Path Following and Stabilization of an Autonomous Bicycle

by

David Bickford

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In this thesis we investigate the problem of designing a control system for a modern bicycle so that the bicycle is stable and follows a path. We propose a multi-loop control architecture, where each loop is systematically designed using linear control techniques. The proposed strategy guarantees that the bicycle asymptotically converges to paths of constant curvature. A key advantage of our approach is that by using linear techniques analysis and controller design are relatively simple.

We base our control design on the nonlinear (corrected) Whipple model, which has been previously verified for correctness and experimentally validated. The equations of motion for the nonlinear model are very complicated, and would take many pages to explicitly state. They also have no known closed form solution. To enable analysis of the model we linearize it about a trajectory such that the bicycle is upright and travelling straight ahead. This linearization allows us to arrive at a parameterized linear time-invariant state-space representation of the bicycle dynamics, suitable for analysis and control design.

The inner-loop control consists of a forward-speed controller as well as a lean and steer controller. To keep the bicycle at a constant forward speed, we develop a high-bandwidth proportional controller that uses a torque along the axis of the rear wheel of the bicycle to keep the angular velocity of the rear wheel at a constant setpoint. To stabilize the bicycle at this forward speed, lean torque and steer torque are treated as the control signals. We design a state-feedback controller and augment integrators to the output feedback of the lean angle and steer angle to provide perfect steady-state tracking. To arrive at the gains for state feedback, linear-quadratic regulator methods are used.

When following a constant-curvature path, a vehicle has a constant yaw rate. Using this knowledge, we begin designing the outer-loop path-following control by finding a map that converts a yaw rate into appropriate lean angle and steer angle references for the inner loop. After the map is completed, system identification is performed by applying a yaw-rate reference to the map and analyzing the response of the bicycle. Using the linear approximation obtained, a classical feedback controller for yaw-rate tracking is designed. In addition to yaw-rate control, to track a path the yaw angle of the bicycle must match that of the path and the bicycle must physically be on the path. To analyze these conditions a linear approximation for the distance between the bicycle to the path is found, enabling construction of a linear approximation of the entire system. We then find that by passing the signal for the difference in yaw rate and the distance through separate controllers, summing their output, and subtracting from the reference yaw rate of the path, the bicycle converges to the path.

After developing the general design procedure, the final part of the thesis shows a step by step design example and demonstrates the results of applying the proposed control architecture to the nonlinear bicycle model. We highlight some problems that can arise when the bicycle is started far from the path. To overcome these problems we develop the concept of a virtual path, which is a path that when followed returns the bicycle to the actual path. We also recognize that, in practice, typical paths do not have constant curvature, so we construct more practical paths by joining straight line segments and circular arc segments, representing a practical path similar to a path that would be encountered when biking through a series of rural roads. Finally, we finish the design example by demonstrating the performance of the control architecture on such a path.

From these simulations we show that using the suggested controller design that the bicycle will converge to a constant curvature path. Additionally with using the controllers we develop that in the absence of disturbance the bicycle will stay within the intended traffic lane when travelling on a typical rural road.

Acknowledgements

I would like to thank my supervisor, Professor Daniel Davison, for affording me the opportunity to return to the University of Waterloo and pursue research in bicycle control as part of the Systems and Control Group. The time he has spent providing support and guidance has been invaluable during this entire process. I would also like to thank my readers, Professors Christopher Nielsen and Ehsan Toyserkani, for their comments on this thesis.

Finally I would like to thank my family and friends, especially my parents, Lennard and Colleen Bickford, for their support and encouragement.

Table of Contents

List of Tables

List of Figures

Nomenclature

$\mathbf{a}$ Unknown vector of constants when splitting γ^{mod} into known and unknown parts in system identification.

a_n, ..., a_0 Coefficients of s^n, ..., s^0 in the denominator of $P_{\dot{\gamma}}(s)$.

$\bar{\mathbf{A}}$ Parameterized state matrix for the linearized lean and steer dynamics of the bicycle augmented with integrator states.

$\mathbf{A}(v)$ Parameterized state matrix for the linearized lean and steer dynamics of the bicycle.

b_m, ..., b_0 Coefficients of s^m, ..., s^0 in the numerator of $P_{\dot{\gamma}}(s)$.

B Point of contact between the bicycle rear wheel and the ground.

$\mathbf{B}$ Input matrix for the linearized lean and steer dynamics of the bicycle.

$\bar{\mathbf{B}}$ Input matrix for the linearized lean and steer dynamics of the bicycle augmented with integrator states.

$\bar{\mathbf{B}}_r$ Input matrix for the reference trajectory vector for the lean and steer angles in the lean and steer controller.

c_j Line segment connecting the start and end point of the circular arc in the corner of waypoint j.

$\mathbf{c}_{rw}$ Cross product of $\mathbf{n}_P$ and $\hat{e}_3$.

$\mathbf{c}_{fw}$ Cross product of $\mathbf{n}_L$ and $\hat{e}_3$.

$\mathbf{C}$ Output matrix for the linearized lean and steer dynamics of the bicycle.

$\bar{\mathbf{C}}$ Output matrix for the linearized lean and steer dynamics of the bicycle augmented with integrator states.

C_1 Circle in virtual path construction that is positioned such that the rear wheel contact with the ground is on the circle and the bicycle yaw angle forms a line tangent to the circle. If the direction is counter clockwise this circle is denoted $C_{1,CCW}$ and if it is clockwise the circle is denoted $C_{1,CW}$.

C_2 Circle in virtual path construction that is positioned such that the closest point on the path to the bicycle rear wheel contact point and the ground is on the circle and the yaw angle of the path at this point forms a line tangent to the circle. If the direction is counter clockwise this circle is denoted $C_{1,CCW}$ and if it is clockwise the circle is denoted $C_{1,CW}$.

$C_d(s)$ Controller providing corrections based on d_ρ.

$C_\gamma(s)$ Controller providing corrections based on $|\gamma - \lambda|$ ($|\phi + \pi - \lambda|$ in the actual system).

$C_{\dot{\gamma}}(s)$ Classical feedback controller for controlling $\dot{\gamma}$ of the bicycle. ($\dot{\gamma}$ is estimated by $\dot{\phi}$ in the actual implementation.)

d_ρ Distance of the bicycle to the path.

$\dot{d}_\rho$ Change in distance of the bicycle to the path with respect to time.

$D_C(s)$ The denominator of $C_{\dot{\gamma}}(s)$.

$D_P(s)$ The denominator of $P_{\dot{\gamma}}(s)$.

$\mathbf{e}$ Error vector in system identification.

$\hat{e}_1$ Unit vector in the direction of the global Cartesian x axis.

$\hat{e}_2$ Unit vector in the direction of the global Cartesian y axis.

$\hat{e}_3$ Unit vector in the direction of the global Cartesian z axis.

f_{n-1}, ..., f_0 Coefficients of z^{n-1}, ... z^0 in the denominator of $P_{\dot{\gamma}}[z]$.

F Point of contact between the bicycle front wheel and the ground.

$\mathbf{F}$ State-feedback gains for the lean and steer controller.

$\mathbf{F}_1$ State-feedback gains for $\boldsymbol{\chi}$ in the lean and steer controller.

$\mathbf{F}_2$ State-feedback gains for $\boldsymbol{\zeta}$ in the lean and steer controller.

$F_{d,I}$ Gain of the integrator term in $C_d(s)$ in the design example.

$F_{d,P}$ Gain of the proportional term in $C_d(s)$ in the design example.

$F_{\dot{\beta}}$ Gain used in the forward speed proportional controller.

F_γ Gain of the proportional term in $C_\gamma(s)$ in the design example.

$F_{\dot{\gamma}}$ Gain of the integrator term in $C_{\dot{\gamma}}(s)$ in the design example.

g Gravity.

g_n, ..., g_0 Coefficients of z^n, ... z^0 in the numerator of $P_{\dot{\gamma}}[z]$.

G Centre of gravity of the bicycle rear frame and rigid rider.

$\mathbf{G}(\mathbf{q}, \dot{\mathbf{q}}, t)$ Remaining terms in the equations of motion that are not the coefficients of $\ddot{\mathbf{q}}$ or $\boldsymbol{\varrho}$ and not the generalized forces.

h Distance that point of contact the bicycle front wheel trails the intersection of the steer axis of the front frame and the ground.

$\mathbf{H}_G$ Angular momentum of the rear frame and rigid rider at G.

$\mathbf{H}_K$ Angular momentum of the front frame at K.

$\mathbf{H}_L$	Angular momentum of the front wheel at L.
$\mathbf{H}_P$	Angular momentum of the rear wheel at P.
$H_{\dot{\gamma}}(s)$	Transfer function of the closed loop response of the bicycle with the yaw-rate controller, $C_{\dot{\gamma}}(s)$.
$\tilde{\mathbf{I}}$	General inertia tensor.
$\tilde{\mathbf{I}}_{G,rf}$	Inertia tensor of the rear frame and rigid rider at G.
$\tilde{\mathbf{I}}_{G,rf,ref}$	Inertia tensor of the rear frame and rigid rider at G in the bicycle reference configuration.
$\tilde{\mathbf{I}}_{K,ff}$	Inertia tensor of the front frame at K.
$\tilde{\mathbf{I}}_{K,ff,ref}$	Inertia tensor of the front frame at K in the bicycle reference configuration.
$\tilde{\mathbf{I}}_{L,fw}$	Inertia tensor of the front wheel at L.
$\tilde{\mathbf{I}}_{L,fw,ref}$	Inertia tensor of the front wheel at L in the bicycle reference configuration.
$\tilde{\mathbf{I}}_{P,rf}$	Inertia tensor of the rear wheel at P.
$\tilde{\mathbf{I}}_{P,rf,ref}$	Inertia tensor of the rear wheel at P in the bicycle reference configuration.
I_{xx}	Moment of inertia about the global x axis.
I_{xy}, I_{yx}	Products of inertia described by $I_{xy} = -\int_m xy\,dm$.
I_{xz}, I_{zx}	Products of inertia described by $I_{xz} = -\int_m xz\,dm$.
I_{yy}	Moment of inertia about the global y axis.
I_{yz}, I_{zy}	Products of inertia described by $I_{yz} = -\int_m yz\,dm$.
I_{zz}	Moment of inertia about the global z axis.
J	Quadratic cost function to minimize for computing optimal control gains.
K	Centre of gravity of the bicycle front fork.
l_j	Linear distance between waypoint j and start point (or equivalently the end point) of the circular arc in the corner of waypoint j.
$\mathbf{l}_l$	Vector containing the length of each straight sections between the circular arcs in a practical path.
l_{lj}	Linear distance between the circular arcs at the corners of waypoint j and waypoint $j+1$.
L	Centre of gravity of the bicycle front wheel.
$\mathcal{L}$	Lagrangian of the bicycle
m_{ff}	Mass of the front frame.
m_{fw}	Mass of the front wheel.
m_{rf}	Mass of the rear frame and rigid rider.
m_{rw}	Mass of the rear wheel.
$\mathbf{M(q)}$	Inertia matrix in equations of motion that is a function of generalized coordinates.

$\mathbf{n}_L$ Vector parallel to the rotation axis of the front wheel.

$\mathbf{n}_P$ Vector parallel to the rotation axis of the rear wheel.

$\hat{n}_\delta$ Unit vector along the positive steering axis.

$\hat{n}_\theta$ Unit vector indicating the positive axis of rotation for θ.

$\hat{n}_\phi$ Unit vector indicating the positive axis of rotation for ϕ.

$N_C(s)$ The numerator of $C_{\dot{\gamma}}(s)$.

$N_P(s)$ The numerator of $P_{\dot{\gamma}}(s)$.

O Origin of the global Cartesian coordinate axes.

P Centre of gravity of the bicycle rear wheel.

$\bar{\mathbf{P}}$ Unique positive-semidefinite matrix used in solving the Riccati equation.

$P_{\dot{\gamma}}(s)$ Linear continuous-time transfer function estimate from $u_{\dot{\gamma}}(s)$ to $\dot{\gamma}(s)$. ($\dot{\gamma}(s)$ is estimated by $\dot{\phi}(s)$.)

$P_{\dot{\gamma}}[z]$ Discrete time approximation of $P_{\dot{\gamma}}(s)$.

$\mathbf{q}$ Generalized position vector for the nonlinear bicycle model.

$\dot{\mathbf{q}}$ Generalized velocity vector for the nonlinear bicycle model.

$\ddot{\mathbf{q}}$ Acceleration velocity vector for the nonlinear bicycle model.

q_j Generalized position at index j in $\mathbf{q}$.

$\dot{q}_k$ Generalized velocity at index j in $\dot{\mathbf{q}}$.

$\bar{\mathbf{Q}}$ Positive-semidefinite matrix that weights the relative importance of the states in $\bar{\boldsymbol{\chi}}$ in J.

$\mathbf{r}$ Vector of reference trajectories for the lean and steer angles in the lean and steer controller.

$\bar{\mathbf{r}}$ Ordered set defining the radii of the circular arcs located at corner in a practical path.

$\mathbf{r}_B$ Position vector of point B.

$\mathbf{r}_d$ Example vector given in global coordinates.

$\mathbf{r}'_d$ Example vector given in local coordinates.

r_{fw} Radius of the front wheel.

$\mathbf{r}_G$ Position vector of point G.

$\mathbf{r}_{G/P,ref}$ Relative position vector of point G with respect to point P in the bicycle reference configuration.

$\mathbf{r}_K$ Position vector of point K.

$\mathbf{r}_{K/S,ref}$ Relative position vector of point K with respect to point S in the bicycle reference configuration.

$\mathbf{r}_L$ Position vector of point L.

$\mathbf{r}_{L/F}$ Relative position vector of point L with respect to point F.

$\mathbf{r}_{L/S,ref}$ Relative position vector of point L with respect to point S in the bicycle reference configuration.

$\mathbf{r}_P$ Position vector of point P.

$\mathbf{r}_{P/B}$ Relative position vector of point P with respect to point B.

r_{rw} Radius of the rear wheel.

$\mathbf{r}_S$ Position vector of point S.

$\mathbf{r}_{S/P,ref}$ Relative position vector of point S with respect to point P in the bicycle reference configuration.

$r_{\dot{\beta}}$ Reference for $\dot{\beta}$ in the forward speed controller.

$r_{\dot{\gamma}}$ Reference for the closed loop system described by $H_{\dot{\gamma}}(s)$.

r_δ Reference trajectory for the steer angle in the lean and steer controller.

r_δ^{sat} Saturation value of r_δ.

r_θ Reference trajectory for the lean angle in the lean and steer controller.

$\mathbf{r}_\rho$ Position vector of point ρ.

$\mathbf{R}_e(\vartheta)$ General rotation matrix along axis e with an angle of rotation ϑ.

$\mathbf{R}_{ff}$ Rotation matrix of the front frame.

$\mathbf{R}_{fw}$ Rotation matrix of the front wheel.

$\bar{R}_j$ Radius of the circular arc in the corner of waypoint j.

$\mathbf{R}_{rf}$ Rotation matrix of the rear frame.

$\mathbf{R}_{rw}$ Rotation matrix of the rear wheel.

S Point along the front fork steer axis of the bicycle.

$\bar{\mathbf{S}}$ Positive-definite matrix that weights the inputs, $\mathbf{u}_L$ in J.

t Time.

T Sample time.

T_{ff} Kinetic energy of the front frame.

T_{fw} Kinetic energy of the front wheel.

T_{rf} Kinetic energy of the rear frame and rigid rider.

T_{rw} Kinetic energy of the rear wheel.

T_{tot} Total kinetic energy for the nonlinear bicycle model.

$\mathbf{u}$ Generalized input force vector for the nonlinear bicycle model.

u_d Output of $C_d(s)$.

u_d^{sat} Saturation value of u_d.

$\mathbf{u}_L$ Input vector for the linearized lean and steer dynamics of the bicycle.

u_γ Output of $C_\gamma(s)$.

$u_{\dot{\gamma}}$ Input signal to the map from yaw rate to the steer and lean angles.

$\mathbf{U}$ Matrix of coefficients for the Lagrange multipliers in the equations of motion.

v Forward speed of the bicycle.

$\mathbf{v}_B$ Velocity vector of point B.

v_c	Smallest forward speed of the bicycle where the capsize eigenmode becomes unstable.
v_d	Smallest forward speed of the bicycle where the weave eigenmodes become complex.
$\mathbf{v}_F$	Velocity vector of point F.
$\mathbf{v}_G$	Velocity vector of point G.
$\mathbf{v}_K$	Velocity vector of point K.
v_l	Smallest forward speed that the nonlinear bicycle model exhibits a limit cycle.
$\mathbf{v}_L$	Velocity vector of point L.
$\mathbf{v}_P$	Velocity vector of point P.
v_w	Smallest forward speed of the bicycle where the weave, castering, and capsize eigenmodes are stable.
v_ρ	Speed of the point ρ.
V_{ff}	Potential energy of the front frame.
V_{fw}	Potential energy of the front frame.
V_{rf}	Potential energy of the rear frame and rigid rider.
V_{rw}	Potential energy of the rear wheel.
V_{tot}	Total potential energy for the nonlinear bicycle model.
w	Bicycle wheelbase.
$\mathbf{w}_{\rho j}$	Waypoint j in a practical path.
$\bar{\mathrm{w}}_{\rho j,x}$	Global Cartesian x coordinate of waypoint j in a practical path.
$\bar{\mathrm{w}}_{\rho j,y}$	Global Cartesian y coordinate of waypoint j in a practical path.
$\bar{\mathbf{W}}$	Controllability matrix of the linearized lean and steer dynamics of the bicycle augmented with integrator states.
$\mathbf{W}_\rho$	Ordered set of waypoints used for defining a practical path.
x	Global Cartesian x axis.
x_B	Global Cartesian x coordinate point B.
$\dot{x}_B$	Global Cartesian x velocity of point B.
x_P	Global Cartesian x coordinate point P.
$\dot{x}_P$	Global Cartesian x velocity of point P.
$\ddot{x}_P$	Global Cartesian x acceleration of point P.
x_ρ	Global Cartesian x coordinate point ρ.
$\dot{x}_\rho$	Global Cartesian x velocity of point ρ.
y	Global Cartesian y axis.
y_B	Global Cartesian y coordinate point B.
$\dot{y}_B$	Global Cartesian y velocity of point B.
$\dot{y}_P$	Global Cartesian y velocity of point P.

$\ddot{y}_P$ Global Cartesian y acceleration of point P.

y_P Global Cartesian y coordinate point P.

y_F Global Cartesian y coordinate point F.

y_ρ Global Cartesian y coordinate point ρ.

$\dot{y}_\rho$ Global Cartesian y velocity of point ρ.

z Global Cartesian z axis.

z_P Global Cartesian z coordinate point P.

α Eigenvalues of the state-space representation of the linearized lean and steer dynamics of the bicycle.

β Angle of rotation of the bicycle rear wheel along the rear wheel axis of rotation.

$\dot{\beta}$ Angular velocity of the bicycle rear wheel along the rear wheel axis of rotation.

$\ddot{\beta}$ Angular acceleration of the bicycle rear wheel along the rear wheel axis of rotation.

γ Yaw angle of the bicycle.

$\dot{\gamma}$ Yaw rate of the bicycle.

$\dot{\boldsymbol{\gamma}}$ Vector containing the output values of $\dot{\gamma}$ (estimated by $\dot{\phi}$ of the actual system) at select times for system identification.

$\dot{\boldsymbol{\gamma}}^{mod}$ Vector containing the values of $\dot{\gamma}$ of the transfer function model at select times for system identification.

$\dot{\gamma}^{mod}(s)$ Output of the transfer function model $P_{\dot{\gamma}}(s)$.

$\dot{\gamma}[t_k]$ Output value of $\dot{\gamma}$ (estimated by $\dot{\phi}[t_k]$ of the actual system) at time t_k.

$\dot{\gamma}^{mod}[t_k]$ The value of $\dot{\gamma}$ of the transfer function model at time t_k in system identification.

δ Angle of rotation of the front frame along the front frame steer axis.

$\dot{\delta}$ Angular velocity of the front frame along the front frame steer axis.

$\ddot{\delta}$ Angular acceleration of the front frame along the front frame steer axis.

δ_f Effective steering angle.

Δ This symbol is used as a prefix indicating a linearized variable.

ε Sum of the outputs of $C_\gamma(s)$ and $C_d(s)$. It is the corrective yaw rate signal that is subtracted from λ to cause the bicycle to track a path.

ζ Vector of integrator states augmented to the output of the state-space representation of the linearized lean and steer dynamics of the bicycle.

ζ_d State of the integrator in $C_d(s)$.

$\zeta_{\dot{\gamma}}$ State of the integrator in $C_{\dot{\gamma}}(s)$.

ζ_δ Integrator state in ζ augmented to the output of δ.

ζ_θ	Integrator state in ζ augmented to the output of θ.
$\boldsymbol{\eta}$	Non-linear output vector.
θ	Second Euler angle in the 3-1-3 Euler angle sequence describing the rear frame rotation.
$\dot{\theta}$	Angular velocity of the second Euler angle in the 3-1-3 Euler angle sequence describing the rear frame rotation.
$\ddot{\theta}$	Angular acceleration of the second Euler angle in the 3-1-3 Euler angle sequence describing the rear frame rotation.
κ	Curvature of the path or virtual path.
λ	Yaw angle of the path at point ρ.
$\dot{\lambda}$	Yaw rate of the path at point ρ.
μ	Angle of the steer axis of the front frame.
ξ	Angle of rotation of the bicycle front wheel along the front wheel axis of rotation.
$\dot{\xi}$	Angular velocity of the bicycle front wheel along the front wheel axis of rotation.
$\ddot{\xi}$	Angular acceleration of the bicycle front wheel along the front wheel axis of rotation.
ρ	Point on the path that is closest to point B of the bicycle.
$\boldsymbol{\varrho}$	Vector of Lagrange multipliers used to solve equations of motion.
ϱ_j	Lagrange multiplier j in vector $\boldsymbol{\varrho}$.
ς_j	Angle formed between the line segment defined by waypoint $j-1$ and waypoint j and the line segment defined by waypoint j and waypoint $j+1$
τ_β	Torque applied to the bicycle along the β axis of the rear wheel.
τ_δ	Torque applied to the bicycle along the δ (steer) axis of the front frame.
τ_θ	Torque applied to the bicycle along the θ axis the rear frame.
ϕ	First Euler angle in the 3-1-3 Euler angle sequence describing the rear frame rotation.
$\dot{\phi}$	Angular velocity of the first Euler in the 3-1-3 Euler angle sequence describing the rear frame rotation.
$\ddot{\phi}$	Angular acceleration of the first Euler in the 3-1-3 Euler angle sequence describing the rear frame rotation.
$\varphi(\mathbf{q}, \dot{\mathbf{q}}, t)$	Constraint equation i for the equations of motion of the bicycle.
$\bar{\boldsymbol{\Phi}}$	Known matrix when splitting γ^{mod} into known and unknown parts in system identification.
$\boldsymbol{\Phi}(\mathbf{q}, \dot{\mathbf{q}}, t)$	Matrix representing the velocity constraints of the front and rear wheels.

$\dot{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t)$ Matrix representing the differentiated velocity constraints of the front and rear wheels.

$\boldsymbol{\chi}$ State vector for the linearized lean and steer dynamics of the bicycle.

$\dot{\boldsymbol{\chi}}$ State vector derivative for the linearized lean and steer dynamics of the bicycle.

$\bar{\boldsymbol{\chi}}$ State vector for the linearized lean and steer dynamics of the bicycle augmented with integrator states.

$\dot{\bar{\boldsymbol{\chi}}}$ State vector derivative for the linearized lean and steer dynamics of the bicycle augmented with integrator states.

ψ Third Euler angle in the 3-1-3 Euler angle sequence describing the rear frame rotation.

$\dot{\psi}$ Angular velocity of the third Euler angle in the 3-1-3 Euler angle sequence describing the rear frame rotation.

$\ddot{\psi}$ Angular acceleration of the third Euler angle in the 3-1-3 Euler angle sequence describing the rear frame rotation.

ω General angular velocity vector.

ω_{fw} Angular velocity vector of the front wheel.

ω_{ff} Angular velocity vector of the front frame.

ω_{rf} Angular velocity vector of the rear frame and rigid rider.

ω_{rw} Angular velocity vector of the rear wheel.

ω_x Angular velocity along the global x axis.

ω_y Angular velocity along the global y axis.

ω_z Angular velocity along the global z axis.

$\tilde{\mathbf{\Omega}}$ General angular velocity tensor.

$\tilde{\mathbf{\Omega}}_{ff}$ Angular velocity tensor of the front frame.

$\tilde{\mathbf{\Omega}}_{fw}$ Angular velocity tensor of the front wheel.

$\tilde{\mathbf{\Omega}}_{rf}$ Angular velocity tensor of the rear frame and rigid rider.

$\tilde{\mathbf{\Omega}}_{rw}$ Angular velocity tensor of the rear wheel.

Chapter 1

Introduction

1.1 Motivation and Purpose

The bicycle has a storied history, with a number of design iterations. The first two-wheeled design is credited to Karl von Drais in 1817, and called the draisienne. This first attempt looked similar to modern bicycles, but was propelled by a walking motion rather than pedalling. The next major development was called the velocipede [1]. The design features pedals attached to the front wheel of a draisiene to allow the user to propel it forward. This device is also called the bone-shaker due to the metal wheels and cobblestone roads of the time period making for a bumpy ride [2]. The 1870's saw the rise of the ordinary bicycle or penny farthing. The design is similar to the velocipede, but features a very large front wheel with a rubber tire that reduces the vibration felt by the rider. This large front wheel kept the rider sitting up high. Due to this positioning the combined rider and penny farthing had a high centre of gravity that would allow the machine to easily rotate forward causing the rider to fall face first toward the ground [2]. This safety risk led to the innovation of the rear driven safety cycle introduced by H. J. Lawson in 1879. While the new design was not immediately popular, in 1885 the "Rover" safety cycle by Messrs, Starley, and Sutton became the first commercially successful version [1]. Except for some minor improvements including pneumatic tires that came after the introduction of the safety cycle, this design is essentially the modern bicycle [3].

Today, bicycles provide an inexpensive and environmentally friendly method of transportation that is commonly used in many countries. Yet on a typical commute in North America there are few cyclists to be seen, despite wanting to reduce our carbon footprint. There are a number of issues causing these lower numbers, but bicycle automation may

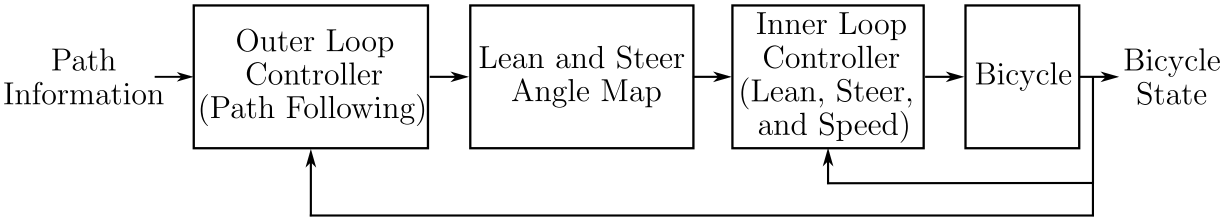

Path Information
Outer Loop Controller (Path Following)
Lean and Steer Angle Map
Inner Loop Controller (Lean, Steer, and Speed)
Bicycle
Bicycle State

Figure 1.1: General structure of multi-loop control approach for the autonomous bicycle

help in cases where distance is a factor, and be helpful for those learning to ride a bike. The problem of automation is also surprisingly complex due to the nonlinear dynamics that vary with forward speed.

In this thesis we present the systematic multi-loop control approach outlined in Figure 1.1 to stabilize a modern bicycle. An important feature is that the technique guarantees asymptotic convergence to constant curvature paths. A key advantage of our technique is that it uses linear control theory, which readily enables stability analysis and controller tuning. Most of the content in this thesis is published in [4].

This thesis is organized as follows: A bicycle model is described in Chapter 2; an inner-loop controller is designed in Chapter 3; an outer-loop path-following controller is designed in Chapter 4; and in Chapter 5 we present a design example and simulations for demonstrating the performance of the example. The remainder of this chapter is dedicated to providing a definition of the types of paths to be followed by the bicycle and a literature review.

1.2 Defining Paths

The main objective in this thesis is to design a control system that causes the point of contact between the ground and rear wheel of a modern bicycle, B, to asymptotically track paths of constant curvature. These paths correspond to lines and circles. The control methods used to achieve path-following while simultaneously balancing the bicycle are outlined in Section 2.1, and a detailed block diagram is given in Figure 4.6. Asymptotic tracking is important as it it means that given enough time the bicycle will converge to the path, and by combining these constant curvature elements more practical paths can be constructed.

To create these paths a set of N points are specified in the global xy-plane,

$$\mathbf{W}_\rho = \{\mathbf{w}_{\rho 1}, \mathbf{w}_{\rho 2}, ..., \mathbf{w}_{\rho N}\}, \tag{1.1}$$

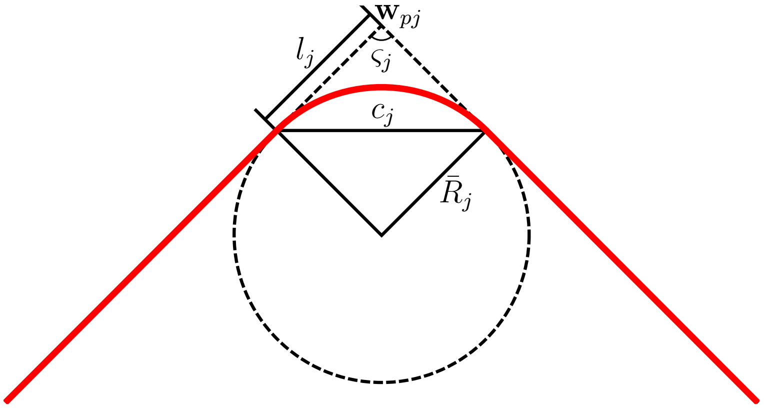


Figure 1.2: A typical corner in a practical path with important measures labelled, where red shows the actual path to be travelled by the bicycle.

where point j, $\mathbf{w}_{\rho j} = \begin{bmatrix} \bar{\mathrm{w}}_{\rho j,x} & \bar{\mathrm{w}}_{\rho j,y} \end{bmatrix}^T$. Line segments are drawn connecting $\mathbf{w}_{\rho j}$ and $\mathbf{w}_{\rho (j+1)}$ as well as $\mathbf{w}_{\rho N}$ and $\mathbf{w}_{\rho 1}$ to form a closed path. The result is N line segments, with an intended direction of travel from $\mathbf{w}_{\rho j}$ towards $\mathbf{w}_{\rho (j+1)}$, and then from $\mathbf{w}_{\rho N}$ to $\mathbf{w}_{\rho 1}$. These lines alone do not provide a useful path to be tracked by this bicycle system, as the curvature for a corner between two line segments is not continuously differentiable and at the point of intersection the curvature is $\kappa = \infty$, which corresponds to a yaw rate $\lambda = \infty$. To ease this transition, a set of N radii are also given in order,

$$\bar{\mathbf{r}} = \{\bar{R}_1, \bar{R}_2, ..., \bar{R}_N\}. \tag{1.2}$$

Each radius in this set represents the radius of a circular arc that is put into the path. The circular arc corresponding to radius j, $\bar{R}_j$, will be inserted such that both line segments that connect to $\mathbf{w}_{\rho,j}$ are tangent to it, as shown in Figure 1.2.

Additionally, the bicycle should attempt to follow the line segments and circular arcs in order. To achieve this task we do not allow the bicycle to leave the current line segment from $\mathbf{w}_{\rho (j-1)}$ to $\mathbf{w}_{\rho j}$ and follow the circular arc related to $\bar{R}_j$ until it passes the line that contains line segment c_j, shown in Figure 1.2. It then cannot continue to the line segment between $\mathbf{w}_{\rho j}$ and $\mathbf{w}_{\rho (j+1)}$ until it again crosses the line that c_j is a segment of.

Two constraints on path specification are imposed by requiring that each line segment and circular arc contribute to the actual path considered. To state them we require two measures. The first is angle ς_j and the second is length l_j, where they are defined as shown in Figure 1.2. These two measures can be found through geometry to be

$$\begin{aligned} \varsigma_j &= \cos^{-1}\left(\frac{\left|\left|\mathbf{w}_{p(j-1)} - \mathbf{w}_{pj}\right|\right|^2 + \left|\left|\mathbf{w}_{pj} - \mathbf{w}_{p(j+1)}\right|\right|^2 - \left|\left|\mathbf{w}_{p(j-1)} - \mathbf{w}_{p(j+1)}\right|\right|^2}{2\left|\left|\mathbf{w}_{p(j-1)} - \mathbf{w}_{pj}\right|\right|\left|\left|\mathbf{w}_{pj} - \mathbf{w}_{p(j+1)}\right|\right|}\right), \\ l_j &= \bar{R}_j \cot\frac{\varsigma_j}{2}. \end{aligned} \tag{1.3}$$

These measures provide the basis for the two constraints that we impose:

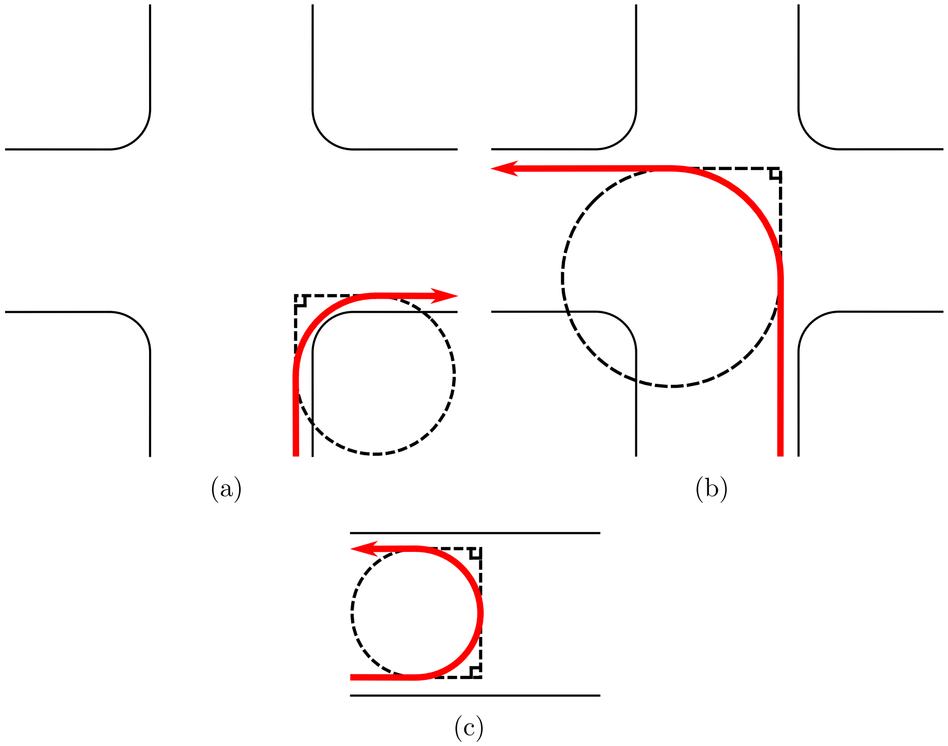
$$\begin{gathered} 0 < \varsigma_j < \pi, \\ l_j + l_{j+1} \leq \left|\left|\mathbf{w}_{pj} - \mathbf{w}_{p(j+1)}\right|\right|. \end{gathered} \tag{1.4}$$

The paths that we consider, made of straight line segments and circular arcs, are continuously differentiable and are related to Dubins paths [5]; however, unlike Dubins paths, we do not force the circular segments on the path to be the maximum curvature that can be tracked by the bicycle, and our points of interest from (1.1) are never actually on the path. An application for this type of path is to consider the bicycle on a road without obstacles, such as a country road with no traffic. The bicyclist may wish to plan a route starting at location A pass by location B and then return to A. In Figure 1.3 common manoeuvres that may be required for such a path are presented and more complicated manoeuvres can be created by using combinations of those shown.

In Chapter 5 an example of a bicycle following a practical path is demonstrated (see Figure 5.15(a)). For that example the bicycle is allowed to use an entire lane on a rural road. To gauge the restrictions on lane width and turning radius the American Association of State Highway and Transportation Officials (AASHTO) recommendations for lane width and curb radius are used. The AASHTO [6] suggests that urban arterial streets in the absence of pedestrian conflicts should have a curb radius of 4.5 to 7.5 m for passenger vehicles and 9 to 15 m for trucks and buses. In rural areas a lane width of 2.7 to 3.6 m is recommended. As there are few pedestrians and speeds are typically higher in rural areas, a curb radius of 7.5 m is assumed, and a 2.7 m lane width. Assuming that the intended path of the bicycle is in the middle of the lane, the radius of right and left turns is 8.85 m and 11.6 m respectively.

1.2.1 Virtual Path

A problem arises when the bicycle starts far from the path and/or is not nearly tangential to it. The nature of the problem is described in Section 5.4.2. The solution we employ



(a)

(b)

(c)

Figure 1.3: Typical path construction for right turns, left turns, and u-turns, respectively, in (a), (b), and (c) for bicycle path planning.

is to create another route, termed the virtual path, that the bicycle can follow to return to the actual path. This virtual path is tangent to the start position of the bicycle and connects tangentially to the closest component on the actual path.

There are a number of ways to construct such a virtual path. The method we use here involves first placing two circles. These circles form the start and end of the virtual path, and they are joined by a line tangent to both circles such that the travel direction is always consistent. The first circle, C_1, should be placed such that the initial yaw angle of the bicycle is tangent to it, and the second, C_2, should be placed such that the closest point on the path, ρ, is on C_2 and the yaw angle is tangent to the circle. There are two possible choices for each of the circles, one in the clockwise direction (CW), and the other counter clockwise (CCW), but some combinations of possible C_1 and C_2 may not work. Additionally, these circles can have any curvature, κ, such that $\kappa < \max \kappa$, where $\max \kappa$ is the maximum curvature path that can be followed by the bicycle. An approximation for $\max \kappa$ is found in Section 4.1. It is suggested that C_1 and C_2 have the same κ since there are zero tangent lines between two circles when one of smaller radius is completely inside a larger one. The result is that the bicycle will follow a circular arc along C_1 until reaching the tangent line that is consistent with the current direction of travel. It will then travel along the line to a second circular arc from the point the tangent line connects to C_2 to the point on the actual path, and will be travelling in approximately the correct direction.

To determine which combinations of the possible circles for C_1 and C_2 will work, we examine the four cases for tangent lines when the circles are the same size as shown in Figure 1.4. If there is a path from C_1 to C_2 that is consistent with the direction of travel, then it is obvious that the bicycle can travel anywhere from C_1 to anywhere on C_2. If the two circles form the case in Figure 1.4(a) then the direction of travel on each circle is arbitrary. If the direction of travel is the same then the tangent line connecting the circles will be one of the outer tangent lines, shown in green, that do not cross each other. If they are different then the appropriate inside, crossing tangent line, shown in blue, will connect them. The same situation is present in the case of Figure 1.4(b), which is a special case of Figure 1.4(a) where the inside tangent lines are instead a common point. If the case in Figure 1.4(c) represents the configuration of the two circles, then both circles must have the same direction of travel, since the only tangent lines present are the outer tangent lines which require the same direction of travel on both circles for a consistent travel direction. The final case is when the circles fully overlap as in Figure 1.4(d). This case is trivial and only admissible if the two circles require the bicycle to travel in the same direction.

The example in Figure 1.5 shows the four possible combinations of the virtual path for a particular κ of C_1 and C_2 when there is no overlap between any combination of C_1 and C_2. In this example the path shown in red can be assumed as a linear path segment, or a

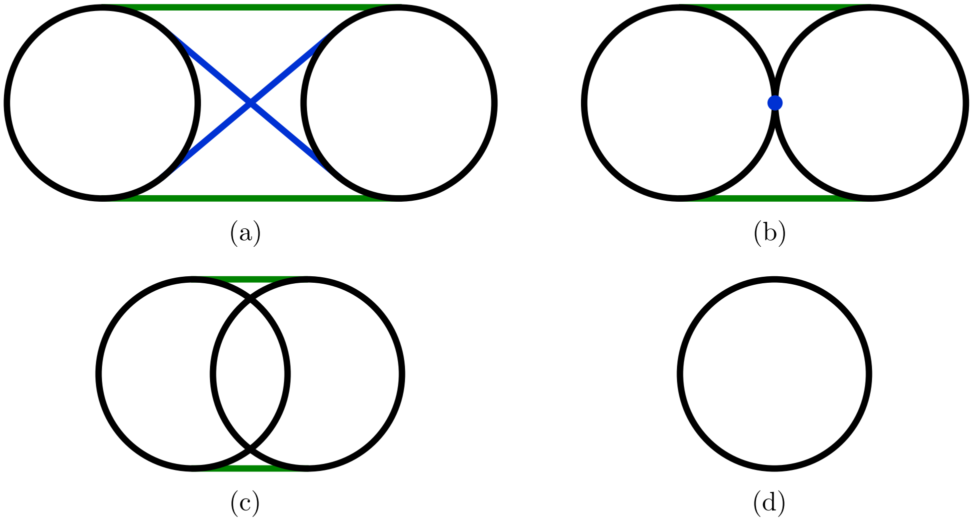


Figure 1.4: Tangent lines between two circles of the same radius are shown for the cases of no overlap, a single point of overlap, there are two points of overlap, and full overlap are shown in (a), (b), (c), and (d), respectively. Any outer tangent lines between the two circles are displayed in green, and inner tangent lines or points of tangency are in blue.

line tangent to a circular path segment.

It should be noted that the resulting virtual path and the connection to the actual path is continuously differentiable. An alternative to this method, when the minimal path is desired, is to construct a Dubins path with a starting condition defined by B and γ, and a terminal condition of ρ and λ. In fact for certain conditions where C_1 and C_2 have $\kappa = \max \kappa$, one of the resulting paths may be the minimal path, as we consider the class of solutions described by a curved segment, followed by a straight segment and finally a curve segment, where the curves have $\kappa = \max \kappa$, as described by Dubins [5]. We do however miss the class of solutions defined by a three consecutive curved segments of $\kappa = \max \kappa$, where the segments alternate between a clockwise and counter-clockwise orientation.

For the purpose of demonstration in Chapter 5, we are not concerned with the minimal path. Instead, C_1 and C_2 are chosen to have $\kappa = \frac{1}{2} \max \kappa$ and give both a consistent rotation direction that minimizes the number of times the bicycle crosses the path when ρ is in a straight path segment. In Figure 1.5 the choice would be $C_{1,CW}$ and $C_{2,CW}$. If ρ is a point on a circular arc in the path, then C_1 and C_2 are set to the same κ and rotation direction as the arc. The intent of these choices is to aid in keeping the bicycle from leaving its lane. In the case of the straight-line path segment, a lower value of κ than

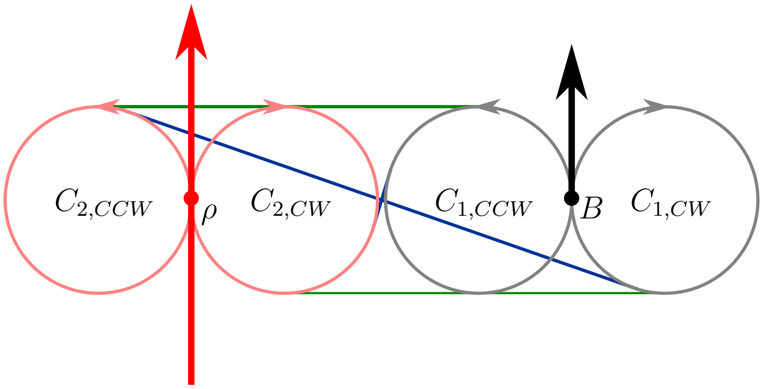


Figure 1.5: Example of virtual path construction when all combinations of C_1 and C_2 do not overlap. The actual path is displayed in red, while the heading of the bicycle is displayed in black. The potential choices of C_1 are in grey, while the options for C_2 are in pink. The tangent line connecting each choice for C_2 to each choice of C_1 is green when the connection is an outer tangent line and blue when it is an inner tangent line.

the maximum causes a smaller yaw-rate step, and in the case of an arc path segment, a rotation direction in the same direction as the turn produces a smaller yaw-rate step than a transition of opposite rotation.

1.3 Literature Review

1.3.1 Bicycle Modelling

The complex nature of bicycle motion has attracted the study of many people, even before the modern incarnation of the bicycle. An early noteworthy analysis is that of Rankine [7], who in 1869 described the motion of a velocipede. Soon after the safety cycle was introduced, dynamic models were derived describing the system dynamics. The first attempt at a mathematical analysis of safety bicycle motion was submitted by Carvallo [8]; however, an independent study by Whipple [9] was published first in 1899. Hand [10] points out that while Carvallo presented correct equations, the model is a specialized version of the model in Whipple's paper, where Carvallo assumed a massless fork. In [9] Whipple developed what is now commonly referred to as the Whipple Model. The assumptions made in this model are discussed further in Section 2.2; it is sufficient for now to point out that the model assumes frictionless internal connections, a rigid rider, knife edge rolling constraints, and a flat horizontal rolling surface.

Since Whipple and Carvallo, there have been many attempts to derive a complete model that describes the dynamics of a bicycle. Both Hand [10] and Meijaard [11] completed extensive reviews of linearized bicycle models, highlighting that many of the models are plagued by mathematical errors. They indicate that Whipples own nonlinear derivation contains an error, which does not affect the linear model. The linear model is correct barring some typographical errors. Hand [10] and Meijaard [11] also found that the first completely correct governing equations for the Whipple model were not developed until 1955 by Döhring [12] using Newtonian mechanics. Hand developed equations linearized about the vertical operating point for the Whipple bicycle model using Lagrangian techniques. He explains that the steps he followed are the corrected version of those in Neǐmark and Fufaev [13]. The resulting equations are in in agreement with the those developed by Döhring. The same model was presented by Papadopoulos (one of Hand's advisors) in 1987 [14], later independently derived by Meijaard [11], and a model of the same form has also appeared in Astrom et al. [15].

There has been work on more compliant bicycle models including more complicated rider models [16], and wheel models [17]. Additionally, significant work on motorcycle dynamics has also been completed. Examples of motorcycle models of high compliance are presented by Koenen [18], and Sharp [19], which is later expanded in [20]. Sharp [21] has also reviewed a number of more compliant motorcycle models for accuracy.

Nonlinear bicycle models have also been presented such as a model by Lennartsson [22], and a further simplified bicycle model with a vertical steer axis angle and no wheel mass or inertia by Getz [23]. Due to the high complexity, the nonlinear models are difficult to critique; however, Basu-Mandal and associates have provided a qualitative analysis of a small set in [24] and [25].

Of particular interest for this thesis are the nonlinear Whipple models developed by Basu-Mandal and associates [24] [25] using both Newtonian and Lagrangian mechanics, as well as the standard Whipple model derived by Moore [26] using Kane's method [27]. These are of specific importance since Moore [26] claims the numerical output of his model matches the two presented by Basu-Mandal [24] [25]. The models also linearize to the (corrected) Whipple model, and the model found using Lagrangian mechanics by Basu-Mandal [24][25] is the model used for control design in this thesis.

1.3.2 Lean and Steer Control

The linear strategies for controlling the lean and steer of a bicycle include linear-parameter-varying (LPV) control presented by Andreo et al. [28]. Mutsaerts [29] utilizes the action of steering into the direction of fall (with a proportional controller) to keep a bicycle built with Lego NXT parts from tipping over. Findlay et al. [30] use a classical lead/lag controller, pole placement, and linear quadratic regulator (LQR) optimal control to stabilize a point mass bicycle. Lam [31] uses a proportional-derivative controller to balance a bicycle using a control moment gyroscope to affect the lean.

Other methods of control have often employed nonlinear and even fuzzy control strategies. These include the use of variable structure under-actuated control by Hwang et al. [32], and fuzzy sliding-mode under-actuated control again by Hwang et al. [33]. More recently, cascade sliding mode control is explored by Yu et al. [34]. Chen and associates use sliding mode control [35] and fuzzy control [36] provide tracking for the lean angle tracking of a bicycle.

There has also been work done to identify and model how humans control single-track vehicles, e.g. see Weir [37] and Moore [26].

1.3.3 Bicycle Path Following

The non-linear bicycle model presented by Getz [23] often appears in analysis of path following and manoeuvring problems. Papers using this model to develop control techniques include those by Getz himself in [23], [38]. Here Getz determines an equilibrium lean angle manifold that causes the bicycle to track the desired path and then uses linearization of the dynamics from steer angle to lean angle to stabilize the lean angle onto this manifold. Frezza et al. [39] also use this model. They treat the roll angle of a motorcycle as the input and used model predictive control to determine the optimal roll angle trajectory for the path preview, which provides the prediction horizon. They then invert the dynamics to determine the steering and forward speed required. Consolini and Maggiore [40] use virtual holonomic constraints to cause a bicycle described by Getz model to perfectly track a closed Jordan curve when certain initial conditions are met. Yi et al. [41] use sliding mode control to stabilize a bicycle when stationary, and adopt internal/external model decomposition methods described by Getz [42] to design a nonlinear tracking controller. Hauser et al. [43], [44], do not provide an actual controller, but have used this model to prove if a prescribed manoeuvre is possible for a motorcycle.

While popular, the Getz model is not exclusively used for path following of single-track vehicles. Yamakita and Utano [45] present equations for a bicycle with an inverted pendulum like device attached to the frame top-tube for balancing the bicycle. They separately design nonlinear stabilizing control to balance the bicycle and use a method based on input-output linearization for tracking control. The interference of the two separate control methods is then compensated for. Keo and Masaki [46] present a model similar to [45]. They use output-zeroing control applied to the balancer to balance the bicycle, and derive path-following control based on input-output linearization.

The Whipple model has also seen application in path following. Sharp [47] presents an optimal linear control strategy for a bicycle described by the Whipple model. The technique is based on a path-preview scheme that simultaneously causes the bicycle to follow the previewed path and maintain balance using only steering input. Chen and Dao [48] use a nonlinear Whipple model developed in [36] and use system identification to obtain the parameters for the linear model in [11]. Using only a steering torque, they control the lean angle of the bicycle using sliding mode control, and use a fuzzy controller to command this lean angle to cause the bicycle to track a path. While not specifically path following, Peterson and Hubbard [49] extend the Whipple model to include a rider that is able to lean relative to the bicycle. They develop an LQR controller based on this model that provides velocity and yaw rate tracking using the rider lean torque and a rear wheel torque.

1.4 Thesis Contributions

In this thesis we develop a novel control architecture to simultaneously stabilize the nonlinear Whipple bicycle, and cause it to asymptotically track paths of constant curvature using systematic multi-loop control based entirely on linear control techniques. We use a linear map that is dependent on velocity to relate a yaw-rate actuator signal, $u_{\dot{\gamma}}$, to a lean angle and steer angle reference for our inner-loop LQR controller (see Figure 4.6). Using such a linear map has the advantage of simplicity in determining required steer angle and desired lean angle. Due to the choice of a linear controller, the inner-loop only typically provides stable and acceptable lean angle and steer angle tracking close to the operating trajectory, so we must appropriately saturate the reference. The linear map also enables us to provide a linear approximation of the maximum yaw rate the bicycle can track based on the saturation of the steer angle. There are some disadvantages to this approach versus a nonlinear map, such as the map losing accuracy as the steer angle and/or lean angle of the bicycle increases. By using a multiple loop approach we also have the disadvantage of having two separate loops to control yaw rate in comparison to [49].

The bicycle path-following control we develop is unique and achieves perfect steady-state tracking of constant curvature paths. There are significant cross-channel interactions in the lean angle and steer angle dynamics of the bicycle (see Section 3.2.2), so we use a system identification approach to determine the yaw-rate response of the inner-loop and the linear map. We use this model to develop feedback to control the bicycle yaw-rate directly. The idea of yaw-rate control is not new, and it has been used previously in [49]; however, the use of system identification to obtain a model for the yaw-rate of a bicycle with a lean and steer control is. An advantage is that our approach provides allows us to separate path-following control from lean and steer control. We then design and analyze the path-following control as a SISO system using linear classical control methods. A disadvantage is that there are some additional steps associated with system identification and the design of an additional controller. This approach is also linear which means that acceptable performance may only be obtained near the operating velocity and for yaw rate signals of similar frequency and amplitude to the signals used in system identification.

To actually track a path we use the difference between the bicycle and path yaw rates, as well as the distance of the bicycle to the path, to adjust the commanded yaw rate and converge to the desired path. We accomplish this task again using linear methods. Many other methods involve calculating the lean angle required to track a path by relating the steer angle to the lean angle. Our linear method allows us to readily analyze the entire system and show that when the bicycle starts close to a path of constant curvature that convergence to the path is guaranteed. However, this approach is again linear and the overall transfer function we find (4.35) is parameterized with velocity so control performance may deteriorate away from the operating velocity.

To the best of the author's knowledge the use of such a virtual path to help with path following is new. In this work it allows us to overcome some limitations that may arise when the bicycle is far from the actual path, we also introduce the concept of a virtual path. By introducing a virtual path when the bicycle is far from the actual path we can ensure that the bicycle is always in close proximity to a path.

Chapter 2

Control Strategy and Modelling

In Section 2.1 we provide a high-level description of the control approach that we develop to both stabilize a bicycle and cause it to perfectly track constant curvature paths in steady state. The chapter progresses into the development of the equations of motion for the nonlinear model that we use to approximate the dynamics of a modern bicycle. We start this discussion in Section 2.2, where we present the assumptions of the model and some important conventions and mathematical notation we use in deriving the equations of motion. The equations of motion are derived using Lagrangian mechanics so we continue model analysis by finding the kinetic and potential energies in Section 2.3, and the constraints on motion in Section 2.4. Finally, in Section 2.5 we discuss how to arrive at the final equations of motion and how to choose initial conditions; we also present a method of solving the equations of motion for simulation.

2.1 Control Overview

A systematic multi-loop control approach is used to stabilize the nonlinear model of a bicycle and cause it to asymptotically track a path of constant curvature. Figure 1.1 shows the two main loops used in this approach. Here the inner-loop controller is responsible for the steer and lean angles of the bicycle and is designed to be higher bandwidth than the outer loop.

The outer loop, which is in fact significantly more complicated than indicated in Figure 1.1, achieves the actual path tracking. This is done by not only considering the distance of the bicycle to the path, but also comparing the yaw rate and yaw angle of the bicycle to a point we determine on the path. This information is used to determine the target yaw rate of the bicycle at a given instant. The desired yaw rate is then mapped to the required steer angle and desired lean angle.

The loops have been designed in a way that asymptotic tracking is guaranteed for constant curvature paths. We design the loops using linear control techniques. A key advantage of this method is that both loops are easy to tune and convergence can be readily proven.

2.2 Modelling Preliminaries

2.2.1 Background

We chose to use the nonlinear version of the (corrected) Whipple Model presented by Basu-Mandal and associates [24] [25]. There can be a high confidence in the correctness of the model as they derived the equations of motion using both Newtonian and Lagrangian mechanics and tested for numerical equivalence. Moore [26] also independently derived the equations of motion for the nonlinear Whipple model using Kane's method, and showed that his results were equivalent. Additionally, validity of the linearized version of this model has been experimentally verified for some velocities by Kooijman [50].

2.2.2 Assumptions

The Whipple model assumes that the bicycle is made of a front fork assembly, a rear frame with a rigidly attached rider, and two wheels. These are connected in the usual configuration of a bicycle as shown in Figure 2.1, with each link assumed to be a frictionless hinge. The wheels are assumed to be axisymmetric, rolling without slipping on a knife edge contact over the ground, which is flat and in the horizontal plane.

Important properties of the bicycle used in derivation of the equations of motion are given in Appendix A. Other than some minor differences, these are consistent with the benchmark bicycle in [24].

2.2.3 Modelling Notation Conventions

Scalar values are presented using both lower and upper case symbols with a normal weight font. Lower case bold symbols represent vectors, and upper case bold symbols matrices.

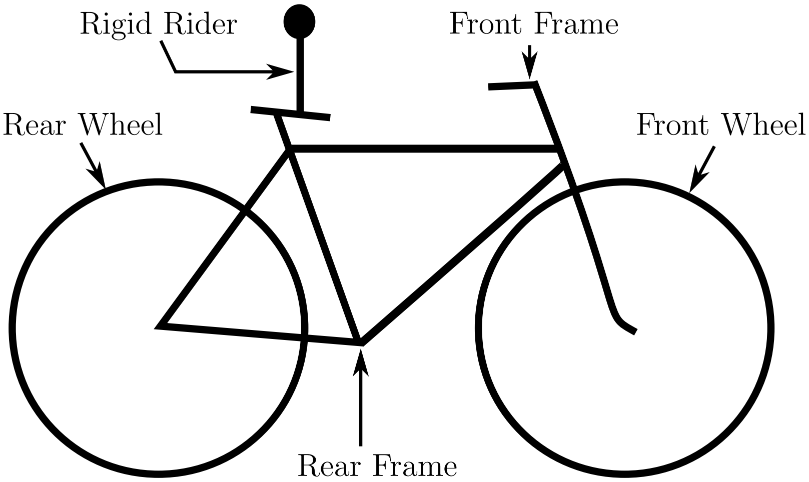


Figure 2.1: Configuration of components on a bicycle with a rigid rider connected to the seat.

Any tensors have a diacritic tilde, and unit vectors feature a circumflex.

The Cartesian coordinate system is used in describing some positions. The global Cartesian axes are denoted as x, y, and z. Unit vectors in the global x, y, and z axis are represented using $\hat{\mathbf{e}}_1$, $\hat{\mathbf{e}}_2$, and $\hat{\mathbf{e}}_3$ respectively.

2.2.4 Rotational Transformations

Rotations are represented using rotational matrices, which transform a vector $\mathbf{r}_d \in \mathbb{R}^3$ in the global coordinate system to $\mathbf{r}'_d \in \mathbb{R}^3$, a vector in local coordinates with the same origin. These matrices are expressed as $\mathbf{R}_e(\vartheta) \in \mathbb{R}^{3x3}$, where e and ϑ represent the axis and angle of rotation respectively. So a general rotation takes the form

$$\mathbf{r}'_d = \mathbf{R}_e(\vartheta)\mathbf{r}_d. \tag{2.1}$$

2.2.5 Angular Velocity and Inertia Tensors

Angular accelerations are described using the derivative of rotation matrices. The matrix produced is the angular velocity tensor,

$$\tilde{\mathbf{\Omega}} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{2.2}$$

After obtaining this skew symmetric matrix it is then possible to find the individual components of the angular velocity and form the angular velocity vector,

$$\boldsymbol{\omega} = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}. \tag{2.3}$$

Inertia tensors are written such that the all elements are positive. Specifically,

$$\tilde{\mathbf{I}} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}, \tag{2.4}$$

which implies that the inertia products are calculated such that $I_{cd} = -\int_m cd\, dm$.

2.2.6 Degrees of Freedom

By assessing the system it can be readily shown that there are seven degrees of geometric freedom. This comes from having four rigid bodies: the fork assembly, a rear frame with a rigidly attached rider, and two wheels. If unconstrained these four bodies would have a total of 24 degrees of freedom; however, the two knife edge constraints of the wheels each eliminate one geometric degree of freedom, and the three hinges each eliminate five degrees of freedom.

The knife edge constraints of the wheels also introduce two nonholonomic constraints, which means that there are only three degrees of freedom in velocity.

2.2.7 Generalized Coordinates

For convenience a non-minimal set of nine coordinates are used to derive the equations of motion of the bicycle as in [24]. The first three describe the position of the center of mass of the rear wheel, P, using Cartesian coordinates, denoted as x_P, y_P, and z_P. The next three coordinates are the Euler angles ϕ, θ, and ψ, which are given in the 3-1-3 sequence, and are used to realize the orientation of the rear frame.

Additionally, the relative rotation angle between the front fork and rear frame is described by δ, the front wheel rotation angle with respect to the front frame assembly is given by β, and the rear wheel rotation angle with the rear frame assembly is denoted ξ.

As in [24] and [25], the reference configuration where ϕ, θ, ψ, and δ are zero is given where both wheels are in the xy-plane and the rear frame is oriented in the position shown in Figure 2.2. In this figure important points are also shown, which define positions of each of the components. The relative vectors between these points are defined in Appendix A.

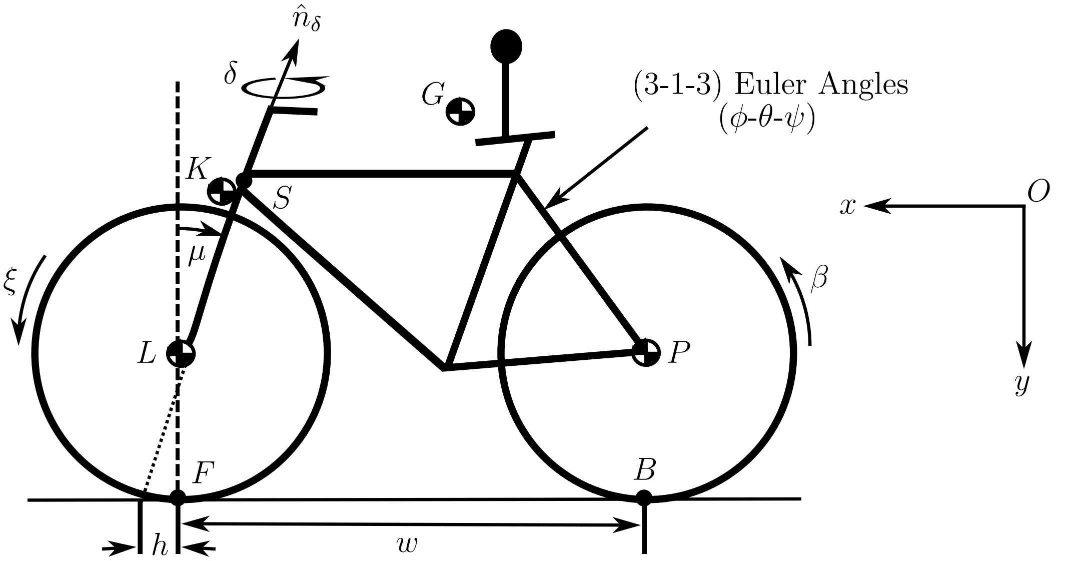


Figure 2.2: Important coordinates of the bicycle model shown in the reference configuration.

2.3 Kinetic and Potential Energies

To analyze the system using Lagrangian mechanics the potential energy and kinetic energy of each body are found. They are then combined to form the Lagrangian of the multi-body system. To find the kinetic and potential energies we follow the same general procedure as [24] and [25], with the difference coming in our treatment of rotational matrices. Although not explicitly written, all generalized coordinates are functions of time.

2.3.1 Frame and Rigid Rider

The first body being analysed is the rear frame and rigid rider. To begin, the rotation matrix $\mathbf{R}_{rf}$ is formed using body fixed rotations through the 3-1-3 Euler angles ϕ, θ, and ψ, resulting in

$$\mathbf{R}_{rf} = \mathbf{R}_z(\psi)\mathbf{R}_x(\theta)\mathbf{R}_z(\phi). \tag{2.5}$$

Recall that this procedure gives the local coordinates of a vector in the new frame when given global coordinates. However, we wish to express everything in terms of global coordinates too, so the reference configuration is treated as though it is given in local coordinates.

We convert to the desired coordinate system by multiplying the local vector by the matrix inverse, which in the case of a rotation matrix is equal to its transpose.

Now we find the position vector, $\mathbf{r}_P$, describing the current location of point P:

$$\mathbf{r}_P = \begin{bmatrix} x_P & y_P & z_P \end{bmatrix}^T. \tag{2.6}$$

Once completed it is possible to find the position vector the centre of mass of the rear frame and rigid rider, G. This position vector is found by using a rotation transformation on the relative position of point G with respect to point P in the reference position, termed $\mathbf{r}_{G/P,ref}$. Since the current relative position is now known and these points are on the same rigid body, the position vector for G is calculated by

$$\mathbf{r}_G = \mathbf{r}_P + \mathbf{R}_{rf}^T \mathbf{r}_{G/P,ref}. \tag{2.7}$$

It is then convenient to use computer software to differentiate the position vector to find the velocity vector $\mathbf{v}_G$.

Once $\mathbf{R}_{rf}$ is calculated it is possible to find the angular velocity tensor

$$\tilde{\mathbf{\Omega}}_{rf} = \dot{\mathbf{R}}_{rf}^T \mathbf{R}_{rf}. \tag{2.8}$$

Through analysis of this tensor it is possible to obtain the angular velocity of the rear frame through each of the major axes.

It is also important to convert the inertia tensor that is given in the reference configuration, $\tilde{\mathbf{I}}_{G,rf,ref}$, to reflect the current orientation of the bicycle. The calculation is

$$\tilde{\mathbf{I}}_{G,rf} = \mathbf{R}_{rf}^T \tilde{\mathbf{I}}_{G,rf,ref} \mathbf{R}_{rf}. \tag{2.9}$$

Once completing the previous steps, it is possible to find the angular momentum:

$$\mathbf{H}_G = \tilde{\mathbf{I}}_{G,rf} \boldsymbol{\omega}_{rf}. \tag{2.10}$$

The total kinetic energy of the rear frame is then

$$T_{rf} = \frac{1}{2} m_{rf} \mathbf{v}_G^T \mathbf{v}_G + \frac{1}{2} \boldsymbol{\omega}_{rf}^T \mathbf{H}_G, \tag{2.11}$$

where m_{rf} is the mass of the rear frame and rider. Finally the potential energy of the rear frame is

$$V_{rf} = m_{rf} g \mathbf{r}_G^T \hat{\mathbf{e}}_3, \tag{2.12}$$

where g is the gravity and $\hat{\mathbf{e}}_3$ is the unit vector in the direction of the z axis.

2.3.2 Rear Wheel

The rear wheel first goes through the same rotation as the rear frame, $\mathbf{R}_{rf}$. There is then an additional rotation of β along the local e_3 axis. The resulting rotation matrix is

$$\mathbf{R}_{rw} = \mathbf{R}_z(\beta)\mathbf{R}_{rf}. \tag{2.13}$$

Using the rotation matrix of the rear wheel, it is then possible to find the angular velocity tensor of the rear wheel,

$$\tilde{\mathbf{\Omega}}_{rw} = \dot{\mathbf{R}}_{rw}^T\mathbf{R}_{rw}. \tag{2.14}$$

The rear wheel inertia tensor at a given instant in time can be found by transforming the given inertia in the reference configuration:

$$\tilde{\mathbf{I}}_{P,rw} = \mathbf{R}_{rw}^T\tilde{\mathbf{I}}_{P,rw,ref}\mathbf{R}_{rw}. \tag{2.15}$$

Since the angular velocity tensor, $\tilde{\mathbf{\Omega}}_{rw}$, and rear wheel inertia, $\tilde{\mathbf{I}}_{P,rw}$, are known the angular momentum can be calculated by

$$\mathbf{H}_P = \tilde{\mathbf{I}}_{P,rw}\boldsymbol{\omega}_{rw}. \tag{2.16}$$

The position vector of the center of mass of the rear wheel has already been found as $\mathbf{r}_P$. After performing the differentiation of this vector, the total kinetic energy of the rear wheel is found to be

$$T_{rw} = \frac{1}{2}m_{rw}\mathbf{v}_P^T\mathbf{v}_P + \frac{1}{2}\boldsymbol{\omega}_{rw}^T\mathbf{H}_P, \tag{2.17}$$

where m_{rw} is the mass of the rear wheel.

The potential energy of the rear wheel can also be found and is

$$V_{rw} = m_{rw}g\mathbf{r}_P^T\hat{\mathbf{e}}_3. \tag{2.18}$$

2.3.3 Front Frame

The procedure for finding the kinetic and potential energy for the front frame is largely the same as the previous components. The process for finding the rotation matrix of the front frame again goes through the same rotation as the rear frame, R_{rf}, but then goes through a rotation δ about the steering axis of the bicycle. Given this information, the rotation matrix of the front frame is

$$\mathbf{R}_{ff} = \mathbf{R}_z\left[-\left(\mu+\frac{\pi}{2}\right)\right]^T \mathbf{R}_x(\delta)\mathbf{R}_z\left[-\left(\mu+\frac{\pi}{2}\right)\right]\mathbf{R}_{rf}. \tag{2.19}$$

A two-step process conveniently finds the position of the center of mass of the front frame, $\mathbf{r}_K$. First the position vector of a point, S, that is common to both the rear frame and front frame is found. This position vector is found by using the relative position vector from P to point S. Since the relative position vector is known in the reference configuration it must be rotated to match the current orientation, which results in the calculation

$$\mathbf{r}_S = \mathbf{r}_P + \mathbf{R}_{rf}\mathbf{r}_{S/P,ref}. \tag{2.20}$$

The second step is to use the new position information, $\mathbf{r}_S$, and rotate the known position vector $\mathbf{r}_{K/S,ref}$ to the current orientation to find the position of the center of mass for the fork:

$$\mathbf{r}_K = \mathbf{r}_S + \mathbf{R}_{ff}\mathbf{r}_{K/S,ref}. \tag{2.21}$$

Again the angular velocity tensor for the front frame can be found using the information contained in the associated rotation matrix;

$$\tilde{\mathbf{\Omega}}_{ff} = \dot{\mathbf{R}}_{ff}^T\mathbf{R}_{ff}. \tag{2.22}$$

To find the kinetic energy of the bicycle the inertia at the current orientation is also required. Again the inertia at the reference configuration, $\tilde{\mathbf{I}}_{K,ff,ref}$, is combined with the information contained in the rotation matrix, $\mathbf{R}_{ff}$, to solve

$$\tilde{\mathbf{I}}_{K,ff} = \mathbf{R}_{ff}^T\tilde{\mathbf{I}}_{K,ff,ref}\mathbf{R}_{ff}. \tag{2.23}$$

Now that the angular velocity and inertia tensors have been found it is possible to find the angular momentum,

$$\mathbf{H}_K = \tilde{\mathbf{I}}_{K,ff}\boldsymbol{\omega}_{ff}. \tag{2.24}$$

Again differentiating the position vector $\mathbf{r}_K$ is required, and the kinetic energy of the front fork can be obtained,

$$T_{ff} = \frac{1}{2}m_{ff}\mathbf{v}_K^T\mathbf{v}_K + \frac{1}{2}\boldsymbol{\omega}_{ff}^T\mathbf{H}_K, \tag{2.25}$$

where m_{ff} is the mass of the front frame.

The potential energy can also now be found as

$$V_{ff} = m_{ff}g\mathbf{r}_K^T\hat{\mathbf{e}}_3. \tag{2.26}$$

2.3.4 Front Wheel

The front wheel is the final body for which potential and kinetic energy must be found. The first step that is taken is again finding the rotation matrix. We again use body fixed axes and realizing that the front wheel goes through the same rotation as the front frame, with an additional rotation along the body fixed z axis of the front frame. The resulting rotation matrix is

$$\mathbf{R}_{fw} = \mathbf{R}_z(\xi)\mathbf{R}_{ff}. \tag{2.27}$$

The position of the centre of mass of the front wheel, L, is found by starting with point S, and rotating the relative position vector in the reference configuration, $\mathbf{r}_{L/S,ref}$, to the current orientation. The resulting vector is

$$\mathbf{r}_L = \mathbf{r}_S + \mathbf{R}_{ff}^T\mathbf{r}_{L/S,ref}. \tag{2.28}$$

The angular velocity tensor of the front wheel is found by

$$\tilde{\mathbf{\Omega}}_{fw} = \dot{\mathbf{R}}_{fw}^T\mathbf{R}_{fw}. \tag{2.29}$$

The inertia tensor in the reference configuration for the front wheel is then transformed to the current orientation by

$$\tilde{\mathbf{I}}_{L,fw} = \mathbf{R}_{fw}^T\tilde{\mathbf{I}}_{L,fw,ref}\mathbf{R}_{fw}. \tag{2.30}$$

Using the newly found inertia tensor the angular momentum of the front wheel can be found by

$$\mathbf{H}_L = \tilde{\mathbf{I}}_{L,fw}\boldsymbol{\omega}_{fw}. \tag{2.31}$$

After differentiating the position vector of the center of mass, it is then possible to calculate the kinetic energy,

$$T_{fw} = \frac{1}{2}m_{fw}\mathbf{v}_L^T\mathbf{v}_L + \frac{1}{2}\boldsymbol{\omega}_{fw}^T\mathbf{H}_L, \tag{2.32}$$

where m_{fw} is the mass of the front wheel.

Finally the potential energy of the front wheel is

$$V_{fw} = m_{fw}g\mathbf{r}_L^T\hat{\mathbf{e}}_3. \tag{2.33}$$

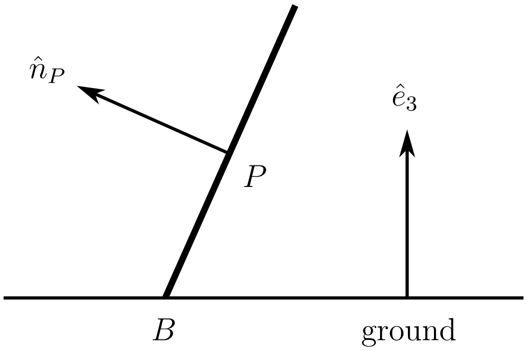


Figure 2.3: Rear wheel from behind the bicycle with important vectors for deriving the rolling constraints.

2.4 Constraints

Now that the potential energy and kinetic energy have been found for each of the rigid bodies in the system, the constraint equations are needed. These equations reduce the system to the appropriate degrees of freedom.

2.4.1 Rear Wheel

The first set of constraints enter due to the knife edge rolling constraint of the front wheel. The same method described by Basu-Mandal [24] is used, and as in [24] we show the important vectors needed to visualize this approach in Figure 2.3.

This method looks at the rear wheel from behind the wheel, along a plane that is parallel to the spin axis of the rear wheel and perpendicular to the global xz-plane. A vector, $\mathbf{n}_P$, is found that is parallel to the rotation axis of the rear wheel:

$$\mathbf{n}_P = \mathbf{R}_{rw}^T \hat{\mathbf{e}}_3. \tag{2.34}$$

The cross product of this vector and a unit vector along the global z axis produces a vector in the direction of travel of the wheel,

$$\mathbf{c}_{rw} = \mathbf{n}_P \times \hat{\mathbf{e}}_3. \tag{2.35}$$

It should be noted that this method does not work when the bicycle is flat on the ground, but the this case is of little consequence as the control system stabilizing the bicycle would be deemed unacceptable before this condition is reached.

The position vector from the instantaneous point of contact of the wheel, B, to the centre of mass of the rear wheel, P, is found by

$$\mathbf{r}_{P/B} = \frac{r_{rw}}{|\mathbf{c}_{rw}|}\mathbf{c}_{rw} \times \mathbf{n}_P. \tag{2.36}$$

The velocity of point P is then

$$\mathbf{v}_P = \mathbf{v}_B + \boldsymbol{\omega}_{rw} \times \mathbf{r}_{P/B}, \tag{2.37}$$

where $\mathbf{v}_B = 0$, due to the rolling constraint.

The result is a differentiated holonomic constraint and two nonholonomic constraints. As indicated by Basu-Mandal [24], the holonomic constraint is

$$z_P = r_{rw}\sin\theta, \tag{2.38}$$

and its differentiated form can be substituted into the equations of motion to reduce the degrees of freedom. However, the non-holonomic constraints must be retained.

2.4.2 Front Wheel

The same steps are taken for the knife edge rolling constraint of the front wheel. First the vector along the axis of the front wheel is found,

$$\mathbf{n}_L = \mathbf{R}_{fw}^T\hat{\mathbf{e}}_3. \tag{2.39}$$

Next, a vector in the direction of travel of the front wheel is obtained as

$$\mathbf{c}_{fw} = \mathbf{n}_L \times \hat{\mathbf{e}}_3. \tag{2.40}$$

After finding these vectors, the position vector from the contact point of the front wheel with the ground, F, to the center of mass, L, is found to be

$$\mathbf{r}_{L/F} = \frac{r_{fw}}{|\mathbf{c}_{fw}|}\mathbf{c}_{fw} \times \mathbf{n}_L. \tag{2.41}$$

By noting that $\mathbf{v}_F = 0$, the velocity of the center of mass of the front wheel is

$$\mathbf{v}_L = \mathbf{v}_F + \boldsymbol{\omega}_{fw} \times \mathbf{r}_{L/F}. \tag{2.42}$$

Again the result is three constraints. As mentioned by Basu-Mandal [24], the vertical component is theoretically a differentiated holonomic constraint, but it is very complex, so it is retained in its differential form. Provided that the initial position of the bicycle is checked to ensure that both wheels of the bicycle touch the ground as expected the differentiated constraint is acceptable as the constraint on velocity ensures that the holonomic constraint remains satisfied.

2.5 Equations of Motion

The holonomic constraint found previously has been substituted in to reduce the number of generalized coordinates to eight. For convenience the generalized coordinates are represented by a vector

$$\mathbf{q} = \begin{bmatrix} x_P & y_P & \phi & \theta & \psi & \delta & \beta & \xi \end{bmatrix}^T. \tag{2.43}$$

We assume that actuators are mounted to the bicycle in such a way that a torque about the lean axis, a torque about the steering axis, and a torque about the rear wheel rotation axis can be generated. From this assumption the nonlinear input forces written vector form are

$$\mathbf{u} = \begin{bmatrix} 0 & 0 & 0 & \tau_\theta & 0 & \tau_\delta & \tau_{\beta_r} & 0 \end{bmatrix}^T, \tag{2.44}$$

where τ_θ is the lean torque, τ_δ is the steer torque, and τ_β is the rear wheel torque. Other general forces can also be added, but careful consideration of the assumptions and constraints used must be made.

To begin writing the Lagrangian and solving for the equations of motion, it is necessary to find the total kinetic energy,

$$T_{tot} = T_{rf} + T_{rw} + T_{ff} + T_{fw}. \tag{2.45}$$

Likewise, the total potential energy is found through

$$V_{tot} = V_{rf} + V_{rw} + V_{ff} + V_{fw}. \tag{2.46}$$

This information is then used to write the Lagrangian,

$$\mathcal{L} = T_{tot} - V_{tot}. \tag{2.47}$$

The reaction forces are of no consequence for the purpose of control, so the remaining five constraint equations can be introduced through the use of Lagrange multipliers. We have four nonholonomic constraints and a holonomic constraint expressed as a constraint on velocity. Each constraint can be can be written in the form

$$\varphi_i(\mathbf{q}, \dot{\mathbf{q}}, t) = 0, \; i = 1, 2, ..., 5. \tag{2.48}$$

This information can then be incorporated and the Euler-Lagrange equation can be written describing the motion of the bicycle,

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j}\right) - \frac{\partial \mathcal{L}}{\partial q_j} = \sum_{i=1}^{5} \frac{\partial \varphi_i}{\partial q_j} \varrho_i + u_j, \; j = 1, 2, ..., 8. \tag{2.49}$$

For convenience we have written φ_i instead of $\varphi_i(\mathbf{q}, \dot{\mathbf{q}}, t)$, and u_j is force j in $\mathbf{u}$. Finally ϱ_i is the i^{th} element in the Lagrange multiplier vector

$$\boldsymbol{\varrho} = \begin{bmatrix} \varrho_1 & \varrho_2 & \varrho_3 & \varrho_4 & \varrho_5 \end{bmatrix}^T. \tag{2.50}$$

The full set of nonlinear equations is not given here, due to their length. The complexity of the equations also makes direct analysis impractical, and thus provides little insight to the behaviour of the bicycle model.

2.5.1 Initial Conditions

There are seven geometric degrees of freedom and eight generalized coordinates. As in [25] we also choose the dependent coordinate to be ψ, which corresponds to the pitch of the bicycle in the reference configuration discussed in Section 3.1. It is of note that the dependent variable could be eliminated by finding the holonomic constraint on the front wheel discussed in Section 2.4.2. Instead, once all of the independent generalized coordinates are set appropriately the initial value of ψ is found numerically. This computation is completed by determining the position vector for the point where the front wheel contacts the ground,

$$\mathbf{r}_L = \mathbf{r}_P + \mathbf{R}_{rf}\mathbf{r}_{S/P,ref} + \mathbf{R}_{ff}\mathbf{r}_{L/S,ref} - \mathbf{r}_{L/F}. \tag{2.51}$$

Here the final values of $\mathbf{R}_{rf}$, $\mathbf{R}_{ff}$ and $\mathbf{r}_{L/F}$ are dependent on $\psi(0)$. There are infinite solutions when the rear wheel is lying flat on the ground. In all other cases there are two solutions, which are shown in Figure 2.4. Here the configuration in Figure 2.4(a) is correct, while the one in Figure 2.4(b) is inadmissible. Multiple solutions occur because the lowest point on the front wheel is considered F when calculating $r_{L/F}$ in (2.41).

From results of numerical simulations, the initial conditions affecting ψ are θ, ϕ and δ. We choose to define the initial conditions of these angles such that $\phi(0) \in [0,\ 2\pi]$ radians, $\theta(0) \in [0,\ \pi]$ radians, and $\delta(0) \in [-\pi,\ \pi]$ radians. By allowing $\psi(0) \in [0,\ 2\pi]$, we can express all possible orientations of the rear frame. By solving for $\psi(0)$ using numerical methods, we find two solutions. The first is $\psi(0) \approx 0$ radians or $\psi(0) \approx 2\pi$ radians and the second is $\psi(0) \approx \pi$ radians. It is found that $\psi(0) \approx \pi$ radians typically corresponds to the correct configuration as shown in Figure 2.4(a), which matches intuition when examining the reference configuration. Also noted by Basu-Mandal [25] there are geometries where the front wheel never touches the ground, but these are not discussed.

In velocity, we know there are three degrees of freedom, and eight derivatives for generalized coordinates in $\dot{\mathbf{q}}$. From knowledge of the linearized model presented in Section 3.1

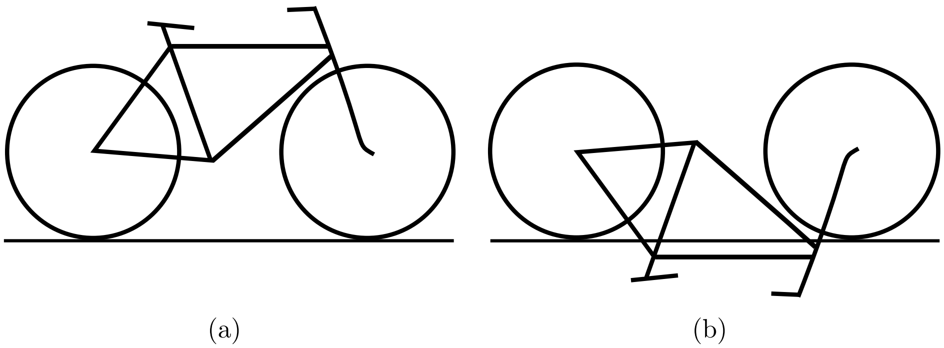


Figure 2.4: Two possible initial bicycle configurations when finding $\psi(0)$.

the independent variables should be chosen as $\dot{\theta}$, $\dot{\delta}$ and $\dot{\beta}$. Once these initial conditions are set to any rate that is reasonable, given the assumptions of the Whipple model, the five constraint equations in Section 2.4 can then be used to explicitly find the remaining values.

2.5.2 Solving Equations of Motion

The dynamics of the bicycle have now been fully defined. A fundamental way of explicitly solving the equations of motion is discussed by Yoon [51]. To use this approach we rearrange (2.49) as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{U}\boldsymbol{\varrho} = \mathbf{G}(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{u}, \tag{2.52}$$

where $\mathbf{M}(\mathbf{q})$ is an inertia matrix that is a function of the generalized coordinates, $\mathbf{U}$ is the matrix of coefficients for the Lagrange multipliers, and $\mathbf{G}(\mathbf{q}, \dot{\mathbf{q}}, t)$ is the remainder of the equations of motion excluding the generalized forces. The velocity constraints from (2.48) are also rewritten as

$$\boldsymbol{\Phi}(\mathbf{q}, \dot{\mathbf{q}}, t) = 0. \tag{2.53}$$

The constraints can then be differentiated:

$$\dot{\boldsymbol{\Phi}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) = 0. \tag{2.54}$$

Here, (2.52) and (2.54) represent thirteen equations that can be solved explicitly, by first solving for $\boldsymbol{\varrho}$, and then substituting back in to find the value $\ddot{\mathbf{q}}$.

It is also important to note that this procedure represents the constraints as acceleration constraints instead of velocity or position constraints, so initial conditions must be chosen carefully. There is also a potential for drift in the constraints due to inaccuracies in solving the system of equations. However, drift was not observed to enter significantly into the system, and methods to counteract this phenomenon are not discussed.

Chapter 3

Inner-Loop Control

The inner loop is designed with a linear compensator to control the lean and steer angles of the bicycle. A high-level view of how the inner loop fits into the overall control strategy is highlighted in Figure 3.1. A linear approach is taken since the model developed previously in Chapter 2 is too complicated to work with directly: a full display of the final equations requires many pages. In contrast, the linearized model, given in Section 3.1, can be readily used for controller design purposes. The resulting linearized model is parameterized with forward speed. This system is analyzed in Section 3.2 to show how the eigenvalues vary with forward speed. This analysis reveals that there are five regions characterized by a unique response of the bicycle to any input, four of which are predicted by the linear model. Each of the four regions exhibits a high cross-channel coupling between lean and steer. The final analysis is to compare the linear and nonlinear models to show how well the linear model approximates the nonlinear model.

The final sections, Sections 3.3 and 3.4, design the linear controller for the lean and steer control and the speed control of the bicycle, respectively.

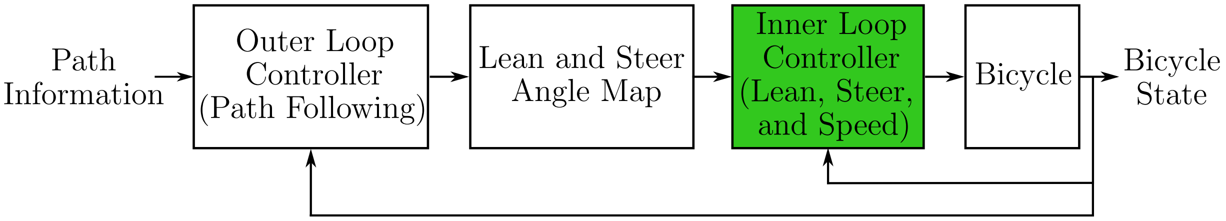


Figure 3.1: Outline of approach with inner loop control highlighted.

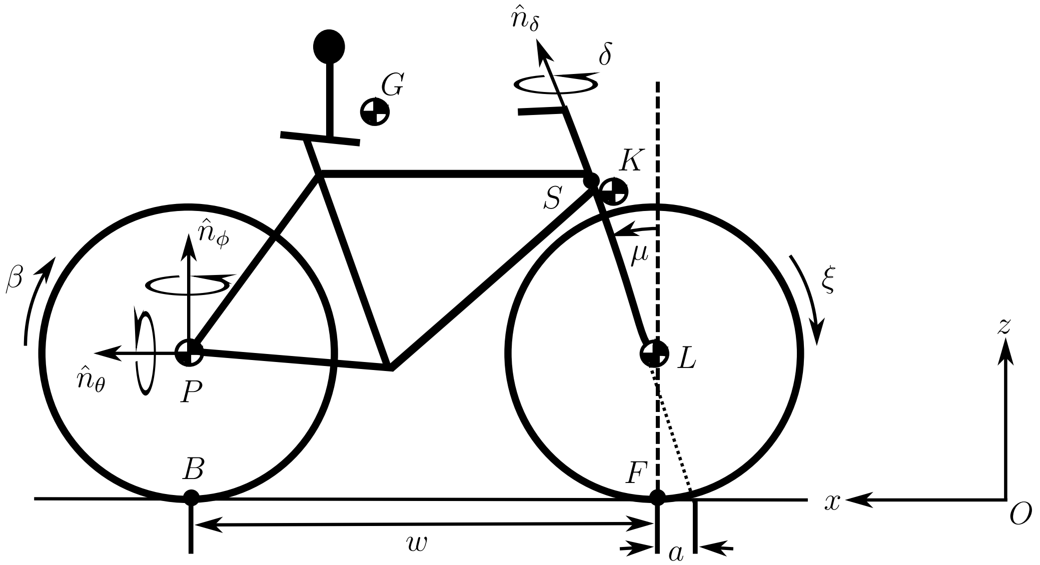


Figure 3.2: Operating trajectory for the linearziation of the bicycle.

3.1 Linearized Model

For linearization of the model we adopt a similar approach to that taken by Basu-Mandal [25]. To begin the process, the operating trajectory is set as the bicycle positioned in an upright configuration, such that the spin axis of both wheels is parallel to the xy-plane, and it is travelling straight ahead along the negative x axis at a forward speed v, as shown in Figure 3.2. At this operating trajectory τ_θ, τ_δ, and τ_β are all zero, and as a function of time the generalized coordinates are

$$\mathbf{q} = \begin{bmatrix} -vt & 0 & 0 & \frac{\pi}{2} & \pi & 0 & \frac{vt}{r_{rw}} & \frac{vt}{r_{fw}} \end{bmatrix}^T. \tag{3.1}$$

From here, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ can be found by differentiating $\mathbf{q}$ with respect to time. Then substituting these values and the parameters values from Appendix A into the equations of motion allows explicit solving of the Lagrange multipliers. Proceeding along these lines shows that for this set of $\mathbf{q}$ the Lagrange multiplier values are

$$\boldsymbol{\varrho} = \begin{bmatrix} 0 & 0 & 0 & 0 & 309 \end{bmatrix}^T. \tag{3.2}$$

Now that the operating trajectory is defined, we take a Taylor series expansion of the eight equations of motion and five constraint equations about this trajectory, and keep only the first-order terms. The result of this calculation is given in (3.3)–(3.15) below, where Δ is used to signify perturbations from the nominal trajectory:

$$
\begin{aligned}
95\Delta\ddot{x}_P - 48.2\Delta\ddot{\psi} - \Delta\varrho_1 - \Delta\varrho_3 &= 0 && (3.3)\\
95\Delta\ddot{y}_P - 32.2\Delta\ddot{\phi} - 48.2\Delta\ddot{\theta} - 0.201\Delta\ddot{\delta} - \Delta\varrho_2 - \Delta\varrho_4 &= 0 && (3.4)\\
-1.6v\Delta\dot{\theta} - 0.247v\Delta\dot{\delta} - 32.2\Delta\ddot{y}_P + 17.1\Delta\ddot{\phi} + 17.7\Delta\ddot{\theta} & \\
+0.33\Delta\ddot{\delta} + 1.02\Delta\varrho_4 &= 0 && (3.5)\\
-799\Delta\theta - 25.5\Delta\delta + 1.6v\Delta\dot{\phi} + 0.761v\Delta\dot{\delta} - 48.2\Delta\ddot{y}_P & \\
+17.7\Delta\ddot{\phi} + 35.7\Delta\ddot{\theta} + 0.091\Delta\ddot{\delta} - 0.35\Delta\varrho_2 - 0.35\Delta\varrho_4 &= \Delta\tau_\theta && (3.6)\\
-472\Delta\psi - 48.2\Delta\ddot{x}_P + 51.8\Delta\ddot{\psi} + 0.28\Delta\ddot{\beta} + 0.28\Delta\ddot{\xi} & \\
-0.35\Delta\varrho_1 - 0.35\Delta\varrho_3 + 1.02\Delta\varrho_5 &= 0 && (3.7)\\
-25.5\Delta\theta - 7.88\Delta\delta + 0.247v\Delta\dot{\phi} - 0.761v\Delta\dot{\theta} - 0.2\Delta\ddot{y}_P & \\
+0.33\Delta\ddot{\phi} + 0.091\Delta\ddot{\theta} + 0.154\Delta\ddot{\delta} - 0.0761\Delta\varrho_4 &= \Delta\tau_\delta && (3.8)\\
0.28\Delta\ddot{\psi} + 0.28\Delta\ddot{\beta} - 0.35\Delta\varrho_1 &= \Delta\tau_\beta && (3.9)\\
0.28\Delta\ddot{\psi} + 0.28\Delta\ddot{\xi} - 0.35\Delta\varrho_3 &= 0 && (3.10)\\
\Delta\dot{x}_P + 0.35\Delta\dot{\psi} + 0.35\Delta\dot{\beta} &= 0 && (3.11)\\
v\Delta\phi + \Delta\dot{y}_P + 0.35\Delta\dot{\theta} &= 0 && (3.12)\\
\Delta\dot{x}_P + 0.35\Delta\dot{\psi} + 0.35\Delta\dot{\xi} &= 0 && (3.13)\\
v\Delta\phi + 0.951v\Delta\delta + \Delta\dot{y}_P - 1.02\Delta\dot{\phi} + 0.35\Delta\dot{\theta} + 0.0761\Delta\dot{\delta} &= 0 && (3.14)\\
-1.02\Delta\dot{\psi} &= 0. && (3.15)
\end{aligned}
$$

The constraint equations, (3.11)–(3.15), are used to solve for $\Delta\dot{x}_P$, $\Delta\dot{y}_P$, $\Delta\dot{\phi}$, $\Delta\dot{\psi}$, and $\Delta\dot{\xi}$. The results are then differentiated since the equations of motion, (3.3)–(3.10), are not dependent on these forward speed terms, but are dependent on the corresponding accelerations. It is then possible to solve explicitly for $\Delta\ddot{x}_P$, $\Delta\ddot{y}_P$, $\Delta\ddot{\phi}$, $\Delta\ddot{\psi}$, $\Delta\ddot{\xi}$, $\Delta\boldsymbol{\lambda}$, $\Delta\ddot{\beta}$, $\Delta\ddot{\theta}$, and $\Delta\ddot{\delta}$.

The purpose of the inner loop is to provide stability and control the lean and steer of the bicycle. We also wish to ensure that the bicycle remains at the linearized forward speed so we are concerned with $\Delta\ddot{\theta}$, $\Delta\ddot{\delta}$, and $\Delta\ddot{\beta}$. The equations below provide the linearized steer, lean and rear wheel rotation dynamics. Here the values are perturbations from the operating point, but the Δ has been dropped for convenience.

$$\begin{aligned}\ddot{\theta} &= 9.52\theta - 0.109v\dot{\theta} - (0.567 + 0.9v^2)\delta \\ &\quad -0.33v\dot{\delta} + 0.0159\tau_\theta - 0.123\tau_\delta \end{aligned} \tag{3.16}$$

$$\begin{aligned}\ddot{\delta} &= 11.5\theta + 3.79v\dot{\theta} + (30.8 - 1.9v^2)\delta \\ &\quad -3.11v\dot{\delta} - 0.123\tau_\theta + 4.31\tau_\delta \end{aligned} \tag{3.17}$$

$$\ddot{\beta} = 0.08198\tau_\beta. \tag{3.18}$$

This linearization matches other linear representations of the Whipple model. Specifically, a coordinate transformation was made to the parameters in Appendix A and then substituted into the equations of Meijaard [11]. When substituted the linear equations derived here were found to match the equations to at least twelve decimal places.

3.2 Model Analysis

For now we assume that the bicycle forward speed is constant and use a state-space representation of the linearized bicycle dynamics which is readily obtained from (3.16) and (3.17). The state vector is chosen as

$$\boldsymbol{\chi} = \begin{bmatrix} \theta & \dot{\theta} & \delta & \dot{\delta} \end{bmatrix}^T, \tag{3.19}$$

and the input vector is

$$\mathbf{u}_L = \begin{bmatrix} \tau_\theta & \tau_\delta \end{bmatrix}^T. \tag{3.20}$$

The state-space representation of the parameterized linear time invariant system is then

$$\begin{aligned} \dot{\boldsymbol{\chi}} &= \mathbf{A}(v)\boldsymbol{\chi} + \mathbf{B}\mathbf{u}_L \\ \boldsymbol{\eta}_L &= \mathbf{C}\boldsymbol{\chi}, \end{aligned} \tag{3.21}$$

where

$$\begin{aligned} \mathbf{A}(v) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 9.52 & -0.109v & -0.567 - 0.9v^2 & -0.36v \\ 0 & 0 & 0 & 1 \\ 11.5 & 3.79v & 30.8 - 1.9v^2 & -3.11v \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} 0 & 0 \\ 0.0159 & -0.123 \\ 0 & 0 \\ -0.123 & 4.31 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \end{aligned} \tag{3.22}$$

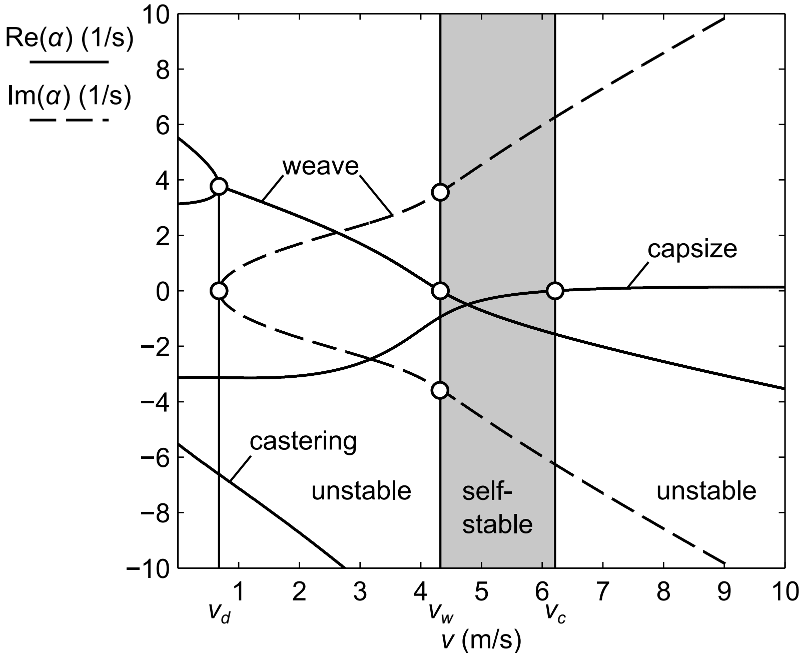


Figure 3.3: Eigenvalues of the linearized bicycle changing with forward speed.

3.2.1 System Eigenvalues

Upon inspection, the system is highly dependent on forward speed. Indeed the eigenvalues change drastically as forward speed is changed. These four eigenvalues are distinct except for special sets of bicycle parameters, and they correspond to well known eigenmodes and have traditional names as described by Meijaard et al [11]. For completeness we also discuss them here.

The eigenmodes for the bicycle parameters in Appendix A are shown in Figure 3.3. This figure is based on a figure similar in [11]. Note that we focus on the forward speed range of 0 m/s to 10 m/s, as outside this range the knife edge rolling constraints assumed when deriving the equations of motion for the model are likely to break down. The most dramatic changes in the eigenvalues of the model also occur in this region, and across this forward speed span the bicycle exhibits all four distinctly different types of motion predicted by the eigenvalues.

From analysis of the eigenvalues at low forward speeds the bicycle is unstable due to two real eigenvalues corresponding to the weave modes. These real eigenvalues are representative of the bicycle falling over in an inverted pendulum like response. Due to

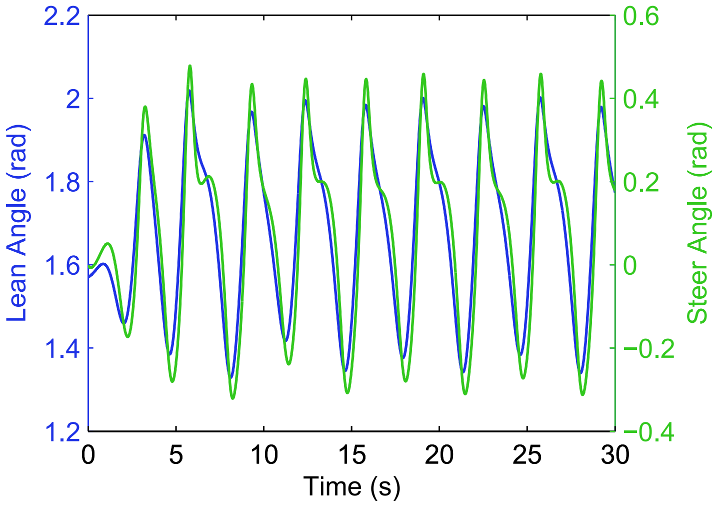


Figure 3.4: Response of the nonlinear bicycle model with initial forward forward speed of 3.5 m/s to an impulse on lean at $t = 0$.

the location of the angle of the hinge connection between the front and rear frame, as well as its location with respect to the ground contact points and centres of mass, a constant signal or disturbance for τ_θ or τ_δ causes the lean and steer to experience exponential growth in opposite directions.

As the bicycle increases in forward speed to $v_d \approx 0.679$ m/s, these eigenvalues become complex and lead to oscillatory eigenmodes. This oscillatory response is representative of the interaction of the steer and lean angles about a given direction. The front fork assembly initially steers away from the lean for $v < v_d$, but then it begins to rotate such that the bicycle steers into the lean. This action causes the bicycle to lean in the opposite direction, and results in a sinusoidal steer and lean angle. Aström et al. [15] explain that the physical reason the that the bicycle steers into the lean is that the lean causes a torque on the front frame mainly due to the contact with the ground causing the front frame to rotate in the direction of the lean. The centrifugal force acting on the rear frame then causes the lean angle to rotate in the opposite direction, so both lean and steer angles exhibit a periodic response where the steer angle lags the lean. The location of the eigenvalues indicates that this action is not stable until the bicycle forward speed is greater than $v_w \approx 4.33$ m/s. However, simulation on the nonlinear system reveals that at a forward speed of $v_l \approx 3.38$ m/s, the limit cycle speed, given a sufficiently small perturbation from the upright equilibrium, there is a limit cycle. Figure 3.4 shows the response of the system within this region.

At forward speed v_w, the weave speed, the interaction causing the limit cycle becomes stable and for a set of velocities the bicycle is self stable, meaning that without any external action the bicycle travels straight ahead without falling over.

As forward speed continues to increase, the bicycle again becomes unstable at a forward speed $v_c \approx 6.22$ m/s, the capsize speed. Here the capsize mode is the driving cause for instability. This mode is dominated by lean and when it is unstable the bicycle increasingly leans in one direction causing the bicycle to steer in that direction. Through simulation it is found that the front frame does not steer into the lean enough to correct it until after it is too late. The bicycle thus falls over and due to a small, slowly increasing steer angle it spirals inward until hitting the ground.

As discussed by Meijaard [11] the final eigenvalue is always stable and corresponds to the castor mode. This mode is dominated by steering and is stable as a result of the forward motion of the bicycle forcing the front wheel to line up with the rear frame.

Figure 3.5 gives another demonstration of how the eigenvalues vary with forward speed, and better demonstrates that as forward speed increases the weave modes continue to be stable, with a faster settling time but becoming increasingly oscillatory. With increasing forward speed the capsize mode remains real and just in the right half plane (RHP) and the caster mode also remains real and stable.

3.2.2 Frequency Response

To help determine the type of control that should be used, the Bode plots for the lean and steer angles in each region are examined in Figure 3.6. In all of the Bode plots in this figure when each input is applied individually it is seen to have approximately the same magnitude response on each of the outputs. Another observation is that τ_δ has a much greater effect on both θ and δ than τ_θ. This difference in response is partially due to the relatively small inertia of the front frame compared to the rear frame. The magnitude plots thus indicate that there are significant cross-channel interactions, and any input on τ_θ is likely to be significantly greater than τ_δ.

Using the knowledge of the eigenvalues from Section 3.2.1, it is also apparent from Figure 3.6 that, at all of the forward speeds studied, there is a non-minimum phase zero in the transfer function from both τ_θ and τ_δ to δ. In fact this is true for all velocities in the range 0 m/s $< v <$ 10 m/s. Additionally, looking at the bode plots in Figure 3.6(a) there are also RHP zeros in the response from both inputs to θ. The presence of these zeros indicates that there will be bandwidth limitations imposed on the system if output feedback control is used. This fact motivates the use of state feedback.

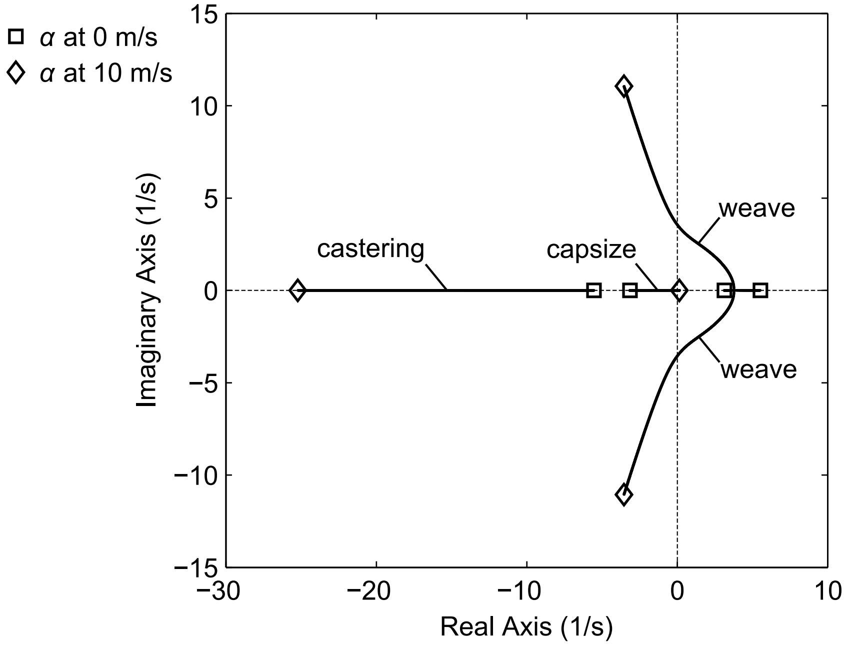
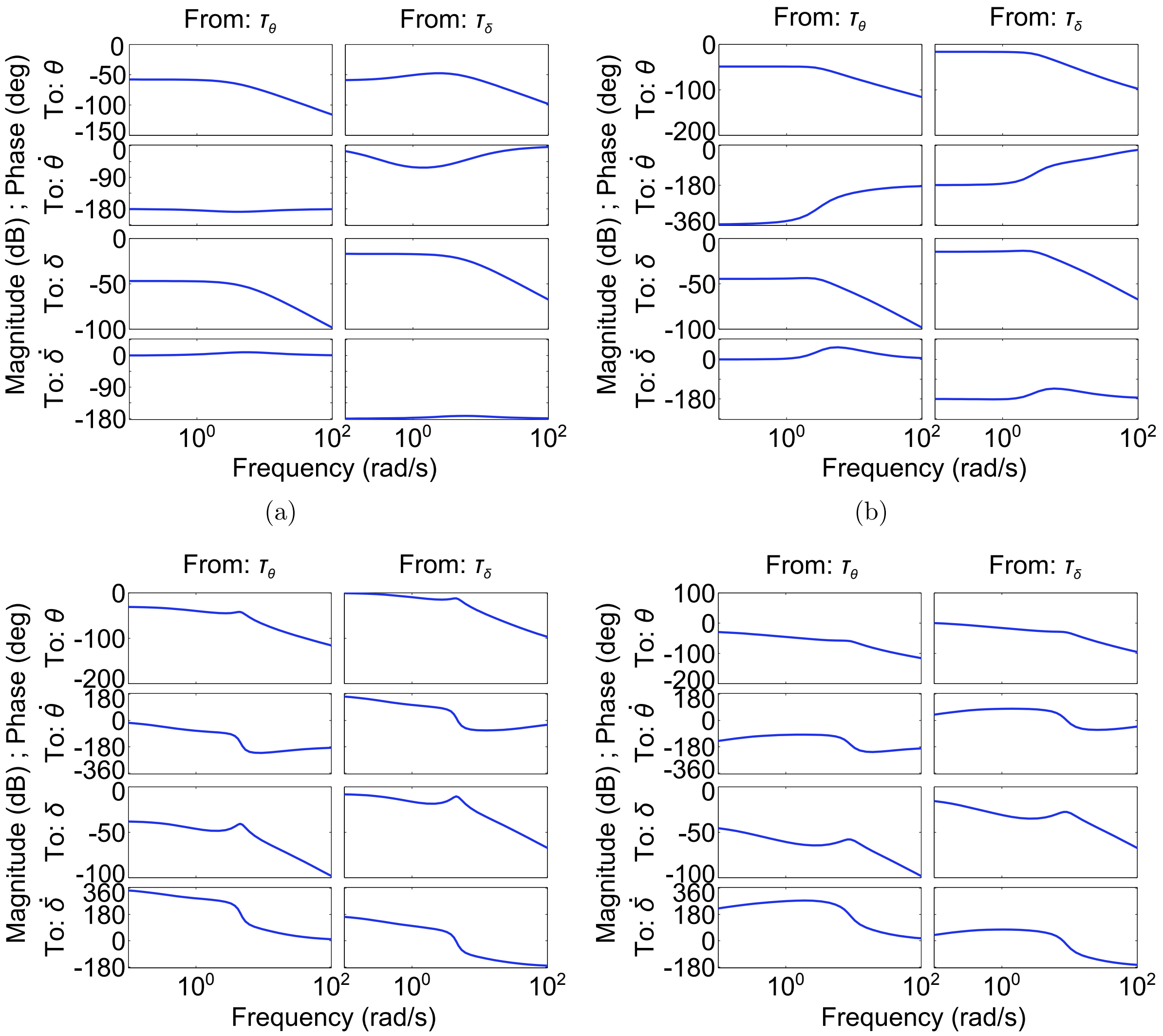


Figure 3.5: Complex plane plot of eigenvalues of the bicycle system as forward speed changes.



(a)

(b)

(c)

(d)

Figure 3.6: Bode plot of states θ, $\dot{\theta}$, δ, and $\dot{\delta}$ from inputs τ_θ and τ_δ at 0.5 m/s, 3 m/s, 5 m/s, and 8 m/s in (a), (b), (c), and (d), respectively.

3.2.3 Comparison Between Nonlinear and Linear Models

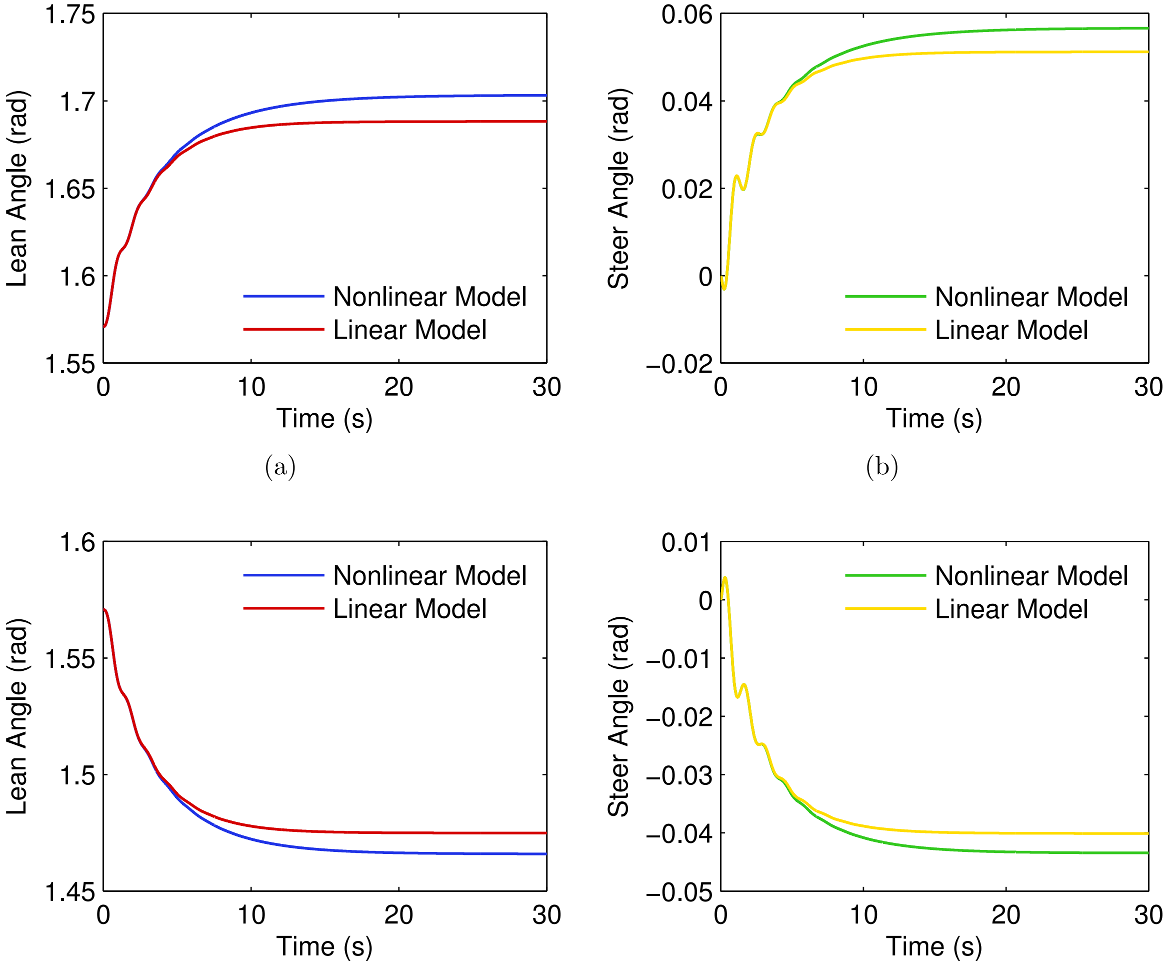
We have seen in Section 3.2.1 that the bicycle has dynamics that change dramatically with forward speed, and in all forward speed regions other than $v_w < v < v_c$ it is unstable. Figure 3.7 shows that, as expected, for a sufficiently small input the linearized model provides a good approximation of the nonlinear dynamics.

However, as one might expect, even in the self-stable region for sufficiently large inputs or initial conditions the bicycle falls over when using the nonlinear model. Figure 3.8 illustrates when the input torque is large enough to cause the bicycle to escape the region of attraction for the upright equilibrium at a forward speed of 5 m/s. Here, as in all future demonstrations, the nonlinear simulation is terminated when the bicycle comes within $\frac{1}{9}\pi$ radians of hitting the ground or the front frame does a revolution of π radians, which corresponds its complete reversal with respect to the rear frame. This tendency to fall over is due to the nonlinear dynamics, and the inability of the linearized model to capture the region of attraction of the upright equilibrium. It is difficult to determine the region where the linearized model retains accuracy for the unstable regions, since they must be stabilized through control first. A high level of agreement between the linear and nonlinear model may just be indicative of a controller that is robust to the system nonlinearities. We highlight this fact to show the importance of testing any controller designed from the linearized model on the non-linear model, and appropriately saturating both θ and δ.

Another important consideration that is not captured by the linear model is that when a steer or lean torque is applied to the bicycle, $\dot{\beta}$ is altered, and thus the forward speed at which point B tracks across the ground has changed from the expected forward speed for the path-following controller in Chapter 4. In Figure 3.9 this change is shown for a bicycle initially travelling at a forward speed of 5 m/s and the τ_θ and τ_δ steps from Figure 3.7. The change is small, and both the front and rear wheels gain angular velocity, which reduces after the step is removed. However, after applying the control in Section 3.3 without any compensation for speed, it was found that the average speed would change significantly over time. A drop in forward speed is a problem for both the inner-loop controller and outer-loop controller. This issue is addressed in Section 3.4, with a simple proportional gain compensator.

3.3 Lean and Steer Control

The linear model (3.21) is used to develop a linear control scheme. We assume that both a lean torque and a steer torque are generated to control the bicycle. In practice the steer



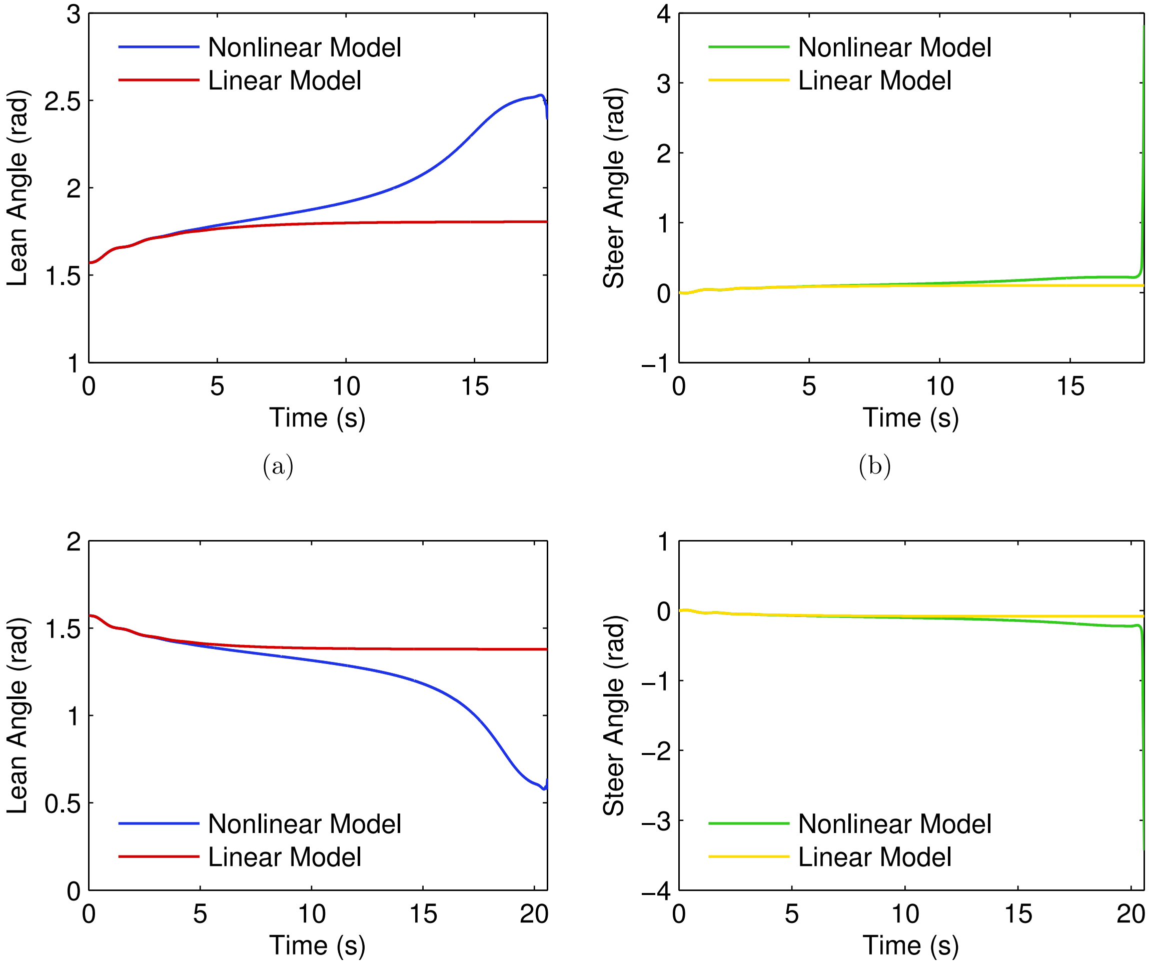
(a)

(b)

(c)

(d)

Figure 3.7: Comparison of the nonlinear bicycle model with the linearized model lean angle in (a) and steer angle in (b) for a lean torque step of 4 Nm at 5 m/s. Likewise the response for a steer torque step of 0.1 Nm on lean angle in (c) and steer angle in (d).



(a)

(b)

(c)

(d)

Figure 3.8: Comparison of the nonlinear bicycle model with the linearized model lean angle in (a) and steer angle in (b) for a lean torque step of 8 Nm at 5 m/s. Likewise the response for a steer torque step of 0.2 Nm on lean angle in (c) and steer angle in (d).

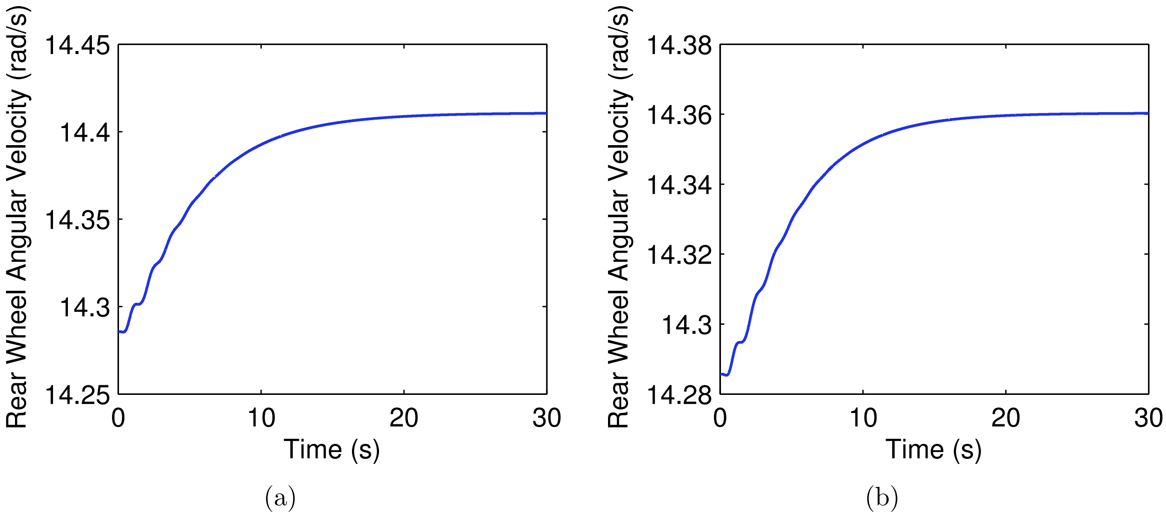


Figure 3.9: $\dot{\beta}$ variation on a bicycle initially moving at 5 m/s after applying a a 4 Nm lean torque in (a) and a 0.1 Nm steer torque in (b).

torque can be generated directly using a motor connected to the front frame; however, the lean torque is not as straight forward, and could be generated by a gyroscope or flywheel. In demonstrations of the inner-loop control it is assumed there is no limit on these torques, so depending on the motor capabilities the controller may need more tuning.

To design the inner-loop controller, Linear Quadratic Regulator (LQR) methods [52] are chosen, under the assumption that all of the states in (3.21) are measured. This method is used since the cross-channel interaction is seen to be significant, and the system exhibits a non-minimum phase response. Using full-state feedback simplifies design, with LQR being easily tunable based on performance criteria, and the method does not suffer from the performance limitations imposed by the RHP zeros. However, even with no disturbance, LQR provides no guarantee of zero steady-state error and the linearized model for which the controller is designed is in error everywhere except the linearized operating trajectory. To overcome this issue the plant state is augmented with integrators to ensure asymptotic steady-state tracking for step lean and steer reference trajectories. We define

$$\boldsymbol{\zeta} = \begin{bmatrix} \zeta_\theta & \zeta_\delta \end{bmatrix}^T, \quad \mathbf{r} = \begin{bmatrix} r_\theta & r_\delta \end{bmatrix}^T, \tag{3.23}$$

where ζ_θ, ζ_δ are the integrator states on the lean and steer respectively, and r_θ, r_δ are the reference trajectories for lean and steer respectively.

The new state-space representation of the open-loop augmented bicycle is

$$\begin{bmatrix}\dot{\boldsymbol{\chi}} \\ \dot{\boldsymbol{\zeta}}\end{bmatrix} = \begin{bmatrix}\mathbf{A}(v) & \mathbf{0} \\ -\mathbf{C} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{\chi} \\ \boldsymbol{\zeta}\end{bmatrix} + \begin{bmatrix}\mathbf{B} \\ \mathbf{0}\end{bmatrix}\mathbf{u}_L + \begin{bmatrix}\mathbf{0} \\ \mathbf{I}\end{bmatrix}\mathbf{r}, \tag{3.24}$$

which, for convenience, is compactly,

$$\dot{\bar{\boldsymbol{\chi}}} = \bar{\mathbf{A}}\bar{\boldsymbol{\chi}} + \bar{\mathbf{B}}\mathbf{u}_L + \bar{\mathbf{B}}_r. \tag{3.25}$$

The open-loop system is then checked for controllability. The rank of the controllability matrix,

$$\bar{\boldsymbol{W}} = \begin{bmatrix}\bar{\mathbf{B}} & \bar{\mathbf{A}}\bar{\mathbf{B}} & \bar{\mathbf{A}}^2\bar{\mathbf{B}} & \bar{\mathbf{A}}^3\bar{\mathbf{B}} & \bar{\mathbf{A}}^4\bar{\mathbf{B}} & \bar{\mathbf{A}}^5\bar{\mathbf{B}}\end{bmatrix}, \tag{3.26}$$

is six, meaning the system is controllable. Now it is possible to compute the control gains using the LQR algorithm. We assume an infinite time horizon, and minimize the cost function

$$J = \int_0^\infty \left(\bar{\boldsymbol{\chi}}^T\bar{\mathbf{Q}}\bar{\boldsymbol{\chi}} + \mathbf{u}_L^T\bar{\mathbf{S}}\mathbf{u}_L\right)dt. \tag{3.27}$$

The optimal control law is a state-feedback controller, $\mathbf{u}_L = -\mathbf{F}\bar{\boldsymbol{\chi}}$, where $\mathbf{F}$ is found through a two-step procedure. The first is to solve the algebraic Riccati equation,

$$\bar{\mathbf{A}}^T\bar{\mathbf{P}} + \bar{\mathbf{P}}\bar{\mathbf{A}} - \bar{\mathbf{P}}\bar{\mathbf{B}}\bar{\mathbf{S}}^{-1}\bar{\mathbf{B}}^T\bar{\mathbf{P}} + \bar{\mathbf{Q}} = 0, \tag{3.28}$$

where $\bar{\mathbf{P}}$ is a unique positive-semidefinite matrix. Then find $\mathbf{F}$ directly by

$$\mathbf{F} = \bar{\mathbf{S}}^{-1}\bar{\mathbf{B}}^T\bar{\mathbf{P}}. \tag{3.29}$$

For convenience in analysis $\mathbf{F}$ is broken into $\mathbf{F}_1$ and $\mathbf{F}_2$, which are the feedback gains for the original state vector, $\boldsymbol{\chi}$, and the augmented integrator channels $\boldsymbol{\zeta}$, respectively. The overall linear equations for the inner loop can be represented as

$$\dot{\bar{\boldsymbol{\chi}}} = \begin{bmatrix}\bar{\mathbf{A}} - \bar{\mathbf{B}}\mathbf{F}_1 & -\bar{\mathbf{B}}\mathbf{F}_2 \\ -\mathbf{C} & \mathbf{0}\end{bmatrix}\bar{\boldsymbol{\chi}} + \begin{bmatrix}\mathbf{0} \\ \mathbf{I}\end{bmatrix}\mathbf{r}. \tag{3.30}$$

This control architecture is shown on the nonlinear system in Figure 3.10.

3.4 Forward Speed Control

As shown in Section 3.2.3 the linear model fails to capture the change to the forward speed of the bicycle when a steer or lean torque is applied away from the operating point. The

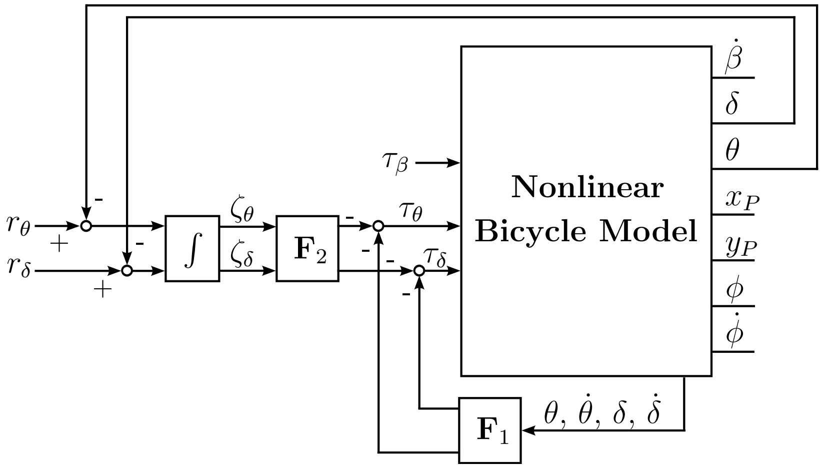


Figure 3.10: Detailed block diagram of the lean and steer controller.

movement away from the operating point is seen to affect $\dot{\beta}$ and $\dot{\xi}$. The change to $\dot{\beta}$ is of more interest since the it is more easily related to the forward speed, v, of point P on the rear frame. The location of this point is important when designing the path-following controller in Chapter 4 and the rear frame has the majority of the mass in the system. To begin designing a simple classic controller to overcome the variation in forward speed, recall the linearized relation between τ_β and $\ddot{\beta}$ in (3.18). By taking the Laplace transformation and integrating we find that

$$\dot{\beta} = \frac{0.082}{s}\tau_\beta. \tag{3.31}$$

Since the transfer function contains an integrator we use a simple proportional gain compensator. This compensator is easily implemented and the closed system has a transfer function of

$$\frac{\dot{\beta}}{r_{\dot{\beta}}} = \frac{0.082F_{\dot{\beta}}}{s + 0.082F_{\dot{\beta}}} \tag{3.32}$$

where $r_{\dot{\beta}}$ is the reference angular velocity for $\dot{\beta}$ and $F_{\dot{\beta}}$ is the gain of the proportional compensator. To not significantly affect the response of the LQR controller we use high bandwidth control that is at least four times faster than the LQR controller. This fast controller is chosen since a change in forward speed from the set point has a significant impact on the stability of the response for θ and δ.

With the addition of this forward speed controller, the control architecture is modified as shown in Figure 3.11

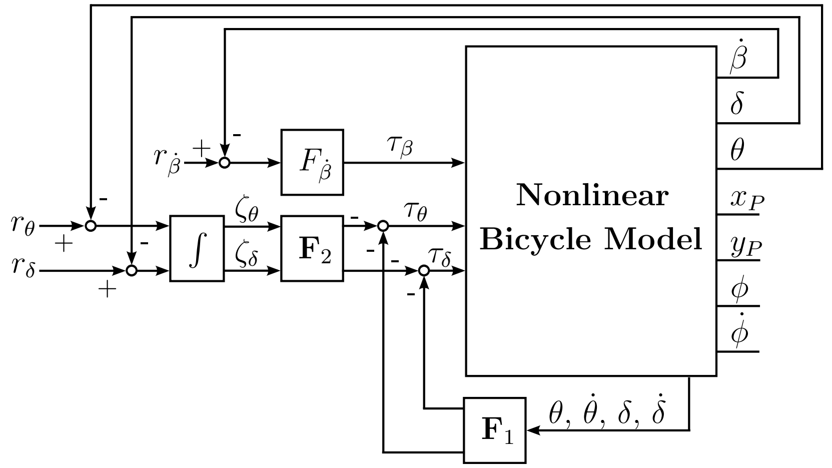


Figure 3.11: Detailed block diagram of the inner loop.

Chapter 4

Path Following

The control structure that provides path-following logic for the bicycle is represented by the outer-loop in Figure 4.1. The purpose of the control architecture is to provide stability and perfect asymptotic tracking of a constant curvature path for the contact point between the rear wheel of the bicycle and the ground.

The process for designing the outer loop begins with designing a lean and steer angle map in Section 4.1. The purpose of this map is to convert the commanded yaw rate for the bicycle into an appropriate value of lean angle reference and steer angle reference for the inner loop. This section also develops an approximation for the maximum curvature of a path that is traceable by the bicycle.

A step on curvature is easily related to a yaw-rate step. This information motivates system identification on the map and inner-loop to find a linear system that approximates the yaw-rate response of the lean and steer angle map and inner-loop to a yaw-rate input. Section 4.2 contains this system identification and proposes a yaw-rate controller.

To successfully converge to a path, the distance to the path should be zero and the yaw

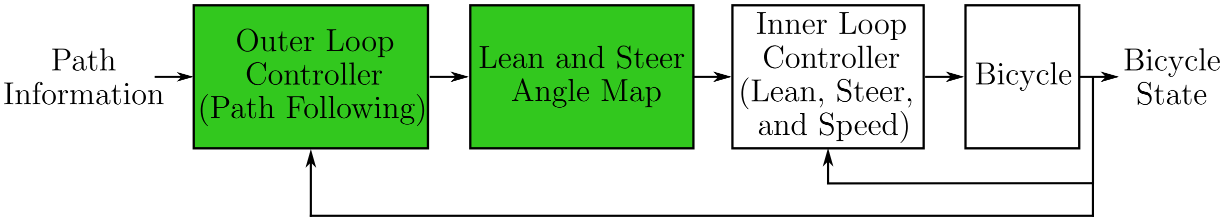

Path Information
Outer Loop Controller (Path Following)
Lean and Steer Angle Map
Inner Loop Controller (Lean, Steer, and Speed)
Bicycle
Bicycle State

Figure 4.1: Outline of approach with outer loop control highlighted

angle of the path and bicycle should also be the same for all time after convergence occurs. The yaw angle is simply the integral of the yaw rate, but the distance calculation is not as simple, and Section 4.3 discusses how to calculate the distance from the rear wheel contact point with the ground and the closest point on the path.

Finally in Section 4.4 the architecture and controllers for handling errors in yaw angle and distance are introduced. An equivalent linear system is then analyzed and we show that using the control method developed, the rear wheel contact point of the bicycle will asymptotically track constant curvature paths.

4.1 Mapping Yaw Rate to Steer Angle

To begin designing a controller for path following, we first relate δ to a specific bicycle yaw rate, $\dot{\gamma}$. This map appears as part of the system architecture as shown in Figure 4.2. To find the required δ for a given, $\dot{\gamma}$ equations developed by Meijaard [11] are adopted, where a completely symbolic relationship is derived by ignoring everything except first-order terms as part of an ad hoc linearization. For convenience we do omit writing Δ in this derivation. By assuming that γ is small and measured counter clockwise from the x-axis, the relationship between the lateral position of the front and rear wheel contact points, respectively y_F and y_B, is

$$y_F = y_B + w\gamma - h\delta\cos\mu, \tag{4.1}$$

where w is the wheelbase, h is the distance the front wheel contact point trails the point where the steering axis intersects the ground plane, and μ is the angle of the steering axis as shown in Figure 2.2. Differentiating this expression with respect to time results in the useful equation

$$\dot{y}_F = \dot{y}_B + w\dot{\gamma} + h\dot{\delta}\cos\mu. \tag{4.2}$$

The linearized y component of the non-holonomic rolling constraint for the rear wheel contact point is given as

$$\dot{y}_B = v\gamma. \tag{4.3}$$

Likewise a similar expression for the linearized y component of the front wheel rolling constraint is derived for point F (see Figure 2.2), and found to be

$$\dot{y}_F = v(\gamma + \delta\cos\mu). \tag{4.4}$$

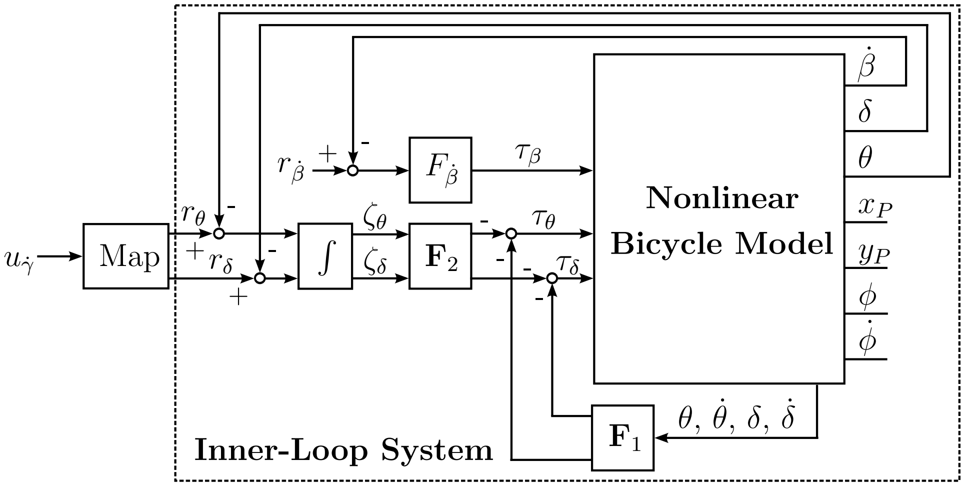


Figure 4.2: Detailed block diagram showing the map from yaw rate to steering angle and lean angle.

By substituting the right hand side (RHS) of (4.2) into the front wheel rolling constraint (4.4) and subtracting the rear wheel rolling constraint (4.3), then isolating for the yaw rate, we find

$$\dot\gamma = \frac{v\delta + h\dot\delta}{w}\cos\mu. \tag{4.5}$$

Recall that the main function of this map is to convert a $\dot\gamma$ and to an appropriate δ. We take the Laplace transform of (4.5) and treat δ as the output, with an input $\dot\gamma$, resulting in the transfer function

$$\frac{\delta(s)}{\dot\gamma(s)} = \frac{w}{(v + hs)\cos\mu}. \tag{4.6}$$

Notice here that the DC gain is unaffected by h, and more importantly h is typically a very small positive value for a standard bicycle. We do not consider the case where h is large or negative, so the map is the equivalent of a low-pass filter, with a high cut off frequency. Also when the bicycle is tracking a straight line or circle in steady state $\dot\gamma$ is constant, and the map is simplified by ignoring the $h\dot\delta$ term in (4.5). Note that the linearizations used to arrive at this map are performed about $\delta = 0$ and $\dot\gamma = 0$, so there is no bias required when integrating the map into the nonlinear system.

Converting to the reference signals used in Figure 4.2, the final map is thus

$$r_\delta = \frac{u_{\dot\gamma} w}{v\cos\mu}, \tag{4.7}$$

where r_δ and $u_{\dot{\gamma}}$ are respectively the reference value of δ for the inner-loop controller and the input $\dot{\gamma}$ from the control system providing path following logic.

There is a need to saturate r_δ so that the bicycle does not leave the region for which the LQR controller developed in Section 3.3 provides stability. This saturation imposes a maximum curvature that can be tracked by the bicycle. Note that the control architecture is designed to track a constant curvature, and the curvature of a circle is defined as $\kappa = R^{-1}$, where R is the radius of the circle. To begin deriving the relation between the maximum value of δ and the maximum κ, it is important to realize that due to the angle of the steering axis the effective steering angle is reduced. Aström [15] identifies that at small angles this relationship is

$$\delta_f = \delta \cos \mu, \tag{4.8}$$

where δ_f is the effective steering angle. Figure 4.3 shows an overhead view of the bicycle travelling around a circular path. The rear frame is always tangential to the path, so the effective steering angle can be related to the radius of the curve being followed by the point B by

$$\tan |\delta_f| = \frac{w}{R}. \tag{4.9}$$

It follows that an approximation of the maximum κ is given as

$$\max \kappa = \frac{\tan(\max |r_\delta| \cdot \cos \mu)}{w}. \tag{4.10}$$

Note that we consider only $|\mu| < \frac{1}{2}\pi$ radians, so $\cos \mu > 0$. The curvature in (4.10) is an approximate limit of what can be tracked when the bicycle starts exactly on the path, and is turning to stay on the path. If the bicycle starts off the path, it is necessary for any path curvature to be limited further.

4.1.1 Choosing an Appropriate Lean Angle

There are a number of possible choices for θ. As long as the reference for θ is chosen small enough that the bicycle does not tip over, θ has only minor influence on the steady-state yaw rate, which is evident in (4.5). There are two approaches that we consider. The first is to continually drive

$$r_\theta = \frac{\pi}{2}, \tag{4.11}$$

while controlling δ to the desired value given by (4.7).

The second approach is to attempt to minimize the lean torque at steady state. This is done by examining the linearized equations for $\ddot{\theta}$ and $\ddot{\delta}$ in (3.16) and (3.17), respectively,

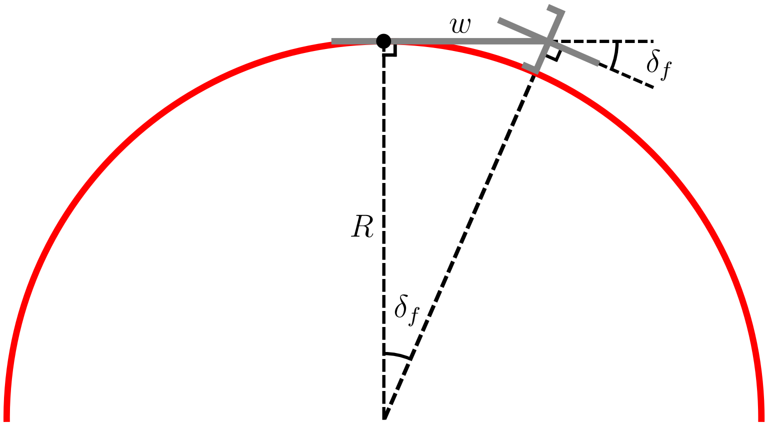


Figure 4.3: Overhead view of a bicycle following a circular path.

and recognizing that $\dot{\theta}$, $\dot{\delta}$, $\ddot{\theta}$ and $\ddot{\delta}$ are zero in steady state. We omit the Δ for convenience, and substitute this information, as well as $\tau_\theta = 0$ in an attempt to minimize the lean torque, which results in

$$\begin{aligned} 0 &= 9.52\theta - (0.567 + 0.9v^2)\delta - 0.123\tau_\delta \quad (4.12)\\ 0 &= 11.5\theta + (30.8 - 1.9v^2)\delta + 4.31\tau_\delta. \quad (4.13)\end{aligned}$$

By eliminating τ_δ from the equations, adding the appropriate bias, and converting to the input and output signals in Figure 4.2 the map is

$$r_\theta = \frac{\pi}{2} + 0.0969 r_\delta v^2 - 0.0319 r_\delta, \quad (4.14)$$

where r_δ is again determined by (4.7) resulting in

$$r_\theta = \frac{\pi}{2} + \frac{u_{\dot{\gamma}} w}{v \cos\mu}(0.0969v^2 - 0.0319). \quad (4.15)$$

The advantage of the first approach is that $\ddot{\theta}$ and $\ddot{\delta}$ are tightly coupled as discussed in Section 3.2.2, so trying to change both at the same time results in significant cross channel interaction, and a reduction in the quality of the transient response. By keeping θ close to $\frac{1}{2}\pi$ radians, the system is also closer to the linearized operating trajectory (3.1), so all approximations are more exact. The second approach of minimizing τ_δ has the advantage that it requires less input in steady state; however, at higher v we are also more limited in the allowable values for r_δ as the map scales with v^2. The second approach would also provide a more natural feel to a rider.

4.2 Controlling Yaw Rate

Controlling the yaw rate is an important step as the yaw rate can be related to curvature by

$$|\dot{\gamma}| = |v|\,\kappa, \tag{4.16}$$

which implies that tracking a constant curvature is equivalent to tracking a step on $\dot{\gamma}$. To control $\dot{\gamma}$ we start by finding an appropriate measure. Observing that when the bicycle parameters found in Appendix A are substituted into (4.5) the result is numerically equivalent to that found by setting (3.12) equal to (3.14) and isolating $\Delta\phi$. Thus when the bicycle is near the operating trajectory (3.1), $\dot{\phi}$ provides a good estimate for $\dot{\gamma}$. Referring to Figure 3.2, $\phi = 0$ holds when the bicycle is travelling along the negative x-axis. This is an offset of π from how we defined γ, thus we use the approximations $\gamma = \phi + \pi$ and $\dot{\gamma} = \dot{\theta}$. By substituting $\theta = \delta = \frac{1}{2}\pi$ radians in as initial conditions of the bicycle and comparing the true value of γ to $\phi + \pi$ the deviation is a mere 0.072 radians, which confirms that even for larger angles this approximation is still reasonable.

To control the yaw rate, we wish replace the system in Figure 4.2 with an equivalent single input single output (SISO) model. We accomplish this goal using least squares to determine an appropriate model, varying the order of the approximation until an acceptable compromise between accuracy and model complexity is reached. To begin discussion of this method consider the system modelled by a general transfer function of the form

$$P_{\dot{\gamma}}(s) = \frac{\dot{\gamma}^{mod}(s)}{u_{\dot{\gamma}}(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0}, \tag{4.17}$$

where $\dot{\gamma}^{mod}(s)$ is the output of the transfer function model and $u_{\dot{\gamma}}(s)$ is the input. In our analysis we use a step input and analyze the response of the output, and choose $m \leq 1$ and $n \leq 2$. We also do not use direct continuous-time identification, as this involves derivative operators on both input and output. Since there is some noise in the output of the system, the derivative operator only serves to amplify it. One method to overcome this discussed by Rao et al. [53], is to first discretize the plant using what they term an unconventional discrete time (UDT) operator, perform system identification, and then relate the discrete plant back to the original continuous model. These UDT operator discretization methods are actually first-order approximations of the exact mapping from continuous to discrete time, $z = e^{sT}$, where T is the sample time. The UDT used here is the trapezoidal difference approximation, which corresponds to a bilinear transformation.

Converting (4.17) to discrete time using the bilinear transformation results in

$$\begin{aligned} P_{\dot{\gamma}}[z] &= P_{\dot{\gamma}}(s)\big|_{s=\frac{2(z-1)}{T(z+1)}} \\ &= \frac{g_n z^n + g_{n-1} z^{n-1} + \ldots + g_1 z + g_0}{z^n + f_{n-1} z^{n-1} + \ldots + f_1 z + f_0}. \end{aligned} \tag{4.18}$$

By cross multiplying, taking the inverse z-transform, and time shifting, the time domain representation is found to be

$$\begin{aligned} \dot{\gamma}^{mod}[t_{k+n}] = &- f_{n-1}\dot{\gamma}^{mod}[t_{k+n-1}] - f_{n-2}\dot{\gamma}^{mod}[t_{k+n-2}] - \ldots - f_1\dot{\gamma}^{mod}[t_{k-n+1}] - f_0\dot{\gamma}^{mod}[t_{k-n}] \\ &+ g_n u_{\dot{\gamma}}[t_{k+n}] + g_{n-1} u_{\dot{\gamma}}[t_{k+n-1}] + \ldots + g_1 u_{\dot{\gamma}}[t_{k-n+1}] + g_0 u_{\dot{\gamma}}[t_{k-n}], \end{aligned} \tag{4.19}$$

where t_k is the time at sample k.

Note the choice to approximate a non-linear system with a linear model means there is inherently error in the system. We express this error as

$$\begin{aligned} \mathbf{e} &= \dot{\boldsymbol{\gamma}} - \dot{\boldsymbol{\gamma}}^{mod} \\ &= \dot{\boldsymbol{\gamma}} - \bar{\boldsymbol{\Phi}}\mathbf{a}, \end{aligned} \tag{4.20}$$

where $\mathbf{e}$ is a vector representing the error between the actual output $\dot{\boldsymbol{\gamma}}$ and the output of the model $\dot{\boldsymbol{\gamma}}^{mod}$ at select times. To perform the analysis $\dot{\boldsymbol{\gamma}}^{mod}$ is broken into a known matrix $\bar{\boldsymbol{\Phi}}$ and an unknown vector of constants, $\mathbf{a}$. Assuming there are N samples and that $\mathbf{a} \in \mathbb{R}^{(2n+1)}$, then $\mathbf{e} \in \mathbb{R}^N$, $\dot{\boldsymbol{\gamma}} \in \mathbb{R}^N$, $\dot{\boldsymbol{\gamma}}^{mod} \in \mathbb{R}^N$, and $\bar{\boldsymbol{\Phi}} \in \mathbb{R}^{N\text{x}(2n+1)}$. To perform least squares we want to minimize the cost function $J(\mathbf{a}) = \mathbf{e}^T\mathbf{e}$. The solution to this problem is well known as

$$\mathbf{a} = (\bar{\boldsymbol{\Phi}}^T\bar{\boldsymbol{\Phi}})^{-1}\bar{\boldsymbol{\Phi}}^T\dot{\boldsymbol{\gamma}}. \tag{4.21}$$

Substituting the discrete-time system (4.19) into the error equation (4.20) results in the expression

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \\ \vdots \\ \dot{\gamma}_N \end{bmatrix}$$

$$- \begin{bmatrix} \dot{\gamma}_0^{mod} & \dot{\gamma}_{-1}^{mod} & \cdots & \dot{\gamma}_{-n}^{mod} & u_{\dot{\gamma},1} & u_{\dot{\gamma},0} & \cdots & u_{\dot{\gamma},-n} \\ \dot{\gamma}_1^{mod} & \dot{\gamma}_0^{mod} & \cdots & \dot{\gamma}_{1-n}^{mod} & u_{\dot{\gamma},2} & u_{\dot{\gamma},1} & \cdots & u_{\dot{\gamma},1-n} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \dot{\gamma}_{N-1}^{mod} & \dot{\gamma}_{N-2}^{mod} & \cdots & \dot{\gamma}_{N-n}^{mod} & u_{\dot{\gamma},N} & u_{\dot{\gamma},N-1} & \cdots & u_{\dot{\gamma},N-n} \end{bmatrix} \cdot \begin{bmatrix} -f_{n-1} \\ -f_{n-2} \\ \vdots \\ -f_0 \\ g_n \\ g_{n-1} \\ \vdots \\ g_0 \end{bmatrix}, \tag{4.22}$$

where for space we have used a subscript in conjunction with e, $\dot{\gamma}$ and $u_{\dot{\gamma}}$ to indicate the sample number, for instance $e_1 = e[t_1]$. We continue to design using continuous-time methods so the discrete-time system is converted back to continuous time and a solution for the unknowns in (4.17) is readily obtained.

In simulation testing, this system identification method has proved satisfactory, and there has not been enough system noise from numerical inaccuracies solving the ordinary differential equations of motion for the bicycle to cause a poor fit. When attempting implementation on a physical system, the output measurements may require preprocessing before performing system identification.

After obtaining the unknowns in (4.17), it is possible to design a controller for $\dot{\gamma}$. We adopt a one-degree-of-freedom classical-control architecture, and term the controller $C_{\dot{\gamma}}(s)$. Recall that we estimate the value of $\dot{\gamma}$ by $\dot{\phi}$. The detailed implementation is shown in Figure 4.4, and the closed-loop system response can be approximated by the transfer function

$$H_{\dot{\gamma}}(s) = \frac{\dot{\gamma}(s)}{r_{\dot{\gamma}}(s)} = \frac{P_{\dot{\gamma}}(s)C_{\dot{\gamma}}(s)}{1 + P_{\dot{\gamma}}(s)C_{\dot{\gamma}}(s)}. \tag{4.23}$$

For the path-following design developed, it is suggested that $C_{\dot{\gamma}}(s)$ is designed to contain an integrator for path-following performance. The only other condition placed on the controller is that it must be stabilizing, but the parameters of $C_{\dot{\gamma}}(s)$ are not chosen until remaining components used for path tracking control are developed in Section 4.4.

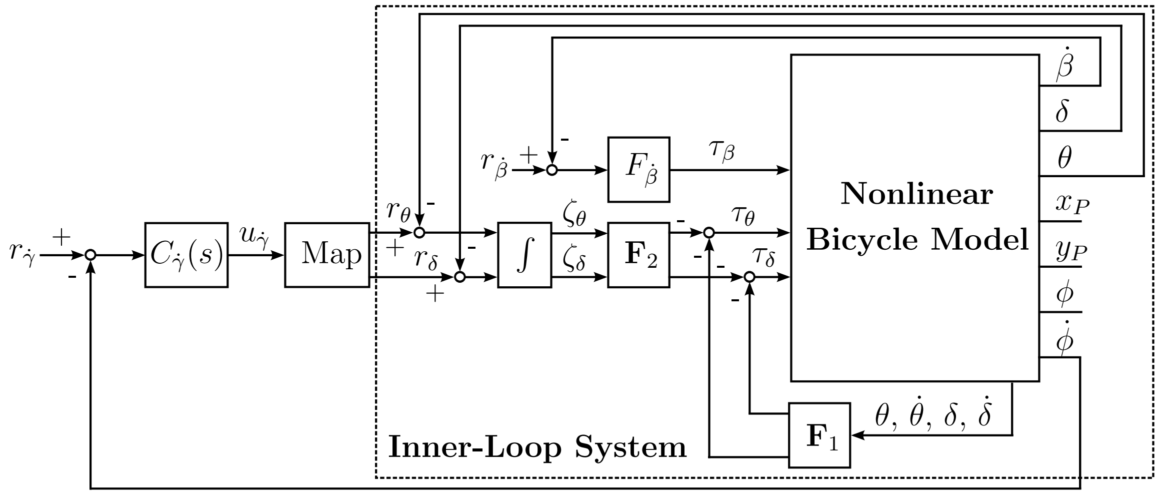


Figure 4.4: Detailed block diagram showing the yaw rate control.

4.3 Calculating Distance to Path

To calculate the distance between the bicycle and the path, d_ρ, the instantaneous rate that the bicycle is diverging from the path is found and then integrated. To accomplish this, expressions for the velocity that the rear wheel contact point moves across the ground, $\dot{x}_B$ and $\dot{y}_B$, are required. Realizing that γ defines only the heading of the bicycle and can be arbitrarily large without changing the lean and steer dynamics of the bicycle, the expressions for the velocity in Cartesian coordinates are

$$\begin{aligned} \dot{x}_B &= v\cos\gamma, \\ \dot{y}_B &= v\sin\gamma. \end{aligned} \tag{4.24}$$

Similar equations are found for the path, which has an instantaneous yaw angle λ. Let

$$\mathbf{r}_\rho = \begin{bmatrix} x_\rho & y_\rho & 0 \end{bmatrix}^T \tag{4.25}$$

be the position vector of the closest point on the path, ρ. An important consideration is that the instantaneous velocity of ρ is not equal in magnitude to the velocity of the bicycle since the bicycle moves at an angle $\gamma - \lambda$ with respect to the path. Ignoring higher-order terms, the velocity of ρ can be found by projecting the bicycle velocity vector onto a line tangent to the path. Solving for this vector gives the speed of ρ as

$$v_\rho = v\cos(\gamma - \lambda). \tag{4.26}$$

The expressions for the velocity of the components of ρ are then

$$\begin{aligned} \dot{x}_\rho &= v\cos(\gamma - \lambda)\cos\lambda, \\ \dot{y}_\rho &= v\cos(\gamma - \lambda)\sin\lambda. \end{aligned} \tag{4.27}$$

The rate that the bicycle diverges from the path is the Euclidean distance between the vectors:

$$\dot{d}_\rho = \sqrt{(\dot{x}_B - \dot{x}_\rho)^2 + (\dot{y}_B - \dot{y}_\rho)^2}. \tag{4.28}$$

Substituting (4.24) and (4.27) into (4.28) results in

$$\dot{d}_\rho = v\sin(\gamma - \lambda). \tag{4.29}$$

We define both γ and δ to be a counter-clockwise angle from the x-axis, so this results in the convention that when γ is larger than λ the change in distance is positive, with the reverse being negative.

By assuming that $\gamma - \lambda$ is small, a first-order Taylor series approximation can be used to approximate (4.29). This is integrated, ignoring initial conditions, and the Laplace transform is taken, giving

$$\Delta d_\rho = \frac{v\Delta\left(\gamma - \lambda\right)}{s}, \tag{4.30}$$

which is the desired linearized expression for the distance of the bicycle to the path.

Care must be taken when γ and λ are defined modulo 2π. When one of γ or λ is marginally smaller than 2π and the other is larger due to the modulo operation the value of $\gamma - \lambda$ is large while the heading is approximately the same. This issue can be overcome by checking the quadrant of γ and λ. If one is in the first quadrant and the other in the fourth, it is advisable to add 2π radians to the angle that is in the first quadrant without performing a modulo operation to calculate the relative angle.

To calculate the distance between the path and the contact point for the actual nonlinear bicycle model it is necessary to find the position vector of B,

$$\mathbf{r}_B = \begin{bmatrix} x_B & y_B & 0 \end{bmatrix}^T. \tag{4.31}$$

The output of the equations of motion contains values for x_P and y_P, and using (2.38) we find z_P, which gives us all components of $\mathbf{r}_P$. The value of $\mathbf{r}_B$ can be found from

$$\mathbf{r}_B = \mathbf{r}_P - \mathbf{r}_{P/B}, \tag{4.32}$$

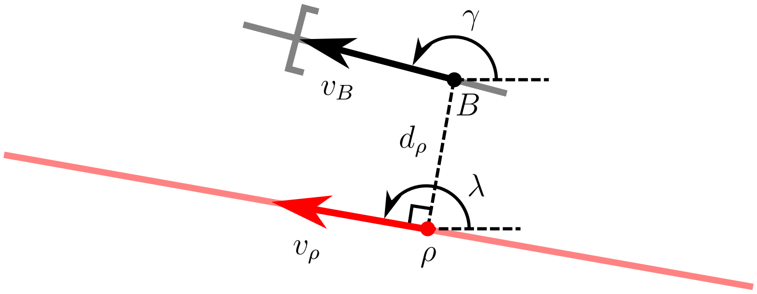


Figure 4.5: Overhead view of the bicycle and a linear path with important measures labelled.

where $\mathbf{r}_{P/B}$ is found using (2.36). In the case of a path that is a line, ρ can then be found by projecting B onto the line. For a path that is a circle, polar coordinates originating at the center of the circle can be used. The polar angle of B is also the polar angle of ρ, and the radius of the circle is known, so ρ is readily found, and the relative position vector is converted to global Cartesian coordinates. In the case that B is in the centre of the circle the bicycle is the same distance from all points on the edge. In this case ρ can be chosen as the point on the path where $|\gamma - \lambda|$ is minimized. An example for a linear path with the important points and measures labelled is shown in Figure 4.5. The example is equivalent for a circular path if the linear path is assumed to be a tangent line of the circular path where ρ is a point on the path.

4.4 Tracking Path

For point B on the bicycle to track a path of constant curvature with zero steady-state error it is obviously necessary to also track the x and y coordinates perfectly. This implies that we desire $d_\rho = 0$. The relationship between relative heading of the bicycle with respect to the path and d_ρ is given in (4.30), where it is clear that $d_\rho \neq 0$ when the yaw angle of the bicycle and that of the path are not equal, which motivates control of γ. Equation (4.30) also ignores the initial value of d_ρ, so if there is no control applied directly to d_ρ there is no guarantee that the bicycle is ever actually on the path.

A block diagram of the architecture designed to achieve path following is presented in Figure 4.6. Recall that we approximate the value of γ by $\phi + \pi$, since we do not have a direct measure of γ. It is shown that the value of ϕ is compared to $\lambda - \pi$, where it is passed into a controller, $C_\gamma(s)$, and likewise the value of d_ρ is found by calculating the Euclidean

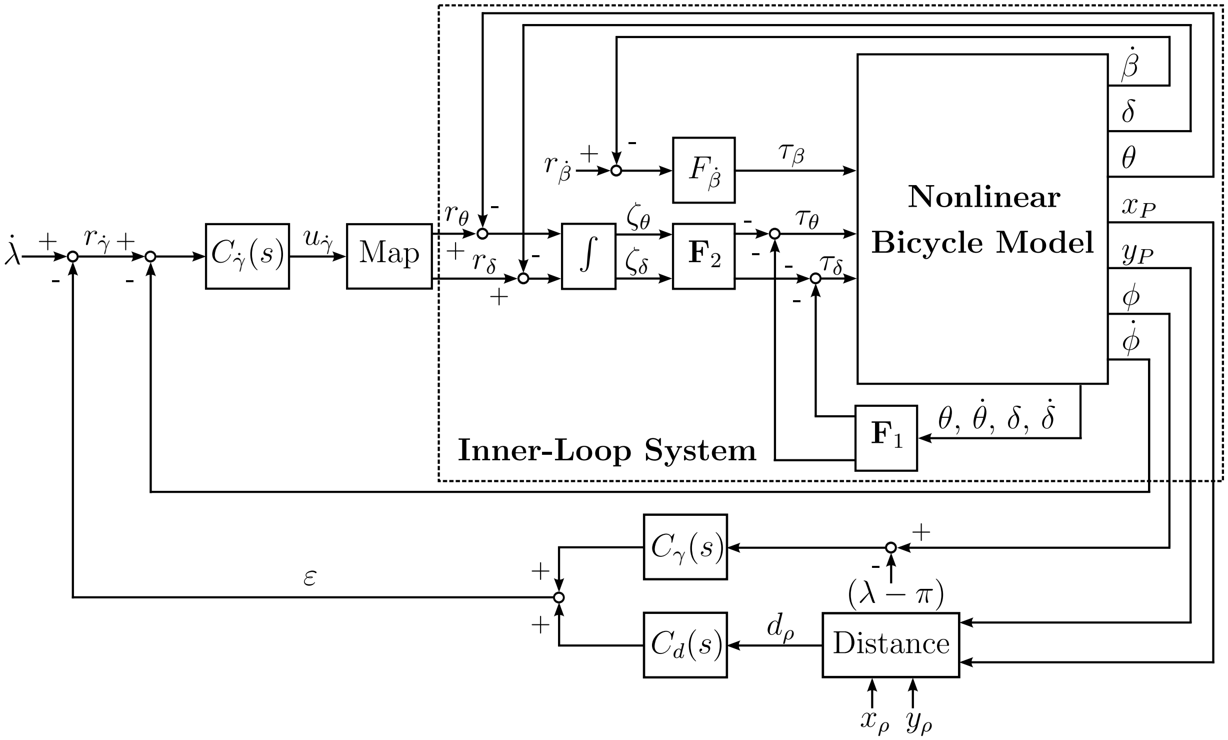


Figure 4.6: Detailed block diagram showing the full control architecture.

distance to the closest point on the path and passed into a controller $C_d(s)$. The values output from $C_\gamma(s)$ and $C_d(s)$ are added together. This sum, ε, is the corrective yaw rate for the bicycle, which is subtracted from $\dot{\lambda}$. The result becomes the reference value, $r_{\dot{\gamma}}$ for the system $H_{\dot{\gamma}}(s)$.

The entire system is rather complicated and nonlinear; to analyze it, we build a system that represents the linear approximation. To achieve this we start with the obvious relations, $\gamma = \frac{1}{s}\dot{\gamma}$, and $\lambda = \frac{1}{s}\dot{\lambda}$, and then use (4.30) to find Δd_ρ. The resulting linear system is shown in Figure 4.7, and after dropping the Δ for convenience, the corresponding transfer function is

$$\frac{d_\rho(s)}{\dot{\lambda}(s)} = \frac{v\left(H_{\dot{\gamma}}(s) - 1\right)}{s^2 + C_\gamma(s)H_{\dot{\gamma}}(s)s + C_d(s)H_{\dot{\gamma}}(s)v}. \tag{4.33}$$

More insight into how this system responds to a step can be gained by substituting

$$H_{\dot{\gamma}}(s) = \frac{N_P(s)N_C(s)}{D_P(s)D_C(s) + N_P(s)N_C(s)}, \tag{4.34}$$

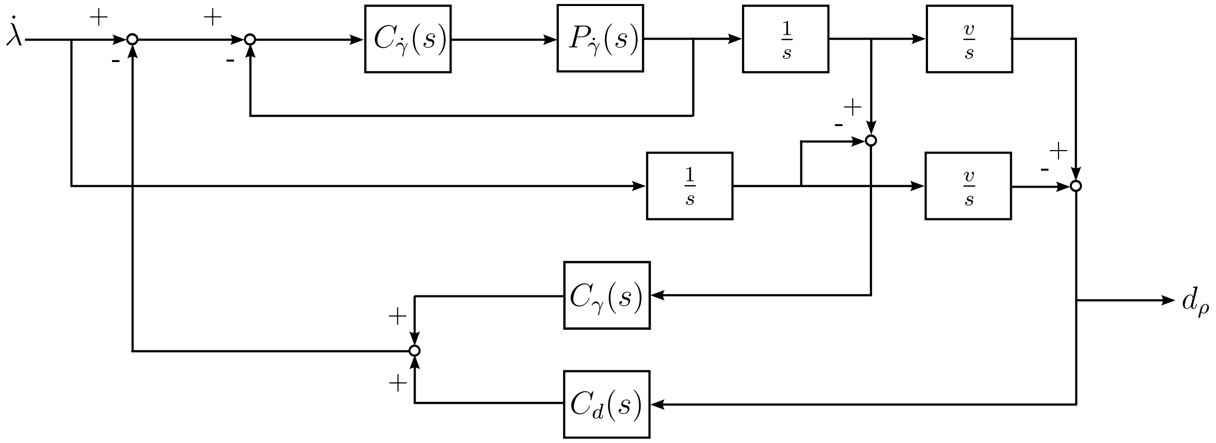


Figure 4.7: Linearized block diagram of the entire system.

where $N_P(s)$ and $N_C(s)$ are respectively the numerators of $P_{\dot{\gamma}}(s)$ and $C_{\dot{\gamma}}(s)$, while $D_P(s)$ and $D_C(s)$ are the corresponding denominators. Substituting (4.34) into (4.33) gives

$$\frac{d_\rho(s)}{\dot{\lambda}(s)} = -\frac{D_P(s)D_C(s)v}{(D_P(s)D_C(s) + N_P(s)N_C(s))s^2 + C_\gamma(s)N_P(s)N_C(s)s + C_d(s)N_P(s)N_C(s)v}. \tag{4.35}$$

An important observation is that assuming at one of $C_{\dot{\gamma}}(s)$, $C_\gamma(s)$ or $C_d(s)$ is designed with an integrator, barring a poor choice for the remaining two controllers, the transfer function has a differentiator term. Assuming that $C_{\dot{\gamma}}(s)$, $C_\gamma(s)$ and $C_d(s)$ are chosen so that at least one of them has an integrator and that (4.35) is stable, after applying the final value theorem, we find due to the differentiator term that

$$\lim_{s\to 0} s \cdot d_\rho(s) \cdot \frac{k}{s} = 0. \tag{4.36}$$

This result shows that the contact point of the rear wheel of the bicycle with the ground will track a path of constant curvature with zero steady-state error.

We recommend that $C_{\dot{\gamma}}(s)$ and $C_d(s)$ are designed to both contain an integrator. Note that the path following control contains multiple loops, and that the integrator in $C_{\dot{\gamma}}(s)$ is recommended as the bandwidth of the yaw-rate control can be higher than that of the yaw angle or distance control. Under ideal conditions this single integrator causes the bicycle to perfectly track a path of constant yaw rate. However, we have estimated the value of $\dot{\gamma}$ and γ by $\dot{\phi}$ and $\phi + \pi$ which introduces a small error. To overcome this limitation an integrator can be introduced to $C_d(s)$ which will actually cause $\lim_{t\to\infty} d(t) = 0$, and thus the bicycle will track the path with zero steady-state error. Since the error in the estimation of $\dot{\gamma}$ and

γ is small, the idea of using two integrators is that the high bandwidth yaw rate control provided by $C_{\dot{\gamma}}(s)$ will get the bicycle close to the path quickly and then $C_d(s)$ will move the bicycle perfectly onto the path sometime after.

4.5 Summary of Path-Following Control

The controllers used to achieve path tracking are $C_{\dot{\gamma}}(s)$, $C_{\gamma}(s)$, and $C_d(s)$, which are configured as shown in the block diagram of Figure 4.6. We have designed path-following control by first providing control of the yaw rate. This is achieved by first developing a map in Section 4.1 to convert a commanded yaw rate to an appropriate r_δ and r_θ for the inner-loop controller. In Section 4.2 we perform system identification on the inner loop to find a linear estimate of the response from $u_{\dot{\gamma}}$ to $\dot{\gamma}$. We estimate the value of $\dot{\gamma}$ by the value of $\dot{\phi}$. Once the plant is identified we use the classical feedback controller $C_{\dot{\gamma}}(s)$ to control the yaw rate.

The yaw rate controller alone is not enough to converge to a particular path of constant yaw rate. We also need to look at the difference in the yaw angle of the bicycle, γ, and that of the path, λ, as well as the distance from the path, d_ρ. We estimate the value of γ by $\phi + \pi$ and develop a linear estimate of d_ρ from the path in Section 4.3. We use $C_\gamma(s)$ to provide a correction for the bicycle heading when compared to the path heading, and $C_d(s)$ to bring the bicycle onto the desired path. In in Section 4.4 we also recommended for both $C_{\dot{\gamma}}(s)$ and $C_d(s)$ contain an integrator to guarantee that the bicycle asymptotically tracks a constant curvature path.

Chapter 5

Design Example

In this chapter we present a step by step design example of the control system described in Chapters 3 and 4. We start with the design of the inner-loop lean and steer controller and speed controller in Section 5.1 and then discuss the performance of this inner-loop design in Section 5.2. The design of the path following outer loop is then described in detail in Section 5.3. The performance of constant curvature path following is analyzed in Section 5.4, and this analysis motivates the notion of a virtual path as a tool to help control the bicycle if it is far from the actual path. This section also shows the system response for following a practical path, and some limitations on the path based on this performance.

5.1 Inner-Loop Design

For demonstration of the inner-loop controller a bicycle at the operating point (3.1) with $v = 5$ m/s is studied. The LQR matrices, $\bar{\mathbf{Q}}$ and $\bar{\mathbf{S}}$, are chosen to be

$$\begin{aligned} \bar{\mathbf{Q}} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}, \\ \bar{\mathbf{S}} &= \begin{bmatrix} 10^{-5} & 0 \\ 0 & 10^{-4} \end{bmatrix}. \end{aligned} \tag{5.1}$$

This choice results in the gain matrix

$$\mathbf{F} = 10^3 \cdot \begin{bmatrix} 0.597 & 0.182 & 0.25 & 0.0014 & -0.23 & -3.15 \\ -0.433 & -0.0616 & 0.0887 & 0.00542 & 0.997 & -0.0728 \end{bmatrix}. \tag{5.2}$$

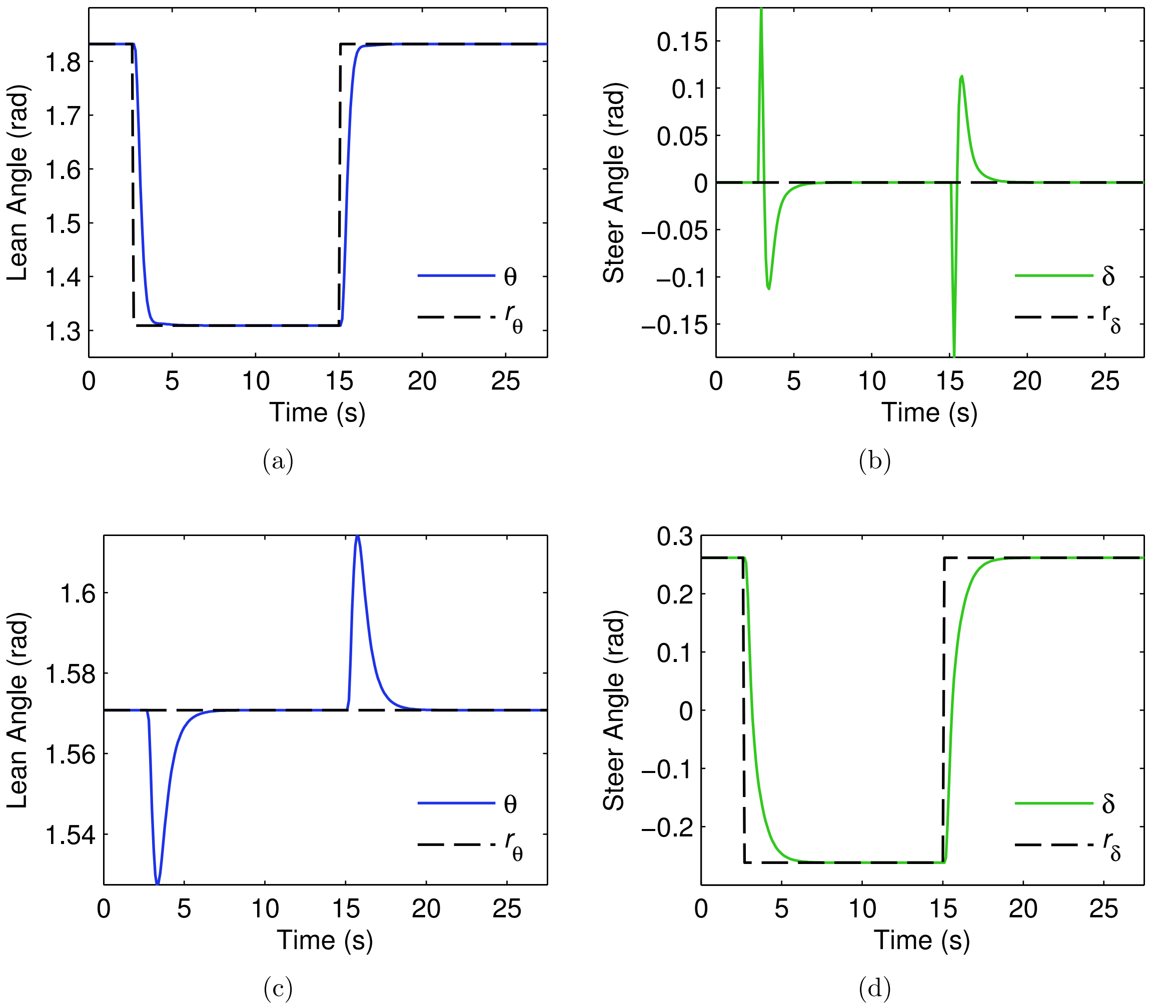
These gains correspond to what would, in practice, be considered to be high performance and can lead to high values of torque, particularly for τ_θ. If the torque is too high in practice the (1,1) and (2,2) entries of **S** can be increased and new gains can be computed.

To determine an appropriate gain for the speed controller, the closed-loop response of the linearized bicycle model and LQR controller are provided in Figure 5.1. Here we see that when applying a step of $\frac{1}{6}\pi$ radians to θ the response reaches a two percent settling time in approximately 1.20 seconds, and when applying a step of $\frac{1}{6}\pi$ radians to δ it settles in approximately 2.65 seconds. From this analysis we decide to use a controller with a two percent settling time of $\frac{1}{4}$ seconds. Recalling that this occurs after four time constants for a first-order system, the time constant of this controller is chosen to be $\frac{1}{16}$ seconds, which corresponds to $F_{\dot{\beta}} = 195$ in (3.32).

5.2 Inner-Loop Performance

In Figure 5.2 the performance of the LQR controller on the nonlinear system is shown for a step on each of the θ and δ angles around the linearized operating trajectory. The transient response is smooth with no overshoot, which is indicative of a high phase margin. The settling time is also very similar to the linear controller for a step of $\frac{1}{6}\pi$ radians on both θ and δ. The settling time for the lean step is slightly faster than the linear case at 1.12 seconds and the steer step is slightly slower at 2.69 seconds. In these responses the proportional compensator for speed control is also active, and a plot of $\dot{\beta}$ for the applied steps is shown in Figure 5.3. There is a transient in the response at each step, but only a small variation in magnitude of $\dot{\beta}$ is seen. For the step on θ this difference is 0.15 radians per second and 0.01 radians per second for a δ step.

The system is also shown in Figure 5.4 to be stable for a step of $\frac{1}{4}\pi \leq r_\theta \leq \frac{3}{4}\pi$ when $r_\delta = 0$ and $-\frac{1}{3}\pi \leq r_\delta \leq \frac{1}{3}\pi$ when $r_\theta = \frac{1}{2}\pi$. The two percent settling time of the r_θ step increases to 1.95 seconds and there is some degradation to the transient in the form of overshoot and a small amount of oscillation. There is similarly an increase in settling time to 5.20 seconds when looking at the step applied to r_δ. Although the settling time increases, the transient is similar to the response seen in the linearized trajectory. In Figure 5.5 the response of the speed controller is shown. The response looks very similar to that for the



(a)

(b)

(c)

(d)

Figure 5.1: The response of the linear closed-loop system at 5 m/s for θ in (a) and δ in (b) when applying a step on θ from $\frac{5}{12}\pi$ radians to $\frac{7}{12}\pi$ radians while holding δ at 0. Likewise the linear response at 5 m/s for for θ in (c) and δ in (d) when applying a step on δ from $-\frac{1}{6}\pi$ radians to $\frac{1}{6}\pi$ radians while holding θ at $\frac{\pi}{2}\pi$ radians.

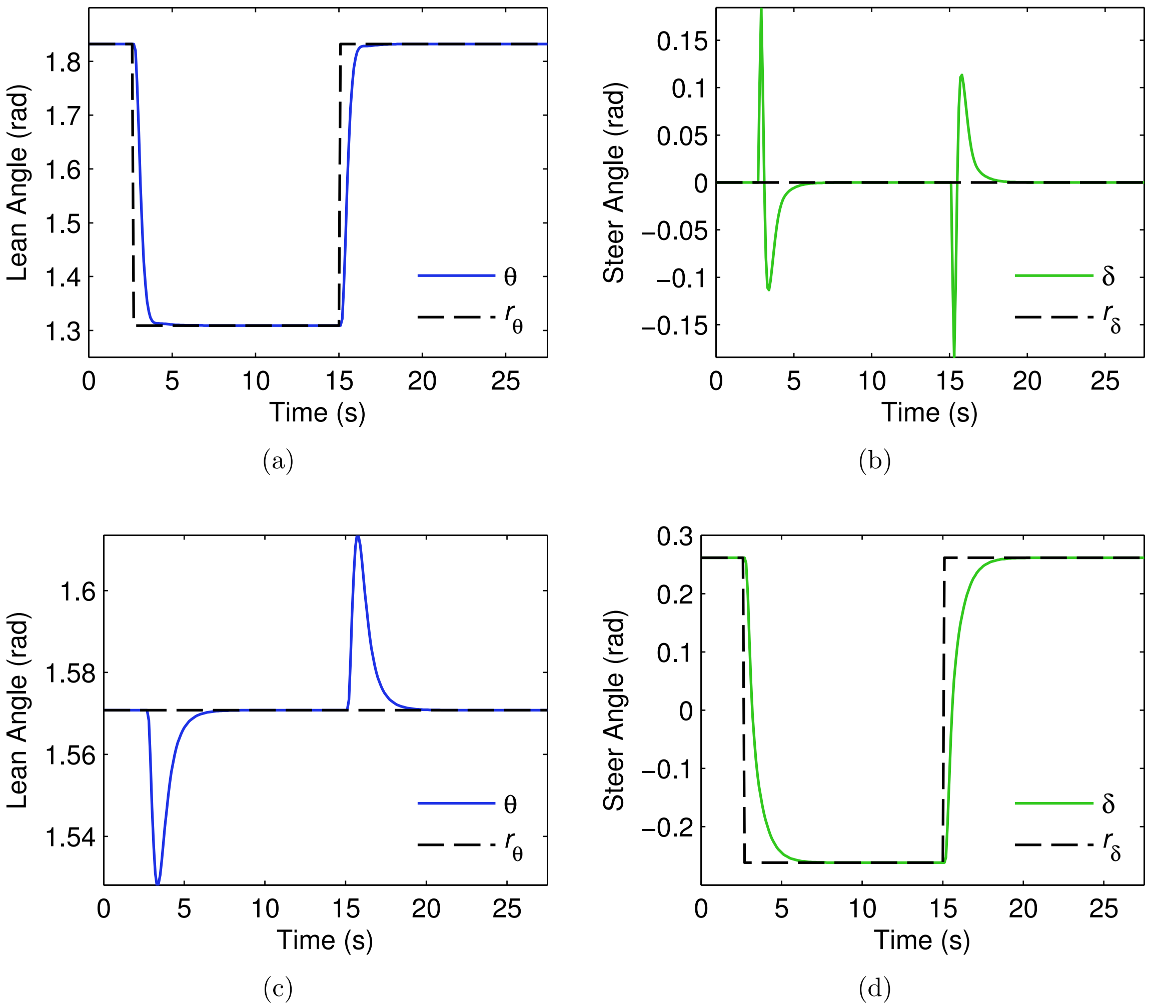


Figure 5.2: The response of the closed-loop system at 5 m/s for θ in (a) and δ in (b) when applying a step on θ from $\frac{5}{12}\pi$ radians to $\frac{7}{12}\pi$ radians while holding δ at 0. Likewise the response at 5 m/s for for θ in (c) and δ in (d) when applying a step on δ from $-\frac{1}{12}\pi$ radians to $\frac{1}{12}\pi$ radians while holding θ at $\frac{1}{2}\pi$ radians.

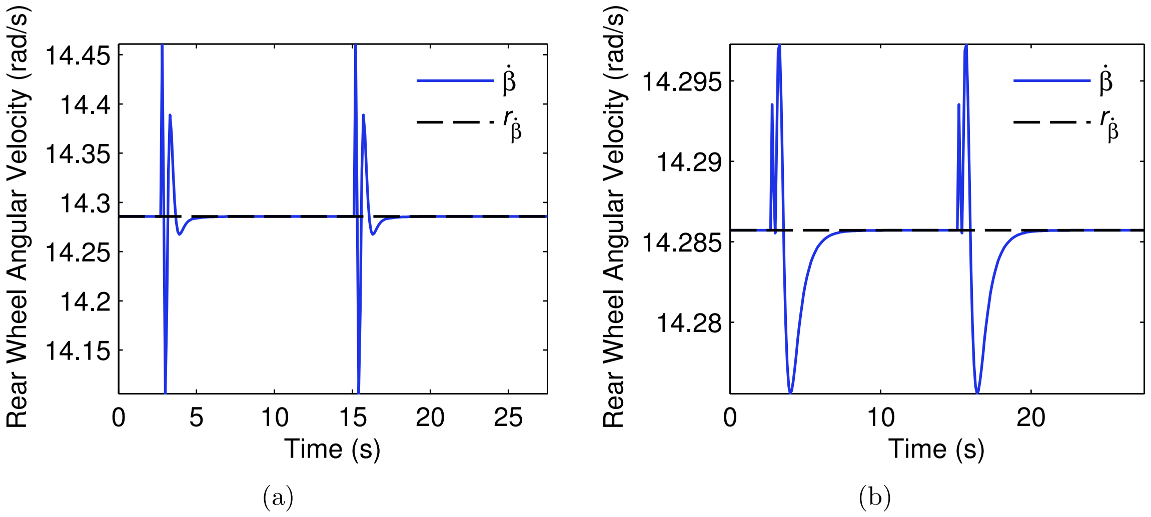


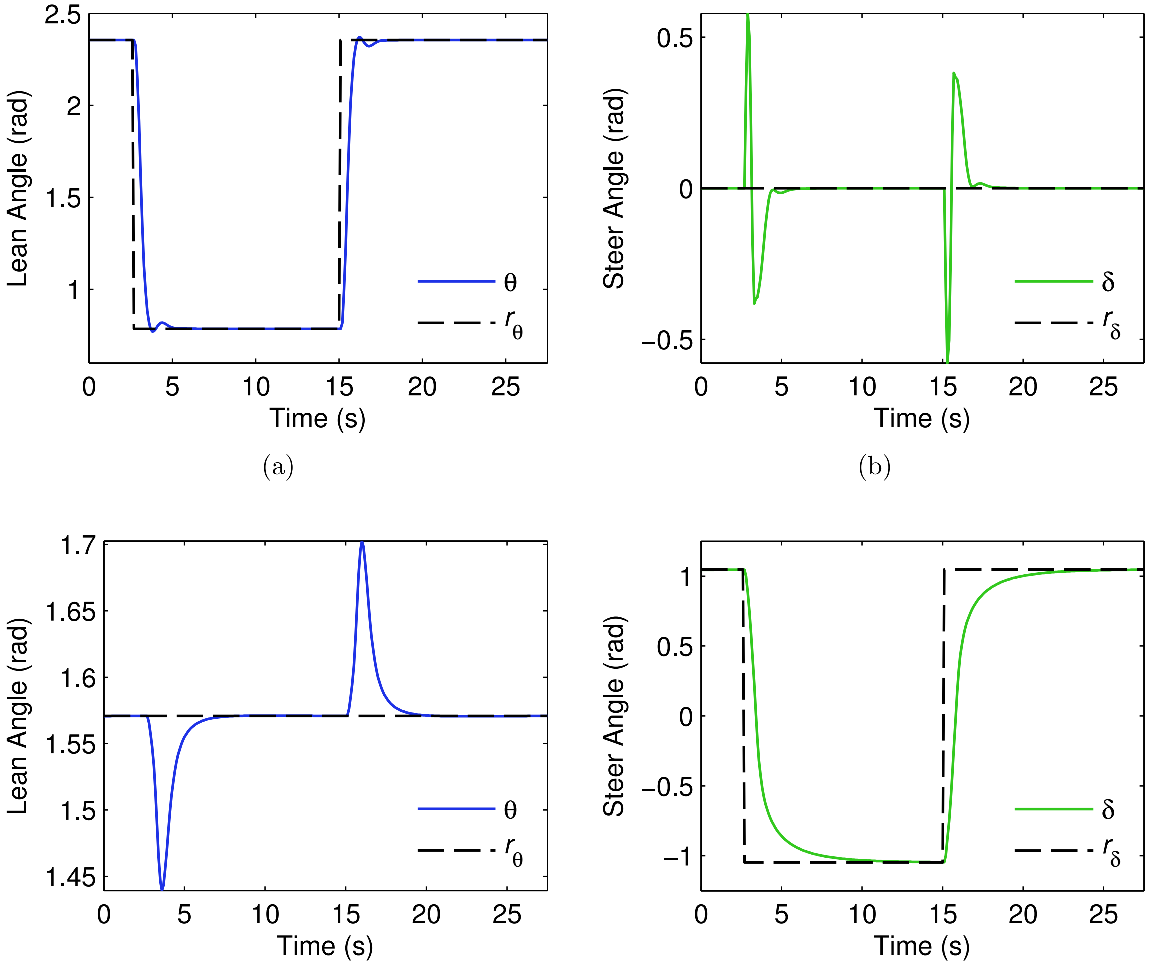
Figure 5.3: Response of $\dot{\beta}$ with the speed controller in (a) and (b) for the step applied to θ and δ in Figure 5.2 respectively.

smaller steps for θ and δ; however, the response also takes longer and the magnitude of deviation in $\dot{\beta}$ is larger at 1.32 radians per second for the lean angle step and 0.399 radians for the steer angle step.

Further testing reveals that the response of the closed-loop system with the LQR controller gains from (5.2) does become unstable with larger steps. In this case the system is shown to go unstable for $\frac{1}{6}\pi \leq r_\theta \leq \frac{5}{6}\pi$ when $r_\delta = 0$ and $-\frac{3}{8}\pi \leq r_\delta \leq \frac{3}{8}\pi$ when $r_\theta = 0$. This analysis leads to the conclusion that θ and δ should be saturated to ensure that the system does not become unstable, and in cases where the increased settling time is unacceptable the controller should be redesigned or saturated at a smaller angle.

5.3 Path-Following Design

We continue the design example of Section 5.1. To prevent the bicycle from tipping over and to maintain performance similar to the linearized LQR model we saturate $|r_\delta| \leq \frac{1}{6}\pi$ radians per second. From (4.7) this corresponds to setting $|u_{\dot{\gamma}}| \leq 2.44$, and (4.10) gives the approximation for maximum curvature is $\max \kappa = 0.533 \text{ m}^{-1}$. For the design example, which uses a constant $v = 5$ m/s, we choose to keep a constant $r_\theta = \frac{1}{2}\pi$ radians instead of minimizing τ_θ at steady state. This is done because minimizing τ_θ further limits the



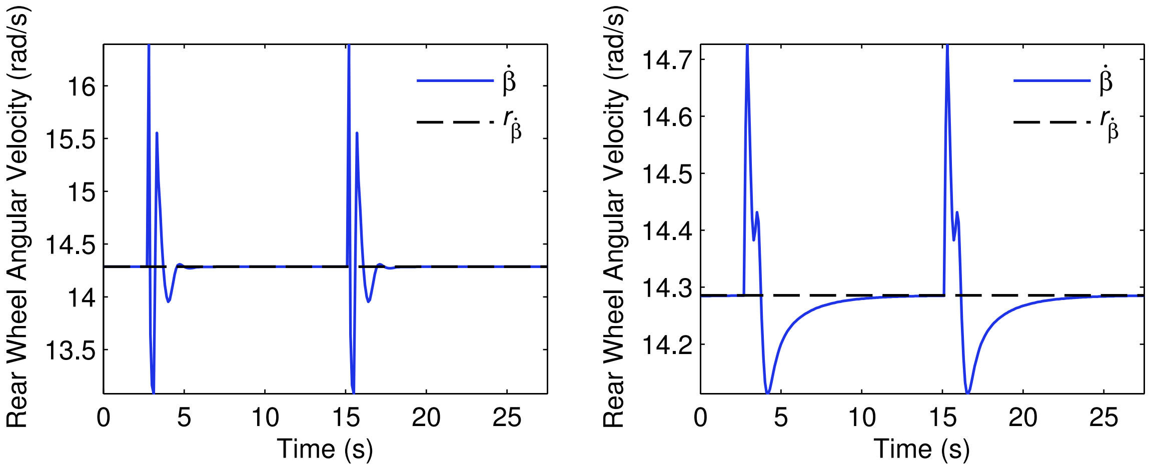
(a)

(b)

(c)

(d)

Figure 5.4: The response of the closed-loop system at 5 m/s for θ in (a) and δ in (b) when applying a step on θ from $\frac{1}{4}\pi$ radians to $\frac{3}{4}\pi$ radians while holding δ at 0. Likewise the response at 5 m/s for for θ in (c) and δ in (d) when applying a step on δ from $-\frac{1}{3}\pi$ radians to $\frac{1}{3}\pi$ radians while holding θ at $\frac{\pi}{2}\pi$ radians.



(a)

(b)

Figure 5.5: Response of $\dot{\beta}$ with the speed controller in (a) and (b) for the step applied to θ and δ in Figure 5.4 respectively.

allowable $u_{\dot{\gamma}}$, as the relationship becomes $r_\theta = 2.39 r_\delta$.

A step response from $u_{\dot{\gamma}} = 0$ radians per second to $u_{\dot{\gamma}} = 2.44$ radians per second is shown in Figure 5.6, which demonstrates that for the design example the system can be approximated using a first-order transfer function

$$P_{\dot{\gamma}}(s) = \frac{b_0}{s + a_0}, \tag{5.3}$$

where a_0, and b_0 are constants. In actuality the values vary depending on the size of the yaw rate step due to nonlinearities in the bicycle.

Note that $P_{\dot{\gamma}}(s)$ is stable and that $P_{\dot{\gamma}}$ always has a positive DC gain due to the map (4.7). This means that necessarily

$$a_0 > 0, \quad b_0 > 0. \tag{5.4}$$

Converting (5.3) to discrete time using the bilinear transformation results in

$$P_{\dot{\gamma}}[z] = \frac{g(z+1)}{z + f_0}, \tag{5.5}$$

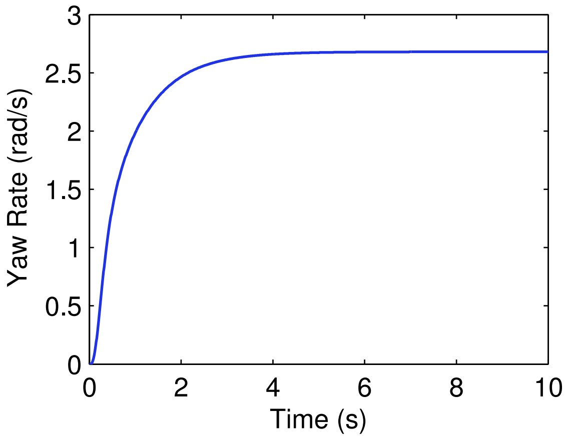


Figure 5.6: Response of the inner loop system to a $u_{\dot{\gamma}}$ step of 2.44 radians per second. The r_δ is determined using (4.7) and $r_\theta = \frac{1}{2}\pi$ radians.

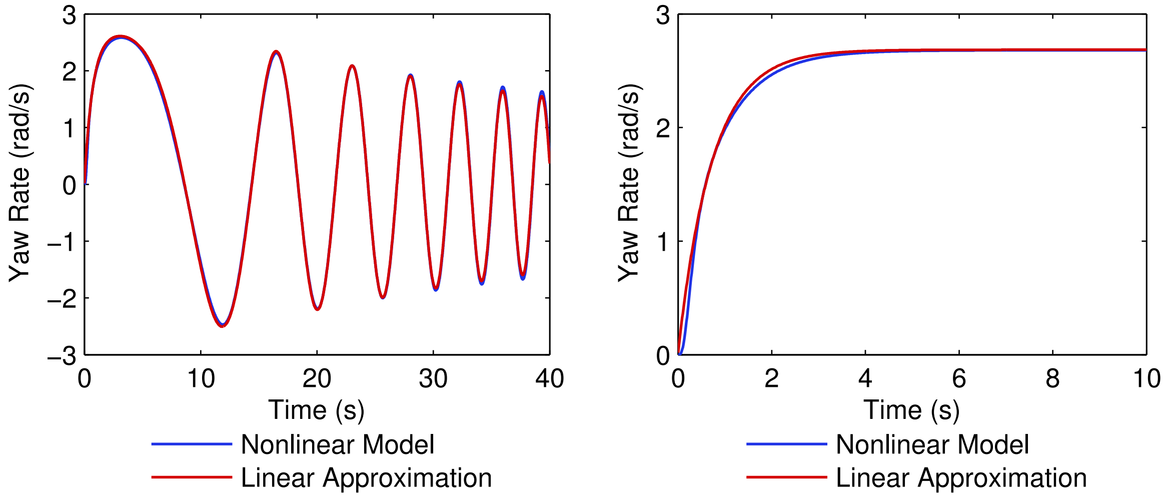
where

$$\begin{aligned} g &= \frac{b_0 T}{a_0 T + 2} \\ f_0 &= \frac{a_0 T - 2}{a_0 T + 2}. \end{aligned} \tag{5.6}$$

To find appropriate values for a_0 and b_0, a least-squares problem is set up as in (4.22). Using more familiar notation this results in

$$\begin{bmatrix} \dot{\gamma}[t_1] \\ \dot{\gamma}[t_2] \\ \vdots \\ \dot{\gamma}[t_N] \end{bmatrix} = \begin{bmatrix} \dot{\gamma}[t_1] \\ \dot{\gamma}[t_2] \\ \vdots \\ \dot{\gamma}[t_N] \end{bmatrix} - \begin{bmatrix} \dot{\gamma}^{mod}[t_0] & u_{\dot{\gamma}}[t_1] + u_{\dot{\gamma}}[t_0] \\ \dot{\gamma}^{mod}[t_1] & u_{\dot{\gamma}}[t_2] + u_{\dot{\gamma}}[t_1] \\ \vdots & \vdots \\ \dot{\gamma}^{mod}[t_{N-1}] & u_{\dot{\gamma}}[t_N] + u_{\dot{\gamma}}[t_{N-1}] \end{bmatrix} \cdot \begin{bmatrix} -f_0 \\ g \end{bmatrix}. \tag{5.7}$$

To apply the least squares algorithm to the problem, a linear swept frequency cosine signal is used as the input, $u_{\dot{\gamma}}$. The system is actually nonlinear and higher order than the estimate, but we wish to have the model fit well at low frequencies, so the signal frequency is kept small. Note that we use the output values of $\dot{\phi}$ to approximate $\dot{\gamma}$ when performing this system identification. From the step response in Figure 5.6, the settling time appears to be approximately 3.18 seconds. This implies that a continuous time pole should be in the range of 1.26, so the linear sweep used starts at zero radians per second and finishes at



(a)

(b)

Figure 5.7: Response of the inner loop system and linear model to a $u_{\dot{\gamma}}$ linear swept frequency cosine signal from zero radians per second at $t = 0$ to 2 radians per second at $t = 40$ with an amplitude of 2.44 radians per second in (a) and a step response with a 2.44 radian per second amplitude in (b).

an instantaneous frequency of two radians per second. This is approximately one and a half times faster than the approximate corner frequency, which should provide enough phase shift and decay in the output to identify the dominant pole in the system, while limiting the decay to minimize the nonlinearities present. This method results in $g = 0.00751$ and $f_0 = -0.986$ for a sample period $T = 0.01$ seconds. Substituting these values into (5.6) and solving for b_0 and a_0 results in a continuous time approximation of

$$P_{\dot{\gamma}}(s) = \frac{1.51}{s + 1.37}. \tag{5.8}$$

Looking at Figure 5.7, this model provides a good estimate at low frequencies and for a step of $u_{\dot{\gamma}} = 2.44$ radians per second.

Since the system behaviour is approximated well by (5.8), a simple integral controller with positive gain can be used to provide perfect steady-state tracking to a step signal. Formally, the controller used is

$$C_{\dot{\gamma}}(s) = \frac{F_{\dot{\gamma}}}{s}, \tag{5.9}$$

where $F_{\dot{\gamma}}$ is a positive constant. The transfer function approximation for the yaw rate of the inner loop with the controller $C_{\dot{\gamma}}$ is then

$$H_{\dot{\gamma}}(s) = \frac{1.51F_{\dot{\gamma}}}{s^2 + 1.37s + 1.51F_{\dot{\gamma}}}. \tag{5.10}$$

We have also imposed the condition that $H_{\dot{\gamma}}$ must be stable. A necessary and sufficient condition for stability is

$$F_{\dot{\gamma}} > 0. \tag{5.11}$$

To implement the path following controller, we start by substituting (5.10) into (4.33). The resulting transfer function for the example system is

$$\frac{d_\rho(s)}{\dot{\lambda}(s)} = -\frac{sv(s+1.37)}{s^4 + 1.37s^3 + 1.51F_{\dot{\gamma}}s^2 + 1.51F_{\dot{\gamma}}C_\gamma(s)s + 1.51F_{\dot{\gamma}}C_d(s)v}. \tag{5.12}$$

A simple choice for $C_\gamma(s)$ and $C_d(s)$ is

$$C_\gamma(s) = F_\gamma, \quad C_d(s) = F_{d,P} + \frac{F_{d,I}}{s}, \tag{5.13}$$

where F_γ and F_d are constant gains. Substituting (5.13) into (5.12) yields

$$\frac{d_\rho(s)}{\dot{\lambda}(s)} = -\frac{s^2v(s+1.37)}{s^5 + 1.37s^4 + 1.51F_{\dot{\gamma}}s^3 + 1.51F_{\dot{\gamma}}F_\gamma s^2 + 1.51F_{\dot{\gamma}}F_{d,P}vs + 1.51F_{\dot{\gamma}}F_{d,I}v}. \tag{5.14}$$

By performing Routh Hurwitz analysis in conjunction with (5.4) and (5.11), the following are necessary and sufficient conditions for stability of the entire system:

$$\begin{aligned}
1.37 - F_\gamma \quad &> \quad 0,\\
F_{d,I} \quad &> \quad 0,\\
1.51F_{\dot{\gamma}}F_\gamma(1.37 - F_\gamma) - 1.37v(1.37F_{d,P} - F_{d,I}) \quad &> \quad 0,\\
-1.51F_{\dot{\gamma}}v(F_{d,I}^2v - 2.75F_{d,P}F_{d,I}v - 1.51F_{\dot{\gamma}}F_\gamma F_{d,I} + 2.08F_{\dot{\gamma}}F_{d,I} &\\
+1.89F_{d,P}^2v + 1.51F_{\dot{\gamma}}F_\gamma^2F_{d,P} - 2.08F_{\dot{\gamma}}F_\gamma F_{d,P}) \quad &> \quad 0.
\end{aligned} \tag{5.15}$$

These conditions also imply that $F_\gamma > 0$, and recalling that we approximated $P_{\dot{\gamma}}$ by a first order system when it is actually higher order nonlinear system there is also a limitation on how large $F_{\dot{\gamma}}$ can be. This restriction becomes apparent by looking at (4.34) and (4.35). Using Root Locus methods for analysis for any system with a relative order greater than two, as gain increases at least one pole moves into the RHP. This implies that in actuality when $F_{\dot{\gamma}}$ is too large (4.34) violates the sufficient condition for stability and since proportional controllers are used in this example for $C_{\gamma}(s)$ and $C_d(s)$, the overall system also becomes unstable. Additionally, a large value of $F_{\dot{\gamma}}$ easily leads to integrator wind-up when implemented on the nonlinear bicycle model.

The performance of the path-following controller is tuned by varying $F_{\dot{\gamma}}$, F_{γ} and F_d. Substituting all values into the stability equations, we have

$$\begin{aligned} F_{\dot{\gamma}} &> 0, \\ 0 < F_{\gamma} &< 1.37, \\ F_d &> 0, \\ 1.51F_{\dot{\gamma}}F_{\gamma}(1.37 - F_{\gamma}) - 6.87(1.37F_{d,P} - F_{d,I}) &> 0, \\ -7.56F_{\dot{\gamma}}(5F_{d,I}^2 - 13.7F_{d,P}F_{d,I} - 1.5F_{\dot{\gamma}}F_{\gamma}F_{d,I} + 2.08F_{\dot{\gamma}}F_{d,I} & \\ +9.44F_{d,P}^2 + 1.51F_{\dot{\gamma}}F_{\gamma}^2F_{d,P} - 2.08F_{\dot{\gamma}}F_{\gamma}F_{d,P}) &> 0. \end{aligned} \tag{5.16}$$

Through experimentation the most reliable way of choosing appropriate gains is by first selecting a large value for $F_{\dot{\gamma}}$ so that the yaw-rate control loop is stable with an acceptable level of oscillation. Then choose $F_{\gamma} = 5F_{d,P} = 30F_{d,I}$, and such that the stability conditions (5.16) are still satisfied. This method typically provides a good starting point from which to tune the controller. After some tuning, the gains chosen for this example are $F_{\dot{\gamma}} = 5.75$, $F_{\gamma} = 0.55$, $F_{d,P} = 0.075$, and $F_{d,I} = 0.01$. The final path-following controllers shown in Figure 4.6 are

$$C_{\dot{\gamma}}(s) = \frac{5.75}{s},\ C_{\gamma}(s) = 0.55, \text{ and } C_d(s) = \frac{0.075s + 0.01}{s}. \tag{5.17}$$

The resulting transfer function is

$$\frac{d_{\rho}(s)}{\dot{\lambda}(s)} = -\frac{5s^2(s + 1.37)}{s^5 + 1.37s^4 + 8.69s^3 + 4.78s^2 + 3.26s + 0.435}. \tag{5.18}$$

Additionally a conditional integrator is used in $C_{\dot{\gamma}}$ as an anti-windup on the integral controller, since the output signal is saturated after passing through the yaw-rate map (4.7). Assuming that $\zeta_{\dot{\gamma}}$ is the state representing the integrator in $C_{\dot{\gamma}}$, and r_{δ}^{sat} is the saturated value of r_{δ}, the logic is as follows:

```
if ($r_\delta > |r_\delta^{sat}|$ AND $F_{\dot{\gamma}}(r_{\dot{\gamma}} - \dot{\phi}) > 0$) OR ($r_\delta < -|r_\delta^{sat}|$ AND $F_{\dot{\gamma}}(r_{\dot{\gamma}} - \dot{\phi}) < 0$)
    $\zeta_{\dot{\gamma}} = 0$
else
    $\zeta_{\dot{\gamma}} = F_{\dot{\gamma}}(r_{\dot{\gamma}} - \dot{\phi})$
```

It is also necessary to saturate the output value of $C_d(s)$, u_d, since a large d_ρ will cause u_d to dominate the value of $r_{\dot{\gamma}}$. When $r_{\dot{\gamma}}$ is large for long enough, it will in turn cause the value of r_δ to saturate. The result is the bicycle spins in a circle with a curvature of approximately $\max \kappa$ or $-\max \kappa$. Defining saturation as $|u_d^{sat}| = 0.275$, and ζ_d as the integrator state in $C_d(s)$, the conditional integration logic is as follows:

```
if ($u_d > |u_d^{sat}|$ AND $F_{d,I} d_\rho > 0$) OR ($u_d > |u_d^{sat}|$ AND $F_{d,I} d_\rho < 0$)
    $\zeta_d = 0$
else
    $\zeta_d = F_{d,I} d_\rho$
```

The output value of $C_\gamma(s)$, u_γ, is not saturated directly. Instead the values of γ and λ are defined modulo 2π radians, and when there is more than π radians between the values of γ and λ, 2π is added to the smaller angle. The difference is found without using the modulus operator, which results in $|\gamma - \lambda| < \pi$. There are some considerations that must be taken into account by not directly saturating u_γ which are discussed in Section 5.4.2. To overcome these limitations the virtual path is introduced.

5.4 Path-Following Performance

In this section the performance for the controllers developed in Section 5.1 and 5.3 is demonstrated. We simulate the entire system from Figure 4.6 including the additional saturators introduced in Section 5.1 and 5.3. The initial conditions for the bicycle in this section are those in (3.1) with $t = 0$ unless otherwise indicated.

5.4.1 Performance Close to Path

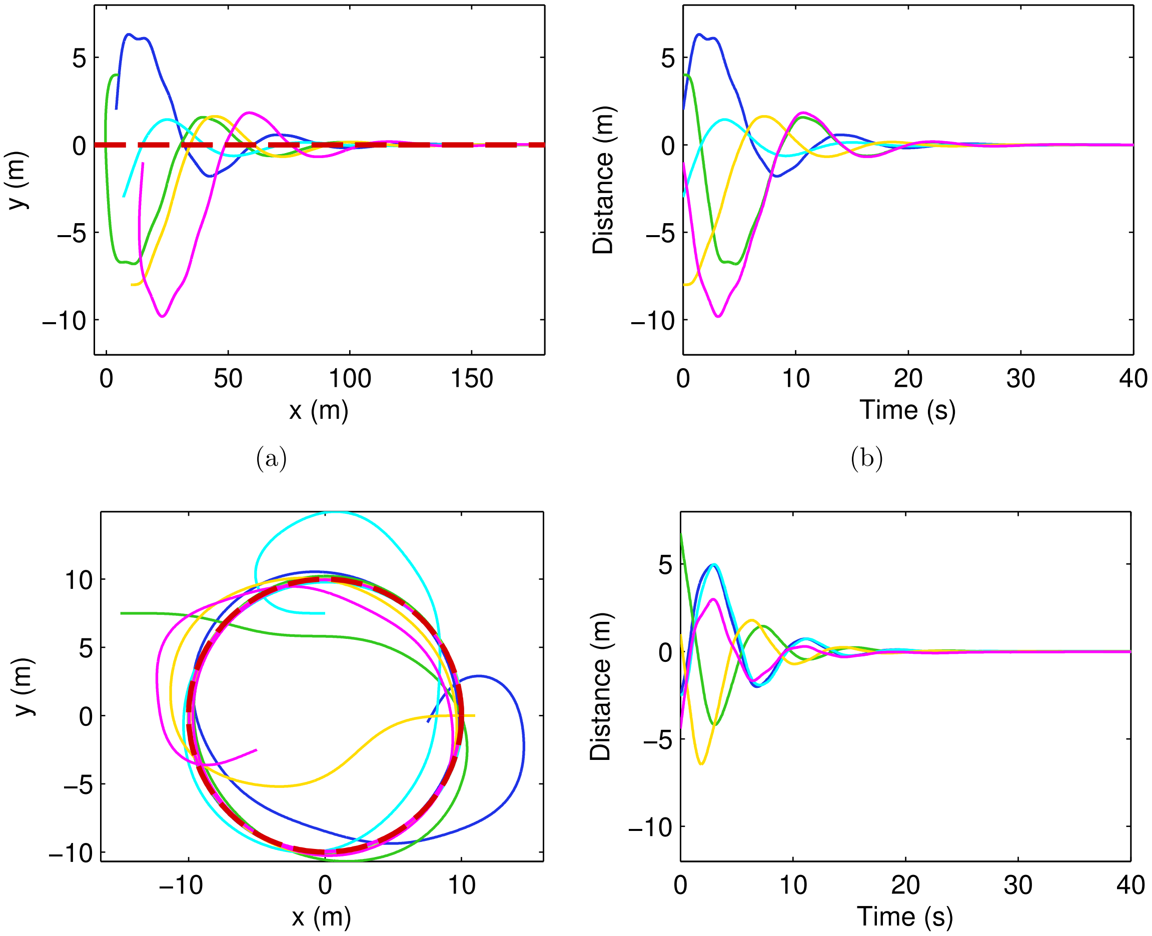
Here we show that the bicycle converges to the path when it starts sufficiently close to the path. Note that the virtual path creation conditions are ignored for simulations in this section. We simulate the system with various initial conditions for tracking a line along the x-axis in Figure 5.8(a). In this figure the bicycle converges for all shown initial conditions, and all trajectories have almost completely converged to the path before reaching $x = 125$ m. Figure 5.8(b) shows that this convergence occurs after approximately 25 seconds.

The convergence to a clockwise circular path of radius 10 m centred at the origin is shown in Figure 5.8(c). Again the rear wheel of the bicycle converges to the path for all the initial conditions shown and almost completely converges within approximately one revolution. For the initial conditions demonstrated Figure 5.8(b) shows that the bicycle again converges before 25 seconds.

5.4.2 Performance Far from Path and Notion of a Virtual Path

To provide justification for use of a virtual path we consider two undesirable results. The first is shown in Figure 5.9(a). Here the bicycle starts far from the 3 m radius counter-clockwise circular path centred at the origin. The result is that the bicycle spins in tight counter-clockwise circles as it converges toward the path. To overcome the undesirable transient we introduce the virtual path shown in Figure 5.9(b) as a desired red curve. At 47 m in length, the straight line segment of the virtual path is too short to allow the bicycle to actually converge; however, the trajectory taken to the actual path is more direct, and in fact it converges to the path faster. This example motivates construction of a virtual path when d_ρ is large or when $|\gamma - \lambda|$ is large. For example here, a virtual path is created if $d_\rho > 5$ m/s and/or $|\gamma - \lambda| > \frac{2}{3}\pi$ radians.

The second case is shown in Figure 5.10(a) where the bicycle starts far from a counter-clockwise circular path of 4 m radius. Here the bicycle can be seen to not converge to the path, and instead converges to a larger radius trajectory in the reverse direction. The reason that this occurs is that the value of u_γ dominates the value of $r_{\dot{\gamma}}$. At this equilibrium, $\dot{\lambda}$ and u_γ are positive, while the value of u_d is negative since the bicycle it is to the right of the path. The value of $r_{\dot{\gamma}}$ is negative due to a large u_γ, and u_d is unable to correct this since it is saturated. This type of equilibrium can occur at other radii for different initial conditions other than the one shown. The result after construction of a virtual path for the system is shown in Figure 5.10(b). Again the virtual path provides a direct route back to the path, and B converges to the path. This motivates not only construction of a virtual path for large values of d_ρ, but also whenever $|\gamma - \lambda| > \frac{2}{3}\pi$ radians. If the value of $|\gamma - \lambda| > \frac{1}{2}\pi$ radians this indicates that the bicycle is beginning to move backward with respect to the path, which is undesirable; setting the magnitude slightly higher allows for a small overshoot if the bicycle starts to move perpendicular to the path. This approach also guarantees that the bicycle will not begin following circular trajectories that are concentric circles in the opposite rotational direction to the intended circular path.



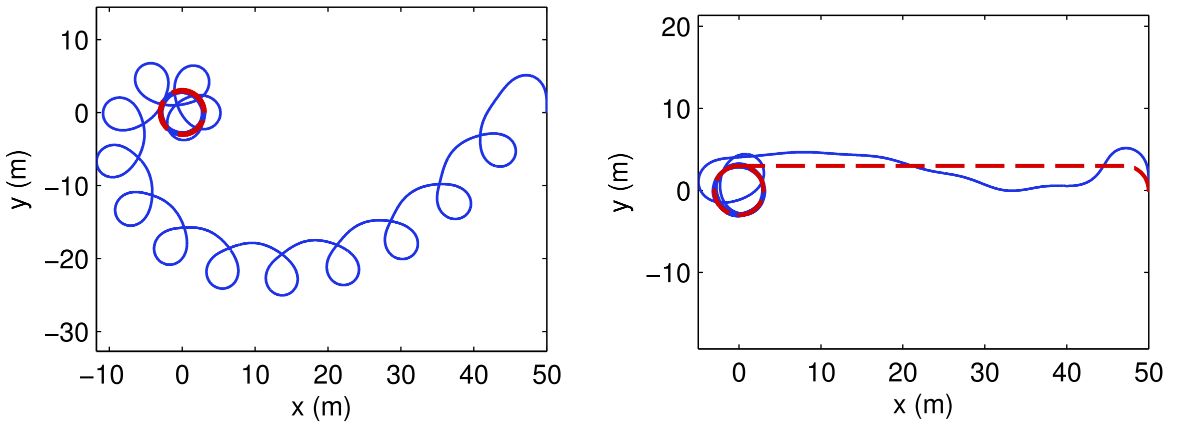
(a)

(b)

(c)

(d)

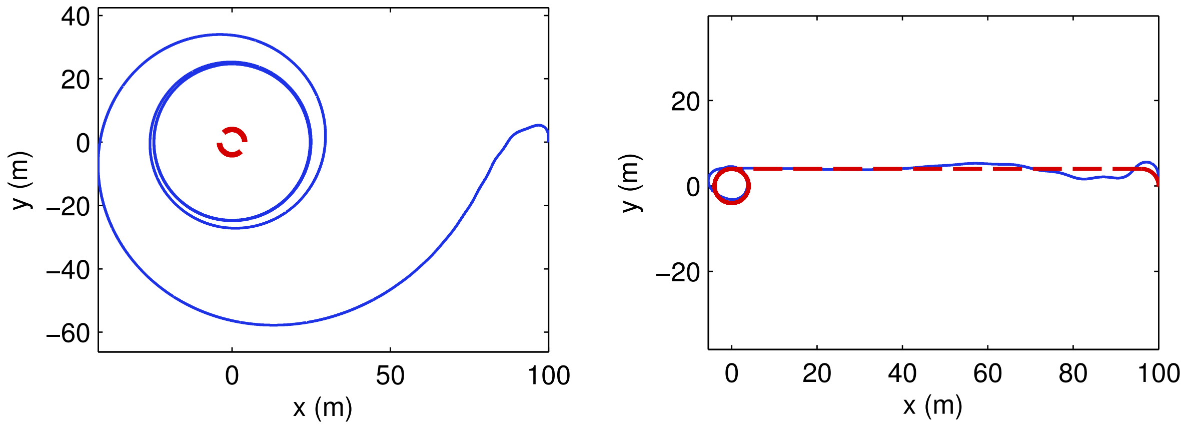
Figure 5.8: Various initial conditions of the bicycle close to the path. Plot (a) shows the trajectory of the bicycle as it converges to a linear path along the x-axis, featured in red, while (a) shows d_ρ versus time for this linear path. Plot (c) shows the trajectory of the bicycle as it begins tracking a clockwise circular path with a radius of 10 m centred at the origin, shown in red, and (a) shows d_ρ versus time for tracking this circular path.



(a)

(b)

Figure 5.9: Bicycle starting at $\mathbf{r}_B = [50 \;\; 0 \;\; 0]^T$ m with an initial $\gamma = \frac{1}{2}\pi$ radians tracking a 3 m radius counter-clockwise circular path centred at the origin without using a virtual path in (a), and with a virtual path constructed in (b). The paths followed are featured in red, and path taken by the bicycle is in blue.



(a)

(b)

Figure 5.10: Bicycle starting at $\mathbf{r}_B = [100 \;\; 0 \;\; 0]^T$ m with an initial $\gamma = \frac{1}{2}\pi$ radians tracking a 4 m radius counter-clockwise circular path centred at the origin without using a virtual path in (a), and with a virtual path constructed in (b). The paths followed are featured in red, and path taken by the bicycle is in blue.

5.4.3 Line Performance

To test the performance of the control system when tracking a line, where the bicycle starts off the path, a linear path along the x-axis is created with $\lambda = 0$. The bicycle is started at $\mathbf{r}_B(0) = [2.5 \;\; 0 \;\; 0]^T$ m and $\gamma(0) = \frac{1}{6}\pi$ radians. The resulting trajectory and d_ρ with respect to time are shown in Figures 5.11(a) and (b) respectively. The initial conditions are chosen such that the creation of a virtual path is not triggered. The bicycle settles within 0.05 m of the path after approximately 21.1 s, which corresponds to an x value of 105 m. The lean and steer torques are shown for the trajectory in Figures 5.11(c) and (d) respectively. There is oscillatory behaviour in addition to the general trend experienced by both τ_θ and τ_δ, which is a consequence of the choice for $C_{\dot{\gamma}}$ in (5.17). By choosing a large value for the yaw rate integral controller gain, in general the magnitude of the first peak in d_ρ is reduced, which means that the bicycle stays closer to the path; however, the oscillations seen in the response of τ_θ and τ_δ become more significant. The fast oscillations are visible in θ and δ (see Figure 5.12) but not in the overall response of d_ρ. If this oscillation is deemed undesirable it can be eliminated by choosing a smaller value for $F_{\dot{\gamma}}$ or choosing something other than a pure integral controller for $C_{\dot{\gamma}}$.

5.4.4 Circle Performance

For testing the performance of the bicycle on a circular path when starting off the path, the clockwise circular path of radius 8.85 m centred at the origin is used. This corresponds to the radius and direction of a right turn on a road discussed in Section 1.2. The initial conditions of the bicycle are $\mathbf{r}_B(0) = [-6.35 \quad 0 \quad 0]^T$ m and $\gamma(0) = \frac{1}{3}\pi$ radians, and the resulting path and d_ρ over time are shown in Figures 5.13(a) and (a). It is difficult to directly compare the performance on the circular path to the straight line since even though the initial magnitude of d_ρ is the same, the path begins to curve in toward the bicycle as it goes forward, which reduces the magnitude of the initial peak of d_ρ. The bicycle converges to less than 0.05 m of the path after 23.8 s, which is slightly longer than the straight line simulation in Section 5.4.3. Additionally, there is an error in the estimated value of $\dot{\gamma}$ which causes an initial positive bias to d_ρ. This bias is taken care of by the slower integrator in $C_d(s)$, but even after 40 s the bicycle is still 0.017 m off the path. This means that the bicycle will take more than one revolution on a circular path of radius 8.85 m for the bicycle to converge to the path.

The oscillations present in τ_θ and τ_δ are also seen in Figures 5.13(c) and (d), however they do not present as strongly. Notice due to the choice of $\theta = \frac{1}{2}\pi$ radians that there is a steady-state torque $\tau_\theta = 231$ Nm. The value of θ and δ during the simulation is shown in

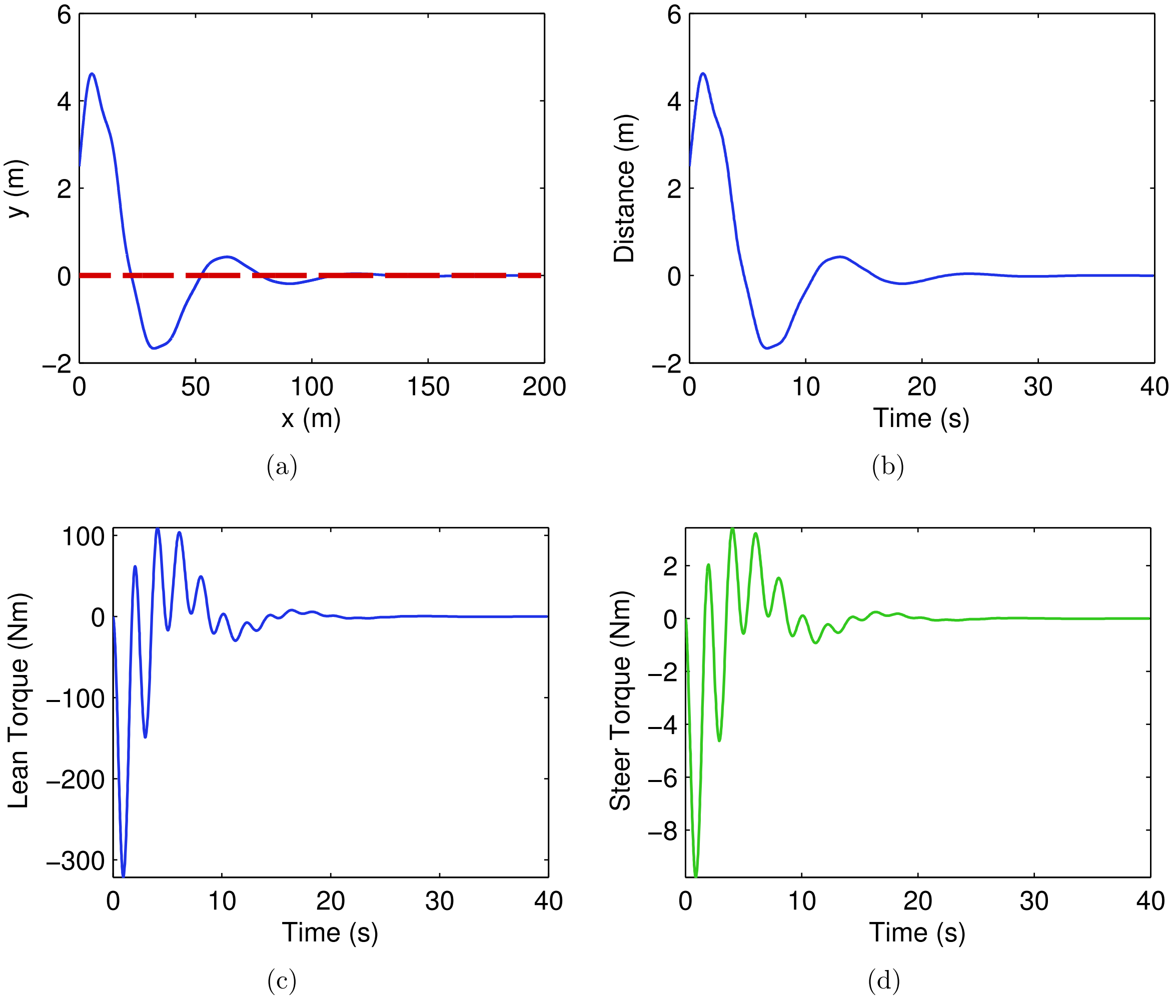
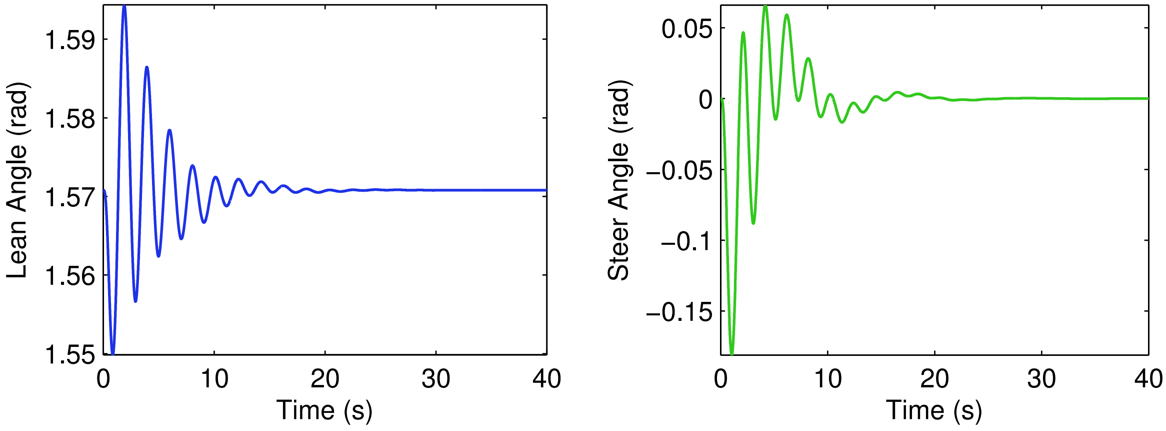


Figure 5.11: Bicycle starting at $\mathbf{r}_B(0) = [2.5 \quad 0 \quad 0]^T$ m and $\gamma(0) = \frac{1}{6}\pi$ radians. Plot (a) shows the trajectory of the bicycle in blue, as it converges to the intended path linear path along the x-axis in red. In (b) d_ρ is shown versus time. Plot (c) shows τ_θ and (d) shows τ_δ required to achieve this trajectory.



(a)

(b)

Figure 5.12: Plot (c) shows θ and (d) shows δ produced during the simulation in Figure 5.11.

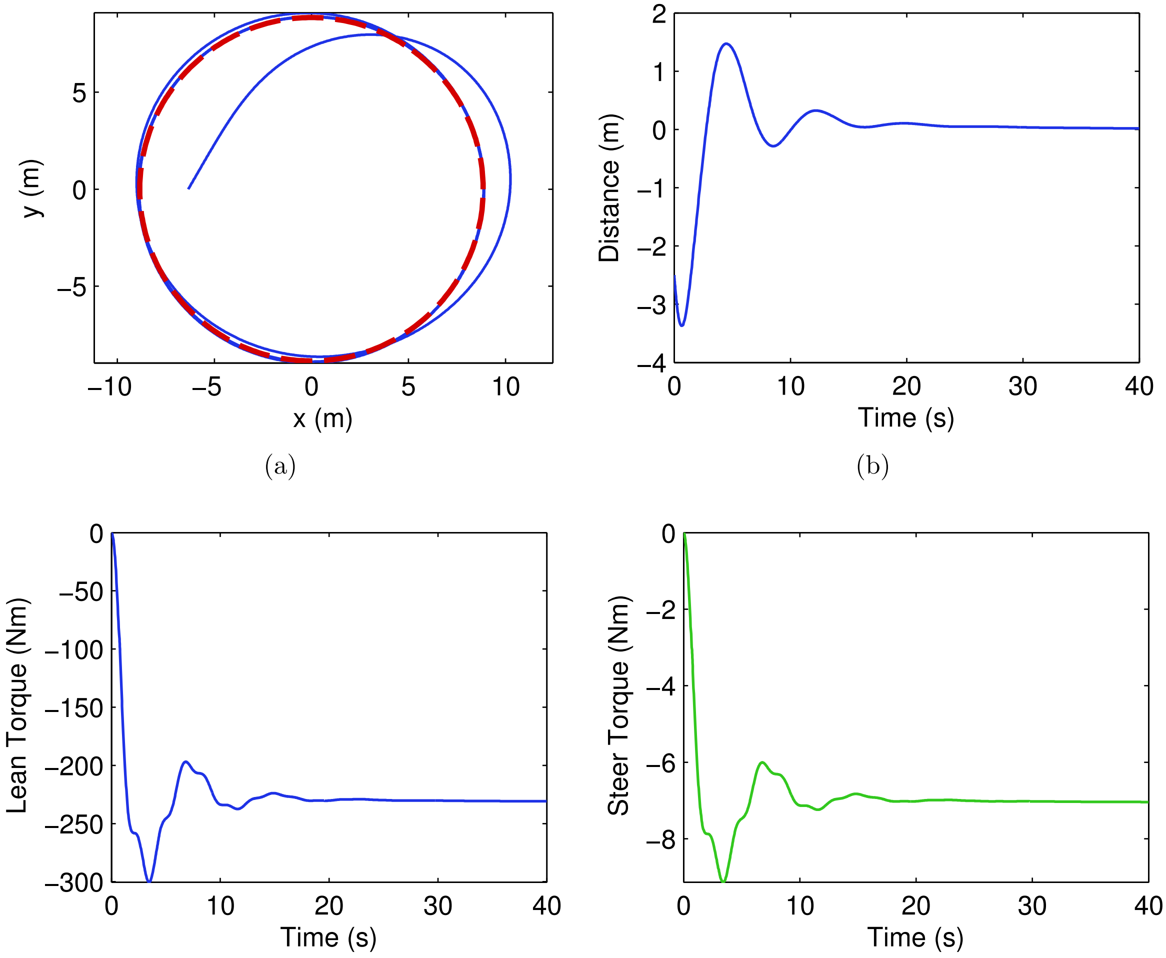
Figure 5.14. Notice also that the fast oscillations are present, but again the magnitude is smaller than the line simulation.

These results are very similar to those of a counter-clockwise circular path of radius 11.6 m, which corresponds to the radius and direction of a left turn on a road discussed in Section 1.2. The only appreciable difference comes from a smaller value of δ, which results in smaller input torques during transient period and steady state. As a result the simulations are omitted.

After additional testing we also set $\max|\kappa| = \frac{1}{3}$ m^{-1}. We adjusted this value as the previous bound from Section 5.3 provides only an approximation for $\max|\kappa|$ from (4.10) which itself is based on a linear approximation in (4.8). The limit (4.10) is also an approximation of $\max|\kappa|$ for a path if the bicycle starts exactly on the path, which is generally not the case. The value of $\max|\kappa| = \frac{1}{3}$ m^{-1} is chosen as testing in simulation shows that the bicycle does not converge to the path or exhibits very poor path-following performance when the path curvature is greater than $\frac{1}{3}$ m^{-1}.

5.4.5 Practical Path Performance

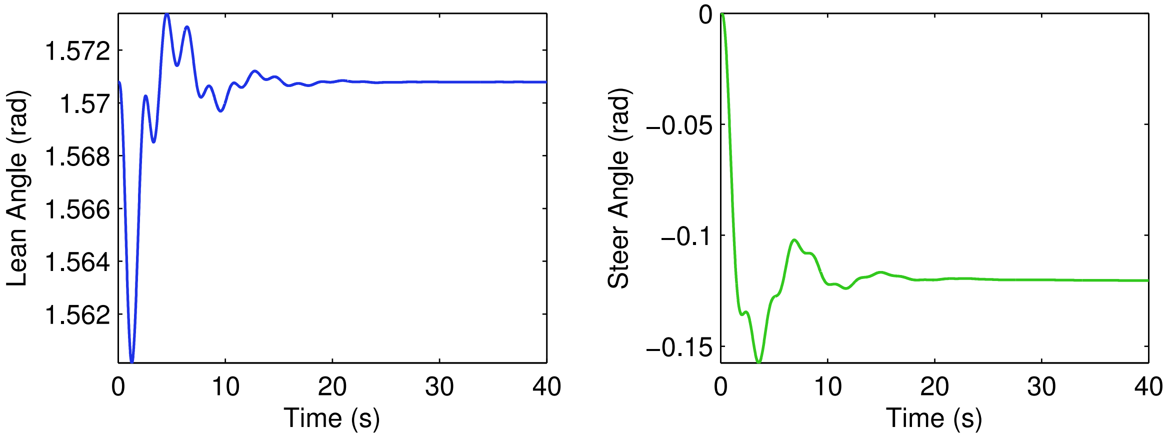
In Sections 5.4.3 and 5.4.4 the performance of the bicycle tracking linear and circular paths, respectively, has been discussed. The major implications from the results for a bicycle on



(a) (b)

(c) (d)

Figure 5.13: Part (a) shows the trajectory of the bicycle in blue, with initial conditions $\mathbf{r}_B(0) = [-6.35\ \ 0\ \ 0]^T$ m and $\gamma(0) = \frac{1}{3}\pi$ radians, as it converges to the intended clockwise circular path of radius 8.85 m centred at the origin in red. In (b) d_ρ is shown versus time. The τ_θ and τ_δ that are applied to the bicycle to achieve this trajectory are shown in (c) and (d), respectively.



(a)

(b)

Figure 5.14: Plot (c) shows θ and (d) shows δ produced during the simulation in Figure 5.13.

a more practical path, discussed in Section 1.2, is the distance it takes for a bicycle to converge to the path. In the case of a line it takes 105 m for the bicycle to come within 0.05 m of the path. For both a circle of radius 8.85 m and 11.6 m it takes longer to converge than the linear case, which implies that it takes more than a single revolution to converge to a circle with the radius of a right or left turn on a practical path. When following a practical path, the circular arc being followed in a turn is less than one revolution, which implies that when cornering the bicycle will not converge. This error is fine, as long as the bicycle does not leave the lane it is travelling in. Additionally, in rural areas the roads are typically spaced far apart so the assumption of at least 125 m of linear distance between corners is reasonable. This assumption allows the bicycle to sufficiently converge to the path between corners so that performance in subsequent corners does not degrade due to accumulation of error from multiple fast changes in yaw rate. However, a virtual path may connect to the actual path anywhere, which may temporarily degrade path-following performance.

To see the typical performance of the control system following a practical path, the following set of waypoints are defined:

$$\begin{aligned}\mathbf{W}_\rho &= \left\{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\175\end{bmatrix}, \begin{bmatrix}200\\175\end{bmatrix}, \begin{bmatrix}50\\450\end{bmatrix}, \begin{bmatrix}-100\\425\end{bmatrix}, \begin{bmatrix}-250\\425\end{bmatrix},\right.\\ &\left.\begin{bmatrix}-250\\275\end{bmatrix}, \begin{bmatrix}-450\\275\end{bmatrix}, \begin{bmatrix}-350\\150\end{bmatrix}, \begin{bmatrix}-200\\150\end{bmatrix}, \begin{bmatrix}-200\\0\end{bmatrix}\right\} \text{ m.}\end{aligned} \quad (5.19)$$

Following the logic that any right-hand turn has a turning radius of 8.85 m and a left turn will have a radius of 11.6 m the set of radii associated with this path are

$$\bar{\mathbf{r}} = \{11.6, 8.85, 11.6, 11.6, 8.85, 11.6, 8.85, 11.6, 11.6, 8.85, 11.6\} \text{ m.} \quad (5.20)$$

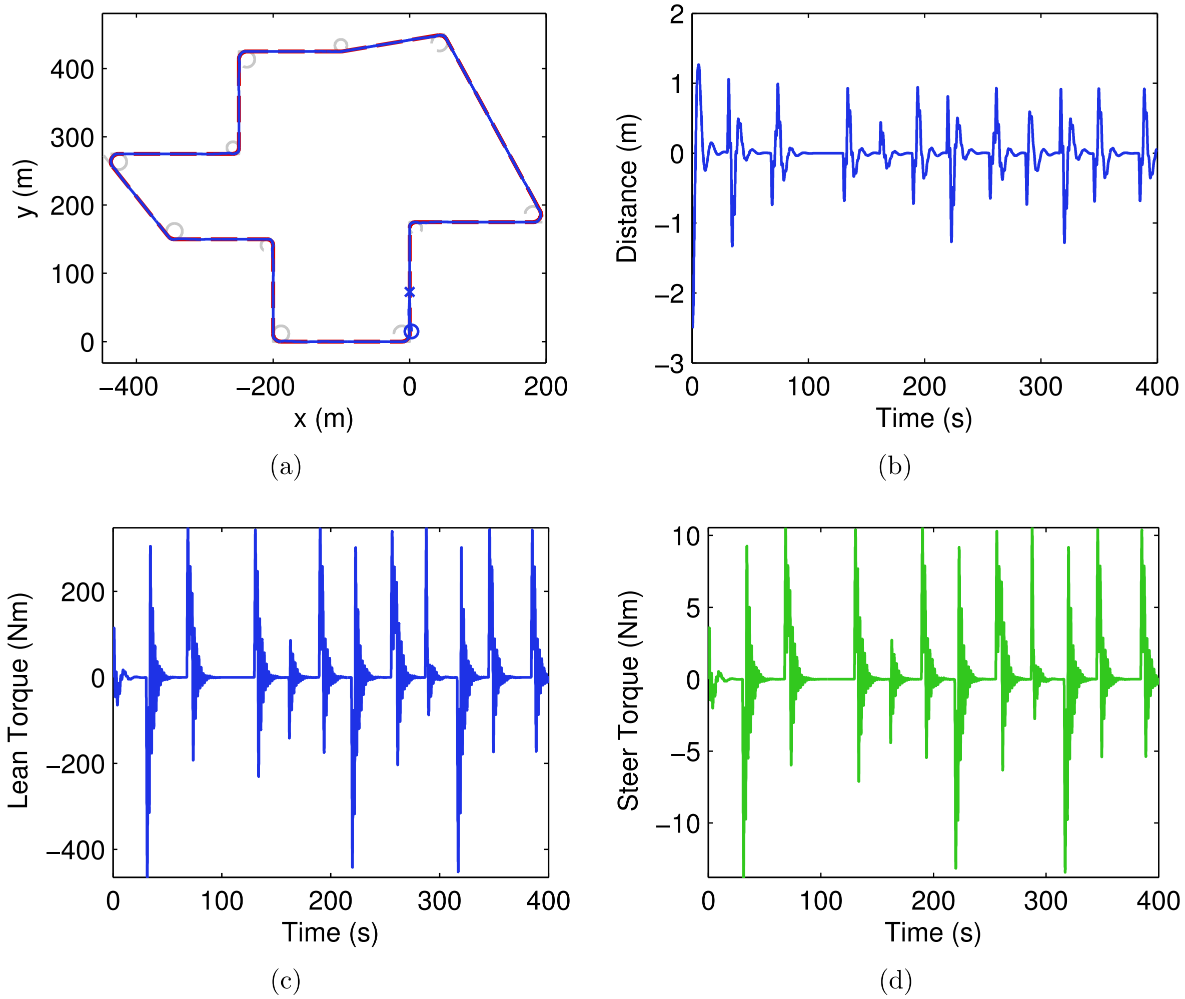
Defining l_{lj} as the linear distance between the circular arcs between $\mathbf{w}_{pj}$, and $\mathbf{w}_{p(j+1)}$ it follows that

$$l_{lj} = \left|\left|\mathbf{w}_{pj} - \mathbf{w}_{p(j+1)}\right|\right| - l_j - l_{j+1}, \quad (5.21)$$

where l_j is found from (1.4). The result of applying this equation to the path being constructed is

$$\mathbf{l}_l = \{155, 172, 286, 143, 138, 130, 167, 131, 136, 130, 177\} \text{ m.} \quad (5.22)$$

The path thus meets the assumed minimum linear distance between corners. The trajectory of a bicycle following this path and starting at $\mathbf{r}_B(0) = [2.5 \quad 15 \quad 0]^T$ m and $\gamma(0) = \frac{1}{2}\pi$ radians is shown in Figure 5.15(a). To evaluate the distance that the bicycle of the bicycle from the path a plot of d_ρ is also included in Figure 5.15(b). From this plot it is evident that after initially moving within the 2.7 m lane, that $d_\rho < 1.33$ m and thus remains within its intended lane. Looking at the values for τ_θ and τ_δ in Figures 5.15(c) and (d), respectively, as well as the value of θ and δ in Figures 5.16(a) and (b), respectively, it is apparent that there are considerable oscillations. As discussed previously there are options available for dealing with this; however, the bicycle remains upright while following the practical path, and it always stays within the lane of travel.



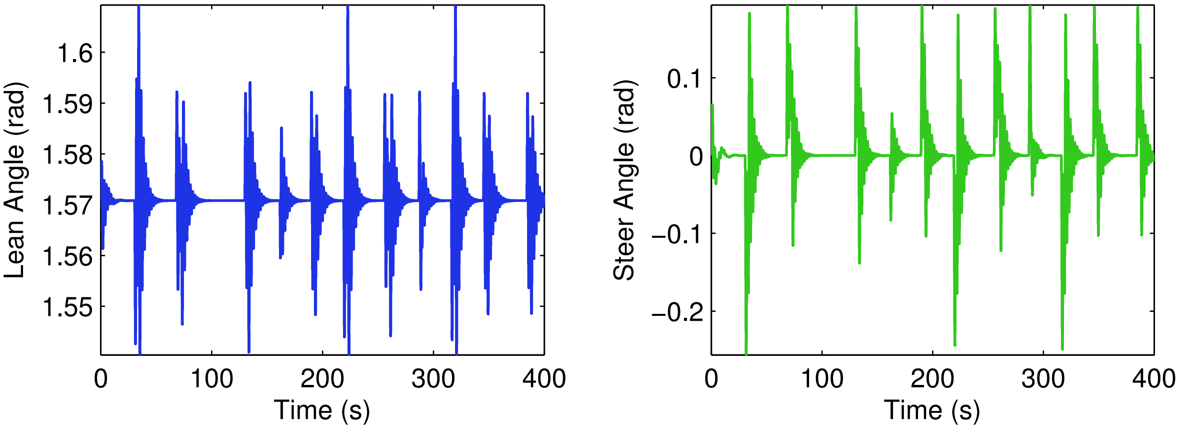
(a)

(b)

(c)

(d)

Figure 5.15: Bicycle with $\mathbf{r}_B(0) = [2.5 \quad 15 \quad 0]^T$ m and $\gamma(0) = \frac{1}{2}\pi$ radians following a practical path for 400 seconds. In Figure 5.15(a) the bicycle trajectory, blue, is shown tracking the closed path, red; starting at 'o' and ending at 'x' after completing one lap. The construction lines and circles for the path are given in grey. In Figure 5.15(b) d_ρ is shown with respect to time. Figures 5.15(c) and 5.15(d) show τ_θ and τ_δ, respectively, to achieve this trajectory.



(a)

(b)

Figure 5.16: Figure 5.15(c) shows θ and Figure 5.15(d) shows δ produced during the simulation in Figure 5.15.

Chapter 6

Conclusions

6.1 Summary

We have presented a systematic multi-loop control approach to stabilize a nonlinear model of a modern bicycle and asymptotically track paths of constant curvature. The model used is the nonlinear Whipple model with the dimensions and parameters listed in Appendix A.

The nonlinear Whipple model was previously derived using Lagrange mechanics by Basu-Mandal and associates [24] [25], and independently verified for accuracy. We also discuss a method of solution for the equations of motion, which are in the form of thirteen differential algebraic equations. The complexity of the model makes it impractical for direct analysis of the nonlinear equations, so we have linearized it to arrive at a parameterized linear time-invariant state-space representation. Analysis of the linear model has revealed that the behaviour of the model varies greatly with forward speed: it exhibits low forward speed and high forward speed unstable regions and a centre region where the bicycle is self-stable. We have also found that when controlling lean and steer angles the magnitude of the forward speed at which point B tracks across the ground is reduced over time. To keep the bicycle at a set speed we have used a proportional controller that provides high bandwidth control for $\dot{\beta}$. To control the lean and steer angles at this speed we used full-state feedback and augment integrators to the output feedback of θ and δ to guarantee perfect steady-state tracking. The feedback gains for this controller were chosen using LQR methods.

Once completing the inner loop, we continued the design with path-following control. We developed a map which changes a yaw angle reference into an appropriate reference on lean and steer angle for the inner loop. To prevent the bicycle from falling over we found in necessary to saturate the lean and steer angles for the inner loop, so we gave an approximation for the maximum yaw angle that can be tracked based on the saturation value for the steer angle reference. The next step in developing the path-following controller was to control the yaw rate since a constant curvature is linearly related to a yaw rate magnitude. We perform system identification on the map and inner-loop system which allowed us to develop a yaw-rate controller. To track a path we concluded that we not only want the yaw rate of the bicycle and path to be the same but also the yaw angle and distance. The yaw angle difference is the integral of the yaw rate, and we provided a linear approximation of the change in distance between the bicycle and the path.

Completing the linear approximation for the change in distance enabled construction of a linear approximation of the entire system. The next step was passing the signal for the difference in yaw angle and distance to the path through separate controllers and summing their output. Then a feedback loop was formed by subtracting the outcome from the path yaw rate, and using the result as the reference for the yaw rate control.

By using linear methods for control throughout the design process, we can readily analyze the system and tune the inner loop and path-following control to achieve the desired results. We can also conclude that the system will have perfect steady-state tracking with appropriate choices for the path-following controllers. Specifically this result was shown to be attainable when $C_d(s)$ contains an integrator, and we have also suggested that $C_{\dot{\gamma}}(s)$ contains one.

Finally we presented a design example for stabilization and path-following of a bicycle travelling with a forward speed of 5 m/s. In simulation we demonstrated the performance of the controller when the bicycle is close to the path and showed some issues that arise when it is far from the path. To overcome these problems we introduced the concept of a virtual path that the bicycle follows to return to the actual path. The performance of the system was then simulated on a practical path constructed from straight lines and circular arcs. This path was meant to represent a condensed closed path that may be followed by the bicycle when travelling on rural roads. From simulations conducted we concluded that in the absence of disturbance the bicycle would stay within the intended traffic lane, and not cross the centre line of a road or run onto the roadway shoulder.

6.2 Future Work

6.2.1 Improved Yaw-Rate Controller

The yaw-rate controller $C_{\dot{\gamma}}(s)$ that is used in the design example in Chapter 5 is a pure integral controller given in (5.17). When used in the control architecture in Figure 4.6, this yaw rate controller provides adequate performance for following practical paths, as discussed in Section 5.4. However, a high-gain integral controller was used for $C_{\dot{\gamma}}$ to keep the bicycle inside a 2.7 m lane. The gain used causes in oscillations in θ and δ, which may be undesirable in practice. To address this issue a different yaw rate controller can easily be implemented, and we demonstrate the results achieved by changing $C_{\dot{\gamma}}(s)$ to a proportional-integral (PI) controller. In this brief example the new $C_{\dot{\gamma}}(s)$ and remaining controllers are

$$C_{\dot{\gamma}}(s) = \frac{0.5s + 7.5}{s}, \; C_{\gamma}(s) = 0.55, \text{ and } C_d(s) = \frac{0.075s + 0.01}{s}. \tag{6.1}$$

Substituting these controllers and the plant from (5.8) into (4.23) and (4.33) gives the following stable transfer function for the system in Figure 4.6:

$$\frac{d_\rho(s)}{\dot{\lambda}(s)} = -\frac{5s^2(s + 1.37)}{s^5 + 2.13s^4 + 11.8s^3 + 6.52s^2 + 4.29s + 0.567}. \tag{6.2}$$

The result of simulating the bicycle with the controllers in (6.1) for the practical path by (5.19) and (5.19), and the same initial conditions as the bicycle in Figure 5.15, is shown in Figure 6.1. We thus have the same path and initial conditions as the example in Section 5.4 and can directly comparing Figures 5.15 and 5.16 to Figures 6.1 and 6.2. We see that once the bicycle is close to the path that the maximum magnitude of d_ρ, τ_θ, τ_δ, θ, and δ is reduced when the controllers in (6.1) are used in place of those in (5.17). Additionally the potentially unwanted oscillations in τ_θ, τ_δ, θ, and δ have been removed. Other controller choices for the inner-loop, $C_{\dot{\gamma}}$, C_γ and C_d may further improve performance.

6.2.2 Control Across Range of Forward Speed

As discussed in Chapter 3, the dynamics of the bicycle are highly dependent on forward speed. Independent of the inner loop, the path-following control from Chapter 4 is forward speed dependent. This implies that the stabilization and path-following control performance may be good only at velocities close to where the controller is designed. Other

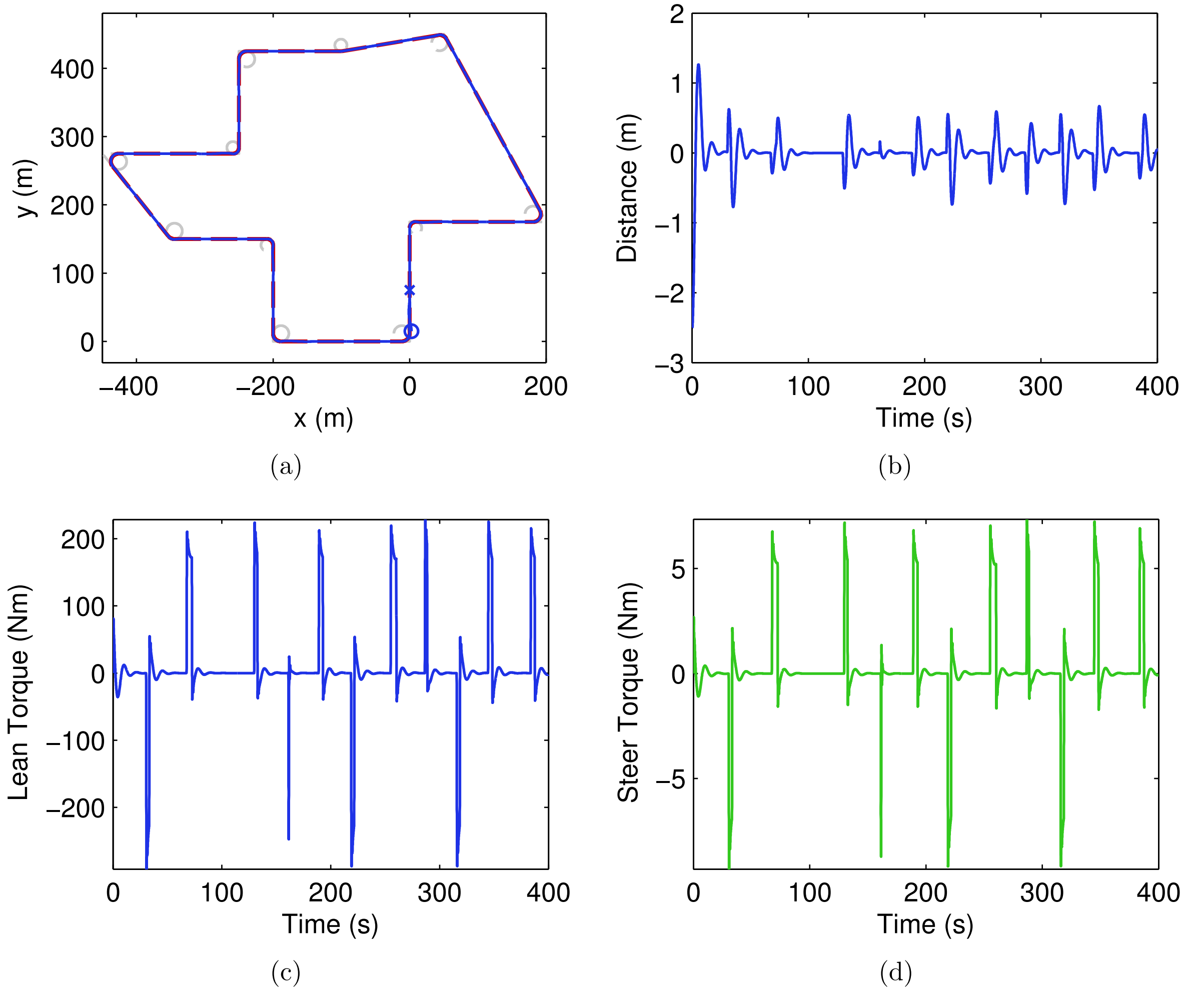
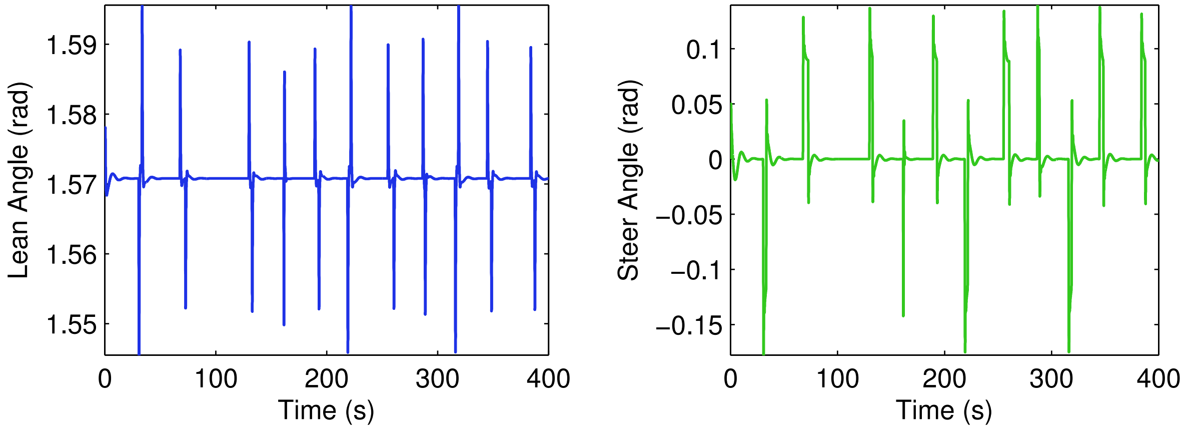


Figure 6.1: Bicycle with $\mathbf{r}_B(0) = [2.5 \;\; 15 \;\; 0]^T$ m and $\gamma(0) = \frac{1}{2}\pi$ radians following a practical path for 400 seconds. In (a) the bicycle trajectory, blue, is shown tracking the closed path, red; starting at 'o' and ending at 'x' after completing one lap. The construction lines and circles for the path are given in grey. In (b) d_ρ is shown with respect to time. Parts (c) and (d) show τ_θ and τ_δ, respectively, to achieve this trajectory.



(a)

(b)

Figure 6.2: Plot (c) shows θ and plot (d) shows δ produced during the simulation in Figure 6.1.

works have employed LPV control [28] and sliding-mode control [48] to balance and steer the bicycle at various forward speeds. An additional suggestion we provide is the use of gain scheduling for controllers designed at different velocities to control θ and δ.

Reviewing the linearized path-following transfer function (5.14) from the design example in Chapter 5 shows that the only denominator terms dependent on speed are also dependent on $C_d(s)$. Thus it is possible to use a LPV controller for $C_d(s)$ to provide stable tracking of constant curvature paths. Without more analysis no comment can be made on how slowly v must change.

Note that none of the methods discussed require a change in the control for $\dot{\beta}$ from Section 3.4, but the value of $\dot{\beta}$ is related to v by $v = \dot{\beta} r_{rw}$ when the bicycle is upright. Due to the high bandwidth suggested for the control of $\dot{\beta}$, it may be necessary to use a ramped $r_{\dot{\beta}}$ rather than a step to change v sufficiently slowly.

6.2.3 Investigation of Steer-Only Control and Lean-Only Control

Using lean-only or steer-only control presents an interesting problem. In Section 3.2 we present the parameterized linear time-invariant state-space system (3.21) for the bicycle, with states (3.19), input (3.20), and state-space matrices (3.22). In the case where $\mathbf{u}_L = \tau_\theta$ or $\mathbf{u}_L = \tau_\delta$, the system becomes single-input multiple-output (SIMO). In Section 3.2.2 we

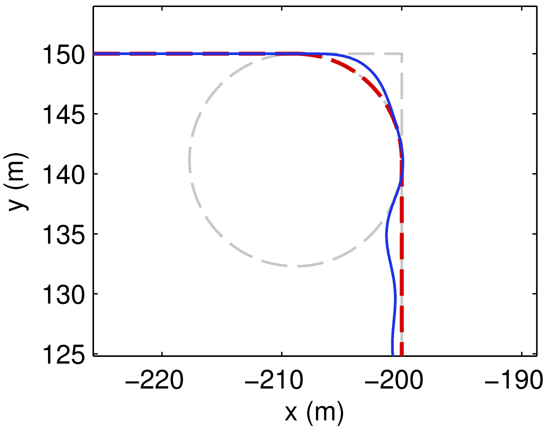


Figure 6.3: Magnification of a the bicycle trajectory, in blue, when following the path, red, through a right turn from Figure 5.15. The bicycle enters the figure from the left and leaves through the bottom.

discuss that the transfer function from each of τ_θ and τ_δ to δ contains a non-minimum phase zero. By using state feedback with integral tracking for δ there will be no transmission zeros in general. However, the transfer function from r_δ to δ will always have a non-minimum phase response. By rearranging and examining (4.6) we find that when δ experiences a non-minimum phase response so too will the yaw rate. This non-minimum phase response presents an interesting problem for path following. Without formal justification, initial testing has shown that asymptotic tracking can be achieved using the control architecture in Figure 4.6, and retuning the controller gains in (5.17).

6.2.4 Investigation of a Method for Path Preview

The path-following architecture presented relies purely on feedback. While we assume that all coordinates of the path as well as λ and $\dot{\lambda}$ are known, there is currently nothing in place to pre-emptively act when approaching a corner. This technique provides adequate tracking; however, Figure 6.3 shows that as the bicycle enters a corner it continues tracking straight briefly before correcting. Similarly, when exiting a corner, the bicycle continues tracking the circle briefly before correcting. This behaviour motivates the that we may be able to reduce the maximum value of d_ρ in each corner by using path preview since these maxima occur immediately following the transition into and out of corners.

6.2.5 Analysis of Control System Robustness

During our analysis we treat the rider as being rigid and assume that there is no slip between the wheels of the bicycle and the ground. In the event that these assumptions break down are inaccurate it is desirable that control system causes the bicycle to still asymptotically track a constant curvature path while maintaining stability. In an application the bicycle may also need to stay within a certain distance of a path such as in our practical path example. To perform such an analysis a more complicated nonlinear bicycle model must be derived and used for testing. Also of interest on the current model is the performance of the control system in the presence of modelling errors, disturbances such as wind, or when introducing sensor noise into the system.

APPENDICES

Appendix A

Bicycle Parameter Values

Parameter	**Value**
g	9.81 m/s^2
h	0.08 m
$\tilde{\mathbf{I}}_{G,rf,ref}$	$\begin{bmatrix} 9.2 & 2.4 & 0 \\ 2.4 & 2.8 & 0 \\ 0 & 0 & 11 \end{bmatrix}$ kg m^2
$\tilde{\mathbf{I}}_{K,ff,ref}$	$\begin{bmatrix} 0.0589 & -0.00756 & 0 \\ -0.00756 & 0.00708 & 0 \\ 0 & 0 & 0.06 \end{bmatrix}$ kg m^2
$\tilde{\mathbf{I}}_{L,fw,ref}$	$\begin{bmatrix} 0.141 & 0 & 0 \\ 0 & 0.141 & 0 \\ 0 & 0 & 0.28 \end{bmatrix}$ kg m^2
$\tilde{\mathbf{I}}_{P,rw,ref}$	$\begin{bmatrix} 0.141 & 0 & 0 \\ 0 & 0.141 & 0 \\ 0 & 0 & 0.28 \end{bmatrix}$ kg m^2
m_{ff}	4 kg
m_{fw}	3 kg
m_{rf}	85 kg
m_{rw}	3 kg

r_{fw}	0.35 m
$\mathbf{r}_{G/P,ref}$	$\begin{bmatrix}0.3 & -0.55 & 0\end{bmatrix}^T$ m
$\mathbf{r}_{K/S,ref}$	$\begin{bmatrix}0.0219 & -0.0171 & 0\end{bmatrix}^T$ m
$\mathbf{r}_{L/S,ref}$	$\begin{bmatrix}0.142 & 0.333 & 0\end{bmatrix}^T$ m
$\mathbf{r}_{P/S,ref}$	$\begin{bmatrix}-0.878 & 0.333 & 0\end{bmatrix}^T$ m
r_{rw}	0.35 m
$\mathbf{r}_{S/G,ref}$	$\begin{bmatrix}0.578 & 0.217 & 0\end{bmatrix}^T$ m
v	various m/s
w	1.02 m
μ	$\pi/10$ rad

Table A.1: Bicycle parameter values.

References

[1] A. Sharp, *Bicycles and Tricycles: An Elementary Treatise on Their Design and Construction, with Examples and Tables.* White Plains, USA: Longmans, Green, and Company, 1896.

[2] D. Limebeer and R. Sharp, “Bicycles, motorcycles and models,” *IEEE Control Systems Magazine, Special issue on Advances in Motorcycle Design*, vol. 26, pp. 34–61, October 2008.

[3] F. Whitt and D. Wilson, *Bicycling Science.* The Massachusetts Institute of Technology, second edition ed., 1982.

[4] D. Bickford and D. Davison, “Systematic multi-loop control for autonomous bicycle path following,” in *Electrical and Computer Engineering, 2013. CCECE 2013. 26th Annual Canadian Conference on*, (Regina, Canada), May 2013.

[5] L. Dubins, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,” *American Journal of Mathematics*, vol. 79, no. 3, pp. pp. 497–516, 1957.

[6] A. A. of State Highway and T. Officials, *A Policy on Geometric Design of Highways and Streets 2011.* 444 North Capitol Street, NW, Suite 249, Washington, DC: American Association of State Highway and Transportation Officials, 2011.

[7] W. Rankine, “On the dynamical principles of the motion of velocipedes,” *The Engineer*, vol. 28, pp. 7,129,153,175, 1869.

[8] E. Carvallo, “Théorie du mouvement du monocycle et de la bicyclette,” *J. de L’Ecole Polytechnique*, vol. 5, pp. 119–188, 1900.

[9] F. Whipple, “The stability of the motion of a bicycle,” *The Quarterly Journal of Pure and Applied Mathematics*, vol. 30, no. 120, pp. 312–348, 1899.

[10] R. Hand, "Comparisons and stability analysis of linearized equations of motion for a basic bicycle model," Master's thesis, Cornell University, Ithica, USA, 1988.

[11] J. Meijaard, J. Papadopoulos, A. Ruina, and A. Schwab, "Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review," *Proc. R. Soc. A*, vol. 463, pp. 1955–1982, August 2007.

[12] E. Döhring, "Stability of single-track vehicles," *Forschung Ing.-Wes.*, vol. 21, no. 2, pp. 50–62, 1955.

[13] J. Neǐmark and N. Fufaev, "Dynamics of nonholonomic systems," *American Mathematical Society Translations of Mathematical Monographs*, vol. 33, pp. 330–374, 1972.

[14] J. Papadopoulos, "Bicycle steering dynamics and self-stability: A summary report on work in progress," tech. rep., Cornell University, Ithica, USA, 1987.

[15] K. Aström, R. Klein, and A. Lennartsson, "Bicycle dynamics and control, adapted bicycles for education and research," *IEEE Control Systems Mag.*, vol. 25, pp. 26–47, August 2005.

[16] A. Schwab, J. Meijaard, and J. Kooijman, "Lateral dynamics of a bicycle with a passive rider model: stability and controllability," *Vehicle System Dynamics*, vol. 50, pp. 1209–1224, August 2012.

[17] G. Frosali and F. Ricci, "Kinematics of a bicycle with toroidal wheels," *Communications in Applied and Industrial Mathematics*, vol. 3, no. 1, 2012.

[18] C. Koenen, *The Dynamic Behavior of a Motorcycle when running straight ahead and when cornering.* PhD thesis, Delft University of Technology, Delft, Netherlands, 1983.

[19] R. Sharp, "Stability, control and steering responses of motorcycles," *Vehicle System Dynamics*, vol. 35, pp. 291–318, August 2001.

[20] R. Sharp, S. Evangelou, and D. Limebeer, "Advances in the modelling of motorcycle dynamics," *Multibody System Dynamics*, vol. 12, pp. 251–283, October 2004.

[21] R. Sharp, "The lateral dynamics of motorcycles and bicycles," *Vehicle System Dynamics*, vol. 14, no. 4-6, pp. 265–283, 1985.

[22] A. Lennartsson, *Efficient Multibody Dynamics.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1999.

[23] N. Getz, "Control of balance for a nonlinear nonholonomic non-minimum phase model of a bicycle," in *American Control Conference*, pp. 5204–5209, 1994.

[24] P. Basu-Mandal, A. Chatterjee, and J. Papadopoulos, "Hands-free cirular motions of a benchmark bicycle," *Proc. R. Soc. A*, vol. 463, pp. 1983–2003, August 2007.

[25] P. Basu-Mandal, *Studies on the Dynamics and Stability of Bicycles*. PhD thesis, Indian Institute of Technology, Bangalore, India, 2007.

[26] J. Moore, *Human Control of a Bicycle*. PhD thesis, University of California at Davis, Davis, USA, 2012.

[27] T. Kane and L. D.A., *Dynamics, Theory and Applications*. New York, USA: McGraw-Hill, 1985.

[28] D. Andreo, V. Cerone, D. Dzung, and D. Regruto, "Experimental results on LPV stabilization of a riderless bicycle," in *2009 American Control Conference*, (St. Louis, USA), pp. 3124–3129, June 2009.

[29] J. Mutsaerts, "Ntxbike-gs," tech. rep., Delft University of Technology, Delft, Netherlands, May 2010.

[30] C. Findlay, J. Moore, and C. Perez-Maldonado, "SISO control of a bicycle-rider system," in *MAE 272 Report 2*, 2006.

[31] P. Lam, "Gyroscopic stabilization of a kid-size bicycle," in *Cybernetica and Intelligend Systems, 2011 IEEE 5th International Conference on*, (Qingdao, China), pp. 247–252, 2011.

[32] C. Hwang, H. Wu, and C. Shih, "Autonomous dynamic balance of an electrical bicycle using variable structure under-actuated control," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Nice, France), pp. 3737–3743, September 2008.

[33] C. Hwang, H. Wu, and C. Shih, "Fuzzy sliding-mode under-actuated control for autonomous dynamic balance of an electric bicycle," in *2008 IEEE International Conference on Fuzzy Systems*, (Hong Kong, China), pp. 251–257, June 2008.

[34] X. Yu, S. Wei, and L. Guo, "Cascade sliding mode control for bicycle robot," in *Artificial Intelligence and Computational Intelligence, 2010 International Conference on*, vol. 1, (Sanya, China), pp. 62–66, Oct. 2010.

[35] T. Dao and C. Chen, "Sliding-mode control for the roll-angle tracking of an unmanned bicycle," *Vehicle System Dynamics*, vol. 49, pp. 915–930, June 2011.

[36] C. Chen and T. Dao, "Fuzzy control for equilibrium and roll-angle tracking of an unmanned bicycle," *Multibody System Dynamics*, vol. 15, pp. 321–346, May 2006.

[37] D. Weir, *Motorcycle Handling Dynamics and Rider Control and the Effect of Design Configuration on Response and Performance.* PhD thesis, University of California, Los Angeles, USA, 1972.

[38] N. Getz and J. Marsden, "Control for an autonomous bicycle," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1397–1402, May 1995.

[39] R. Frezza, A. Beghi, and A. Saccon, "Model predictive for path following with motorcycles: application to the development of the pilot model for virtual prototyping," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 1, (Nassau, Bahamas), pp. 767–772, December 2004.

[40] L. Consolini and M. Maggiore, "Control of a bicycle using virtual holonomic constraints," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, (Atlanta, USA), pp. 5204–5209, December 2010.

[41] J. Yi, D. Song, A. Levandowski, and S. Jayasuriya, "Trajectory tracking and balance stabilization control of autonomous motorcycles," in *International Conference on Robotics and Automation, Proceedings of the 2006*, (Orlando, USA), pp. 2583–2589, May 2006.

[42] N. Getz, *Dynamic Inversion of Nonlinear Maps with Applications to Nonlinear Control and Robotics.* PhD thesis, University of California, Berkeley, USA, 1995.

[43] J. Hauser, A. Saccon, and R. Frezza, "Achieveable motorcycle trajectories," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 4, pp. 3944–3949, December 2004.

[44] J. Hauser and A. Saccon, "Motorcycle modeling for high-performance maneuvering," *Control Systems, IEEE*, vol. 26, pp. 89–105, October 2006.

[45] M. Yamakita and A. Utano, "Automatic control of bicycles with a balancer," in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, (Monterey, USA), pp. 1245–1250, July 2005.

[46] L. Keo and Y. Masaki, "Trajectory control for an autonomous bicycle with balancer," in *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, (Xian, China), pp. 676–681, July 2008.

[47] R. Sharp, "Optimal stabilization and path-following controls for a bicycle," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 221, no. 4, pp. 415–427, 2007.

[48] C. Chen and T. Dao, "Speed-adaptive path-following control of a riderless bicycle via road preview," in *Bicycle and Motorcycle Dynamics 2010, Symposium on the Dynamics and Control of Single Track Vehicles*, (Delft, The Netherlands), October 2010.

[49] D. Peterson and M. Hubbard, "Yaw rate and velocity tracking control of a hands-free bicycle," in *ASME 2008 International Mechanical Engineering Congress and Exposition*, (Boston, USA), October 2008.

[50] J. Kooijman, "Experimental validation of a model for the motion of an uncontrolled bicycle," Master's thesis, Delft University of Technology, Delft, Netherlands, 2006.

[51] S. Yoon, R. Howe, and D. Greenwood, "Constraint violation stabilization using gradient feedback in constrained dynamics simulation," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 6, pp. 1467–1474, 1992.

[52] R. Kalman, "Contributions to the theory of optimal control," *Boletin de la Sociedad Matematica Mexicana*, vol. 5, pp. 102–19, 1960.

[53] G. Rao and H. Unbehauen, "Identification of continuous-time systems," *Control Theory and Applications, IEE Proceedings*, vol. 153, no. 2, pp. 185–220, 2006.

[54] E. Carvallo, "Théorie du movement du monocycle. part 2: Théorie de la bicyclette," *J. de L'Ecole Polytechnique*, vol. 6, pp. 1–118, 1901.