

---

# Self-Balancing Bicycle

Final Project

---

Philip Salmony  
Wolfson College  
pms67@cam.ac.uk

Supervisor: Dr. Fulvio Forni



4 September 2018

## Summary

Modelling, analysis, simulation, and implementation of an autonomous, self-balancing bicycle.

# Contents

<b>1</b>	<b>Introduction and Objectives</b>	<b>3</b>
<b>2</b>	<b>Notation</b>	<b>3</b>
<b>3</b>	<b>Bicycle Modelling</b>	<b>3</b>
3.1	Mass and Center of Mass . . . . .	3
3.2	Inertia Tensor . . . . .	4
3.3	Overall Description . . . . .	4
<b>4</b>	<b>Equations of Motion</b>	<b>5</b>
4.1	Second Order Model . . . . .	5
4.1.1	General . . . . .	5
4.2	Fourth Order Model . . . . .	6
4.2.1	Decoupling and Transfer Functions . . . . .	7
4.3	State Space Form . . . . .	8
<b>5</b>	<b>Controller Design</b>	<b>8</b>
5.1	Stability Analysis . . . . .	8
5.1.1	Dependence on Forward Speed . . . . .	8
5.2	Single Input Single Output (SISO) . . . . .	9
5.2.1	Proportional Controller . . . . .	9
5.2.2	P-D Controller . . . . .	10
5.3	State Feedback . . . . .	12
5.4	Optimal Control . . . . .	12
5.5	Zero-Speed Stabilisation . . . . .	12
<b>6</b>	<b>Trajectory Tracking</b>	<b>12</b>
<b>7</b>	<b>Simulation</b>	<b>12</b>
7.1	MatLAB . . . . .	12
7.2	Unity . . . . .	12
<b>8</b>	<b>Implementation</b>	<b>12</b>
<b>9</b>	<b>Results and Discussion</b>	<b>13</b>
<b>10</b>	<b>Conclusions</b>	<b>13</b>
<b>11</b>	<b>References</b>	<b>13</b>
<b>12</b>	<b>Appendix</b>	<b>13</b>

# 1 Introduction and Objectives

1. Autonomous self-balancing at close-to-zero forward velocity and above.
2. Robustness to modelling inaccuracies as well as to mass and geometry changes.
3. Trajectory tracking.

# 2 Notation

Coordinate systems, notation(e.g.  $m_{rf}$ ,  $I_{xx}$ , etc..)

# 3 Bicycle Modelling

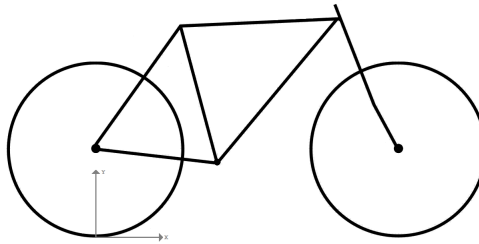


Figure 1: Simplified Bike Model and Coordinate System

## 3.1 Mass and Center of Mass

The purchased bike was disassembled and the masses of both wheels and the remaining frame measured using a luggage scale. To estimate the masses of the rear and front frames, the dimensions of each connecting tube of the frames was measured to approximate their respective volumes. As mass is proportional to volume, this enabled an estimate of the individual masses of the tubes. Summing the tubes belonging to the appropriate frame gave an approximation of the mass of that frame. The results are as follows:

- $m_{rf} = 7.8kg$
- $m_{ff} = 2.1kg$
- $m_{rw} = 3.1kg$
- $m_{fw} = 2.5kg$

### Rear Frame

$$(x, y, z)_{rf} = [0.507, 0.000, -0.534]$$

### Front Frame

$$(x, y, z)_{ff} = [0.960, 0.000, -0.597]$$

### Rear Wheel

$$(x, y, z)_{rw} = [0.000, 0.000, -0.320]$$

### Front Wheel

$$(x, y, z)_{ff} = [1.080, 0.000, -0.320]$$

### 3.2 Inertia Tensor

The moment of inertia tensor describes the resistance to rotational motion of a three-dimensional object around certain axes about its center of mass. It is defined as follows,

$$\mathbf{I} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix}.$$

We make the simplifying assumption that the terms  $I_{xz}$ ,  $I_{zx}$ ,  $I_{yz}$ ,  $I_{zy}$  are zero. The definitions for the remaining terms are,

$$\begin{aligned} I_{xx} &= \int (y^2 + z^2) dm \\ I_{yy} &= \int (x^2 + z^2) dm \\ I_{zz} &= I_{xx} + I_{yy} \\ I_{xy} &= I_{yx} = - \int (x \cdot y) dm \end{aligned}$$

**Rear Frame**

$$\mathbf{I}_{\mathbf{rf}} = \begin{pmatrix} 0.206 & 0 & 0.164 \\ 0 & 0.643 & 0 \\ 0.164 & 0 & 0.444 \end{pmatrix}$$

**Front Frame**

$$\mathbf{I}_{\mathbf{ff}} = \begin{pmatrix} 0.061 & 0 & -0.017 \\ 0 & 0.059 & 0 \\ -0.017 & 0 & 0.013 \end{pmatrix}$$

**Rear Wheel**

$$\mathbf{I}_{\mathbf{rw}} = \begin{pmatrix} 0.159 & 0 & 0 \\ 0 & 0.317 & 0 \\ 0 & 0 & 0.159 \end{pmatrix}$$

**Front Wheel**

$$\mathbf{I}_{\mathbf{fw}} = \begin{pmatrix} 0.128 & 0 & 0 \\ 0 & 0.256 & 0 \\ 0 & 0 & 0.128 \end{pmatrix}$$

### 3.3 Overall Description

**Total System**

*Mass and Center of Mass*

$$m_t = m_{rw} + m_{rf} + m_{ff} + m_{fw} = 15.5 \quad (1)$$

$$x_t = \frac{1}{m_t} (x_{rf}m_{rf} + x_{ff}m_{ff} + x_{fw}m_{fw}) = 0.559 \quad (2)$$

$$z_t = \frac{1}{m_t} (-R_{rw}m_{rw} + z_{rf}m_{rf} + z_{ff}m_{ff} - R_{fw}m_{fw}) = -0.465 \quad (3)$$

### Moments and Products of Inertia

$$T_{xx} = A_{xx} + B_{xx} + C_{xx} + D_{xx} + m_{rw}R_{rw}^2 + m_{rf}z_{rf}^2 + m_{ff}z_{ff}^2 + m_{fw}R_{fw}^2 = 4.10 \quad (4)$$

$$T_{xz} = B_{xz} + C_{xz} - m_{rf}x_{rf}z_{rf} - m_{ff}x_{ff}z_{ff} + m_{fw}wR_{fw} = 4.33 \quad (5)$$

$$T_{zz} = A_{zz} + B_{zz} + C_{zz} + D_{zz} + m_{rf}x_{rf}^2 + m_{ff}x_{ff}^2 + m_{fw}w^2 = 7.60 \quad (6)$$

### Front Frame

#### Mass and Center of Mass

$$m_f = m_{ff} + m_{fw} = 4.6 \quad (7)$$

$$x_f = \frac{1}{m_f}(x_{ff}m_{ff} + w m_{fw}) = 1.025 \quad (8)$$

$$z_f = \frac{1}{m_f}(z_{ff}m_{ff} - R_{fw}m_{fw}) = -0.446 \quad (9)$$

### Moments and Products of Inertia

$$F_{xx} = C_{xx} + D_{xx} + m_{ff}(z_{ff} - z_f)^2 + m_{fw}(R_{fw} + z_f)^2 = 0.277 \quad (10)$$

$$F_{xz} = C_{xz} - m_{ff}(x_{ff} - x_f)(z_{ff} - z_f) + m_{fw}(w - x_f)(R_{fw} + z_f) = -0.0549 \quad (11)$$

$$F_{zz} = C_{zz} + D_{zz} + m_{ff}(x_{ff} - x_f)^2 + m_{fw}(w - x_f)^2 = 0.157 \quad (12)$$

### Miscellaneous

$$\lambda = \frac{\pi}{2} - \alpha = \frac{\pi}{9} \quad (13)$$

$$u = (x_f - w - t) \cdot \cos(\lambda) - z_f \cdot \sin(\lambda) = 0.021 \quad (14)$$

$$S_r = \frac{A_{yy}}{R_{rw}} = 0.991, S_f = \frac{D_{yy}}{R_{fw}} = 0.800, S_t = S_r + S_f = 1.79 \quad (15)$$

## 4 Equations of Motion

### 4.1 Second Order Model

#### 4.1.1 Excluding Fork Dynamics

Simplified tilt equation:

$$\ddot{\theta} = \frac{Mgl}{J} \sin(\theta) + \frac{Ml}{J} \left( \frac{v_r^2}{b} \tan(\phi) + \frac{a}{b} \frac{d(v \tan(\phi))}{dt} \right) \cos(\theta) \quad (16)$$

For near-upright motion with constant velocity  $v$ :  $\dot{v} = 0$ ,  $\sin(\theta) \approx \theta$ ,  $\tan(\phi) \approx \phi$ , and  $\cos(\theta) \approx 1$ . We can therefore linearise the equation as follows:

$$\ddot{\theta} = \frac{Mgl}{J} \theta + \frac{Ml}{J} \left( \frac{v^2}{b} \phi + \frac{a}{b} v \dot{\phi} \right) \quad (17)$$

Defining  $\gamma = \frac{Ml}{Jb}$ :

$$\ddot{\theta} = bg\gamma\theta + \gamma(v_r^2\phi + av\dot{\phi})$$

Taking Laplace transforms:

$$s^2\bar{\theta} = bg\gamma\bar{\theta} + \gamma(v_r^2\bar{\phi} + sav\bar{\phi})$$

$$\bar{\theta}(s^2 - bg\gamma) = \bar{\phi}\gamma v(sa + v)$$

Which results in the transfer function  $G(s)$  from steer angle  $\phi$  to lean angle  $\theta$ :

$$G(s) = \gamma v a \frac{(s + \frac{v}{a})}{(s^2 - b g \gamma)} \quad (18)$$

We can extract three bits of information:

1. Gain:  $k = \frac{M l v a}{J b}$
2. Zero:  $z = -\frac{v}{a}$
3. Poles:  $p_{1,2} = \pm \sqrt{\frac{M l g}{J}}$

The gain tells us that as our forward velocity  $v$  goes to zero, turning the forks will not influence the tilt angle.

The zero tells us that this is a minimum phase system.

The poles tell us that this system is unstable and non-oscillatory.

For the prototype bicycle the transfer function becomes,

$$G(s) = 0.491 \cdot v \cdot \frac{(s + \frac{v}{0.559})}{(s^2 - 9.30)}. \quad (19)$$

#### 4.1.2 Including Fork Dynamics

Shows stability for large enough forward speed...

### 4.2 Fourth Order Model

A model found in many research papers, is the fourth order linearised model consisting of two coupled second order differential equations. This model can be written in the following matrix form:

$$\mathbf{M} \cdot \ddot{\underline{q}} + [\mathbf{C}_1 \cdot v] \cdot \dot{\underline{q}} + [\mathbf{K}_0 + \mathbf{K}_2 \cdot v^2] \cdot \underline{q} = \underline{f} \quad (20)$$

Where  $\underline{q} = [\phi, \delta]^T$  is a vector containing the lean angle ( $\phi$ ) and the steer angle ( $\delta$ ),  $\underline{f} = [T_\phi, T_\delta]^T$  is a vector consisting of a roll torque disturbance and handlebar torque, and finally the system matrices, which are derived from the bicycle geometry:

1.  $\mathbf{M}$  - Mass matrix
2.  $\mathbf{C}_1$  - Damping matrix, varying *linearly* with velocity
3.  $\mathbf{K}_0$  and  $\mathbf{K}_2$  - Stiffness matrices, varying *quadratically* with velocity

The elements of each matrix are as follows:

$$\mathbf{M} = \begin{bmatrix} T_{xx} & F_{\lambda x} + f T_{xz} \\ F_{\lambda x} + f T_{xz} & F_{\lambda \lambda} + 2f F_{\lambda z} + f^2 T_{zz} \end{bmatrix} \quad (21)$$

$$\mathbf{C}_1 = \begin{bmatrix} 0 & f S_t + S_f \cos(\lambda) + T_{xz} \frac{\cos(\lambda)}{w} - f m_t z_t \\ -(f S_t + S_f \cos(\lambda)) & F_{\lambda z} \frac{\lambda}{w} + f(S_u + T_{zz} \frac{\cos(\lambda)}{w}) \end{bmatrix} \quad (22)$$

$$\mathbf{K}_0 = \begin{bmatrix} g m_t z_t & -g S_u \\ -g S_u & -g S_u \sin(\lambda) \end{bmatrix} \quad (23)$$

$$\mathbf{K}_2 = \begin{bmatrix} 0 & (S_t - m_t z_t) \frac{\cos(\lambda)}{w} \\ 0 & (S_u + S_f \sin(\lambda)) \frac{\cos(\lambda)}{w} \end{bmatrix} \quad (24)$$

#### 4.2.1 Decoupling and Transfer Functions

Taking Laplace Transforms, assuming zero initial conditions:

$$s^2 \cdot \mathbf{M} \cdot \underline{\bar{q}} + s \cdot [\mathbf{C}_1 \cdot v] \cdot \underline{\bar{q}} + [\mathbf{K}_0 + \mathbf{K}_2 \cdot v^2] \cdot \underline{\bar{q}} = \underline{\bar{f}}$$

Rearranging into transfer function form yields:

$$\mathbf{G}(s) = \left( s^2 \cdot \mathbf{M} + s \cdot \mathbf{C}_1 \cdot v + (\mathbf{K}_0 + \mathbf{K}_2 \cdot v^2) \right)^{-1} \quad (25)$$

Since  $K_{211} = K_{221} = C_{111} = 0$ , the determinant of  $\mathbf{G}(s)$  is

$$\begin{aligned} \det(\mathbf{G}(s)) = P(s) = & s^4 \cdot (M_{11}M_{22} - M_{12}M_{21}) \\ & + s^3 \cdot v \cdot (M_{11}C_{122} - (M_{12}C_{121} + M_{21}C_{112})) \\ & + s^2 \cdot (M_{11}K_{022} + M_{22}K_{011} - M_{21}K_{012} - M_{12}K_{021}) + v^2 \cdot (M_{11}K_{222} - M_{21}K_{212} - C_{112}C_{121}) \\ & + s \cdot v \cdot (K_{011}C_{122} - K_{021}C_{112} - K_{012}C_{121} - v^2 \cdot K_{212}C_{121}) \\ & + (K_{011}K_{022} - K_{021}K_{012}) + v^2 \cdot (K_{011}K_{222} - K_{021}K_{212}). \end{aligned}$$

Where the roots of  $P(s)$  are equal to the poles of the system.

As can be seen from the long expression above, it is hard to directly see what parameters in the bicycle's geometry - or the forward speed - influence the stability. This is especially true, since this is a linearised model of a much more complex system!

Continuing with the evaluation of the matrix inverse, we get a set of four decoupled transfer functions relating the roll and steer torque inputs to the roll and steer angular outputs:

(i) *Roll Torque Disturbance to Lean Angle*

$$G_{11}(s) = \frac{m_{22} \cdot s^2 + v \cdot C_{122} \cdot s + (K_{022} + K_{222} \cdot v^2)}{P(s)}$$

(ii) *Roll Torque Disturbance to Steer Angle*

$$G_{12}(s) = -\frac{m_{12} \cdot s^2 + v \cdot C_{112} \cdot s + (K_{012} + K_{212} \cdot v^2)}{P(s)}$$

(iii) *Handlebar Torque to Lean Angle*

$$G_{21}(s) = -\frac{m_{21} \cdot s^2 + v \cdot C_{121} \cdot s + K_{021}}{P(s)}$$

(iv) *Handlebar Torque to Steer Angle*

$$G_{22}(s) = \frac{m_{11} \cdot s^2 + K_{011}}{P(s)}$$

Again, it is hard to see how the bicycle parameters influence the positions of the systems zeros.

For the prototype bicycle the poles  $P(s)$  are then given by,

$$P(s) = 0.711s^4 + 1.30vs^3 + s^2(-22.3 + 5.01v^2) + sv(-21.4 + 6.93v^2) + (123 - 5.54v^2) \quad (26)$$

Which leads to following the set of transfer functions:

(i) Roll Torque Disturbance to Lean Angle

$$G_{11}(s) = \frac{0.214 \cdot s^2 + v \cdot 0.744 \cdot s + (-2.48 + 0.881 \cdot v^2)}{P(s)} \quad (27)$$

(ii) Roll Torque Disturbance to Steer Angle

$$G_{12}(s) = -\frac{0.407 \cdot s^2 + v \cdot 5.18 \cdot s + (-7.25 + 7.83 \cdot v^2)}{P(s)} \quad (28)$$

(iii) Handlebar Torque to Lean Angle

$$G_{21}(s) = -\frac{0.407 \cdot s^2 - 0.884 \cdot v \cdot s - 7.25}{P(s)} \quad (29)$$

(iv) Handlebar Torque to Steer Angle

$$G_{22}(s) = \frac{4.10 \cdot s^2 - 70.7}{P(s)} \quad (30)$$

### 4.3 State Space Form

Choosing states as  $\underline{x} = [\phi \ \delta \ \dot{\phi} \ \dot{\delta}]^T$  and inputs as  $\underline{u} = [T_\phi \ T_\delta]^T$ , we arrive at the following state space model for our prototype bicycle (assuming all our states are measurable):

$$\begin{aligned} \dot{\underline{x}} &= \mathbf{A}\underline{x} + \mathbf{B}\underline{u} \\ \underline{y} &= \mathbf{C}\underline{x} \end{aligned}$$

With the system matrices being,

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 17.1 & 0.761 - 1.85v^2 & -0.506v & -1.13v \\ 1.36 & 10.2 - 0.602v^2 & 5.10v & -1.32v \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.300 & -0.572 \\ -0.572 & 5.77 \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (31)$$

## 5 Controller Design

What can we actuate? What sensor do we have? Controllability/Observability

### 5.1 Stability Analysis

#### 5.1.1 Dependence on Forward Speed

Three controller regimes.



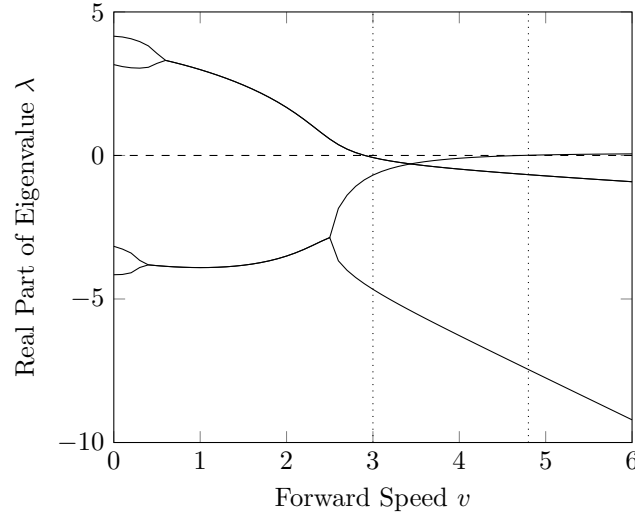


Figure 2: Pole Locations vs Forward Speed

## 5.2 Single Input Single Output (SISO)

### CURRENTLY EXCLUDING FORK DYNAMICS - WILL CHANGE

Which transfer function model do we use? Second order, fourth order, or both? BOTH

#### 5.2.1 Proportional Controller

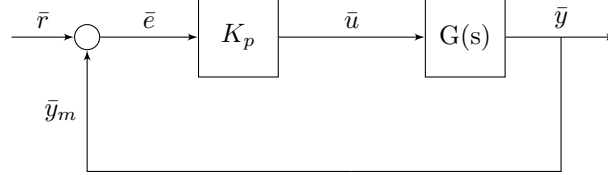


Figure 3: P Controller Structure

#### Transfer Function

Plotting the root locus diagram for the tilt transfer function, we can see that a simple proportional controller ( $K(s) = K_p$ ) in a standard negative feedback configuration can stabilise this plant. The closed-loop transfer function  $P(s)$  is of the following form:

$$P(s) = K_p \cdot (\gamma v a) \cdot \frac{(s + \frac{v}{a})}{s^2 + (\gamma v a) K_p s + \gamma(v_r^2 K_p - b g)} \quad (32)$$

Using the following root locus relationship to find the required proportional gain:

$$K_p = \frac{1}{|G(s)|} \quad (33)$$

We are concerned with finding the minimum proportional gain that will guarantee closed-loop stability, therefore we wish to find a gain that will give us  $s < 0$ . Therefore, at  $s = 0$ ,  $K_p = \frac{b g}{v_r^2}$ , which is the proportional gain needed for marginal stability. We require a gain larger than this to ensure asymptotic stability.

The furthest left we can go in the s-plane by increasing  $K_p$  is when our closed-loop pole meets the open-loop zero at  $\frac{v}{a}$ , which gives an exponential decay time constant of  $\frac{a}{v}$ .

### Prototype Bicycle

For the prototype bicycle, the closed-loop transfer function with a proportional controller becomes,

$$P(s) = 0.491 \cdot K_p \cdot v \cdot \frac{(s + \frac{v}{0.559})}{s^2 + 0.491 \cdot K_p \cdot v \cdot s + 0.878 \cdot (v^2 K_p - 10.6)}. \quad (34)$$

From above, we require a minimum proportional gain of  $K_p = \frac{bg}{v^2} = \frac{10.6}{v^2}$  for stability.

#### 5.2.2 P-D Controller

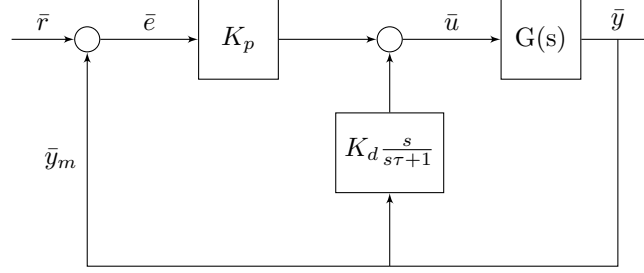


Figure 4: P-D Controller Structure

Derivative on measurement to avoid *derivative kick* when setpoint changes.

#### Derivative on Measurement

$$\begin{aligned} e &= r - y \\ u &= K_p * e - K_d * s / (sT + 1) * y \\ y &= G * u \end{aligned}$$

#### Band-Limited Differentiator

Since a pure differentiator acts as an amplifier for high frequencies, it will also amplify any high-frequency noise present in the measured plant output. Additionally, for a low sample time, this amplification increases by  $\frac{1}{T}$ . This will result in extreme *jittering* of the actuator, drastically reducing its lifespan and the overall control performance. Furthermore, we cannot calculate the exact derivative at time  $t$ . We need to approximate it numerically, for example, using Euler's backward difference method.

To address these problems, instead of implementing a pure derivative controller with transfer function  $s$ , we utilize a band-limited differentiator. This is effectively a pure differentiator cascaded with a single-pole low-pass filter,

$$D(s) = K_d \frac{s}{s\tau + 1}, \quad \tau > 0. \quad (35)$$

Where  $K_d$  is the associated derivative gain and  $\tau$  is the filter time constant. The cut-off frequency (in  $rad s^{-1}$ ) of the low-pass filter is therefore  $\frac{1}{\tau}$ . The closer  $\tau$  is to zero, the closer our approximation is to the actual derivative. However, we will also greatly increase our noise since this is inversely proportional to the cut-off frequency of the filter.

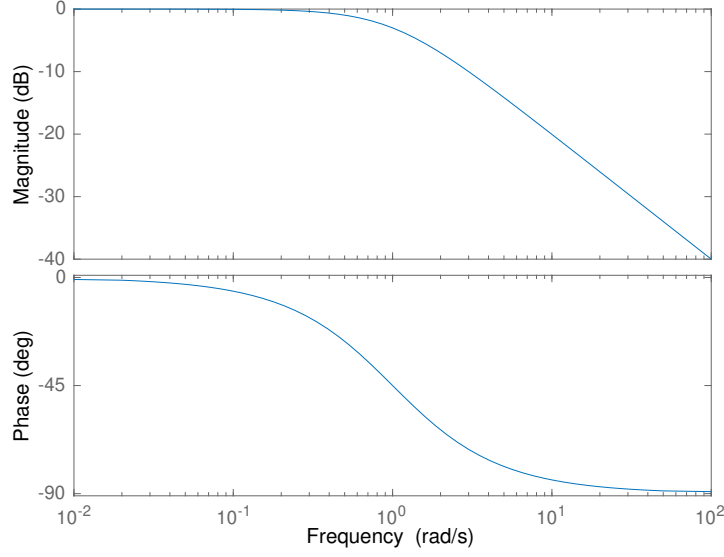


Figure 5: Bode Diagram of Band-Limited Differentiator ( $\tau = 1$ )

### Transfer Function

The closed-loop transfer function of the P-D controller in combination with a general plant  $G(s)$  is thus,

$$H(s) = \frac{K_p \cdot G(s)}{1 + G(s) \cdot (K_p + K_d \frac{s}{s\tau + 1})}.$$

Replacing  $G(s)$  with the transfer function of the linearised second order bicycle model, we arrive at the following closed-loop transfer function:

$$H(s) = K_p \cdot \gamma v a \cdot \frac{(s + \frac{v}{a})}{s^2 - b g \gamma + \gamma v a \cdot (s + \frac{v}{a}) \cdot (K_p + K_d \frac{s}{s\tau + 1})} \quad (36)$$

Where the transfer function can be further expanded by multiplying the numerator and denominator by  $(s\tau + 1)$  to yield,

$$H(s) = K_p \cdot \gamma v a \cdot \frac{(s + \frac{v}{a})(s + \frac{1}{\tau})}{s^3 + s^2 \frac{1}{\tau} (1 + \gamma v a K_p \tau + K_d) + s \frac{\gamma}{\tau} (K_p v (a + v \tau) + K_d v^2 - b g \tau) + \frac{\gamma}{\tau} (v^2 K_p - b g)}.$$

For analysis and control design purposes, this equation is too complex. We therefore simplify the transfer function by assuming we filter the sensor measurements with the low-pass filter ( $LP(s) = \frac{1}{s\tau + 1}$ ) before feeding them into the derivative-part of the controller. Therefore, we can assume a pure differentiator. This results in the simplified closed-loop, second order transfer function:

$$\begin{aligned} H^*(s) &= K_p \cdot \gamma v a \cdot \frac{(s + \frac{v}{a})}{s^2 (1 + K_d \gamma v a) + s \gamma v (a K_p + v K_d) + \gamma (v^2 K_p - b g)} \\ &= \frac{K_p \gamma v a}{1 + K_d \gamma v a} \cdot \frac{(s + \frac{v}{a})}{s^2 + s \frac{\gamma v (a K_p + v K_d)}{1 + K_d \gamma v a} + \frac{\gamma (v^2 K_p - b g)}{1 + K_d \gamma v a}} \end{aligned}$$

### Pole Placement

We need to place two poles for the closed-loop system ( $p_1$  and  $p_2$ ). The second-order characteristic equation is therefore,

$$C(s) = (s - p_1)(s - p_2) = s^2 - (p_1 + p_2)s + p_1p_2$$

To get the the controller gains  $K_p$  and  $K_d$  we equate the coefficients of the transform variable  $s$  of the closed-loop transfer function's denominator with those of the characteristic equation:

$$\frac{\gamma v(aK_p + vK_d)}{1 + K_d\gamma va} = -(p_1 + p_2) \quad (37)$$

$$\frac{\gamma(v^2K_p - bg)}{1 + K_d\gamma va} = p_1p_2 \quad (38)$$

After rearranging, we arrive at the following matrix set of linear relationships for  $K_p$  and  $K_d$ :

$$\begin{pmatrix} \gamma v^2 & -\gamma va p_1 p_2 \\ \gamma va & \gamma v(v + (p_1 + p_2)a) \end{pmatrix} \begin{pmatrix} K_p \\ K_d \end{pmatrix} = \begin{pmatrix} p_1 p_2 + \gamma bg \\ -(p_1 + p_2) \end{pmatrix} \quad (39)$$

These can then be easily solved to give  $K_p$  and  $K_d$ , so that:

$$\begin{pmatrix} K_p \\ K_d \end{pmatrix} = \frac{1}{\gamma^2 v^2 (v(v + (p_1 + p_2)a) + a^2 p_1 p_2)} \begin{pmatrix} \gamma v(v + (p_1 + p_2)a) & \gamma va p_1 p_2 \\ -\gamma va & \gamma v^2 \end{pmatrix} \begin{pmatrix} p_1 p_2 + \gamma bg \\ -(p_1 + p_2) \end{pmatrix}$$

### 5.3 State Feedback

Pole placement - do not want to drastically change system dynamics - essentially only want to provide stability (all real parts of eigenvalues must be negative). Can push poles further left in the s-plane to increase stability margins, especially since plant is modelled only approximately so will contain variations. Must watch that we do not hit the actuator limits by choosing poles magnitudes too large.

Gain-scheduled state feedback depending on forward velocity. Lookup tables, since computing poles is too resource intensive for microcontroller.

### 5.4 Optimal Control

### 5.5 Zero-Speed Stabilisation

## 6 Trajectory Tracking

## 7 Simulation

### 7.1 MatLAB

### 7.2 Unity

## 8 Implementation

Bike, Sensors, Actuators, System as a whole, Arduino, processing (LP Filter), complementary filter, kalman filter, IMU, post-processing (C#/Python?), MatLAB, etc.

**9 Results and Discussion**

**10 Conclusions**

**11 References**

**12 Appendix**