicycle and source code used in the design and implementation of the control system is available as electronic supplementary material to the dissertation through ProQuest.

References for Chapter 5

- [1] Dale Lukas Peterson. “Bicycle dynamics: modelling and experimental validation”. Ph.D. University of California Davis, 2013. URL: <https://github.com/hazelnusse/dissertation> (cited on page 89).