

SLIDING MODE CONTROL TRAJECTORY TRACKING
IMPLEMENTATION ON
UNDERACTUATED DYNAMIC SYSTEMS

by

MATTHEW MIGCHELBRINK

B.S. Mechanical Engineering, Kansas State University 2011

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2014

Approved by:

Major Professor
Dr. Warren N. White

Abstract

The subject of linear control is a mature subject that has many proven powerful techniques. Recent research generally falls into the area of non-linear control. A subsection of non-linear control that has garnered a lot of research recently has been in underactuated dynamic systems. Many applications of the subject exist in robotics, aerospace, marine, constrained systems, walking systems, and non-holonomic systems.

This thesis proposes a sliding mode control law for the tracking control of an underactuated dynamic system. A candidate Lyapunov function is used to build the desired tracking control. The proposed control method does not require the integration of feedback as does its predecessor. The proposed control can work on a variety of underactuated systems. Its predecessor only worked on those dynamic systems that are simply underactuated (torques acting on some joints, no torques acting on others).

For dynamic systems that contain a roll without slip constraint, often a desired trajectory to follow is related to dynamic coordinates through a non-holonomic constraint. A navigational control is shown to work in conjunction with the sliding mode control to allow tracking of these desired trajectories.

The methodology is applied through simulations to a holonomic case of the Segbot, an inverted cart-pole, a non-holonomic case of Segbot, and a rolling wheel. The methodology is implemented on an actual Segbot and shown to provide more favorable tracking results than linear feedback gains.

Table of Contents

List of Figures	vii
List of Tables	x
Acknowledgements.....	xi
Chapter 1: Introduction	1
Underactuated Control.....	2
Current Sliding Mode Control Proposal	5
Outline	8
Chapter 2: Theoretical Development	10
Lagrange's Equation	10
Direct Lyapunov Stability.....	11
Sliding Mode Control	12
Former Underactuated Control Law	14
Patenaude First Matching Equation	16
Patenaude Second Matching Condition	17
Patenaude Third Matching Condition.....	17
Patenaude Ensuring Underactuation.....	18
Proposed Underactuated Sliding Mode Control Method.....	18
Sliding Mode Derivation Recollection	19
Adaptive Underactuated Sliding Mode Control	21
Chapter Summary	23
Chapter 3: Holonomic Examples	24
Inverted-Cart Pole.....	24

Actuated Coordinate Tracking	25
Unactuated Coordinate Tracking	28
Cart-Pole Adaptive Control Implementation	30
Holonomic Segbot	32
Equations of Motion Using Newton Euler.....	33
Equations of Motion Using Lagrange.....	35
Tracking Control.....	37
Utilizing the Proposed Control for Stabilization	40
Chapter Summary	41
Chapter 4: Non-Holonomic Systems	42
Navigational Control.....	42
Navigational Control Derivation.....	45
Navigational Control Recap.....	49
Rolling Wheel.....	49
Previous Work	50
Rolling Wheel Dynamics	51
Rolling Wheel Controllability.....	52
Sliding Mode Control Parameters.....	54
Navigational Control Parameters	57
Blending SMC and Navigational Control Values.....	58
Non-Holonomic Segbot	63
Equations of Motion Using Lagrange.....	64
Control Derivation and Implementation	65
Chapter Summary	72

Chapter 5: Segbot Implementation	74
Dynamic Equation Derivation	74
DC Motor and Motor Driver Modeling	79
System Identification: R_a and K_b	80
System Identification: Inertia.....	81
Remaining Parameters	82
State Determination	83
Linear Control Gains	84
Sliding Mode Control Initialization	86
Non-Holonomic Coordinate Tracking	87
Sliding Mode Control Terms for Figure-8 Trajectory Tracking.....	87
Effects of the Navigational Control Constants	88
Effects of k_1 in the Navigational Control.....	90
Effects of k_2 in the Navigational Control.....	93
Effects of k_4 and k_5 in the Navigational Control.....	94
Comparison to Linear Feedback Control	96
Desired θ Coordinate Tracking	100
Desired Angle Tracking: Control Law Calculation	101
Desired Angle Tracking: Implementation Comparison	102
Implementation Conclusions	104
Chapter 6: Conclusion and Recommendations	106
Conclusion	106
Recommendations.....	108
References.....	109

Appendix A - Rolling Wheel	112
Appendix B - Non-Holonomic Segbot	135
Appendix C - Implementation	162

List of Figures

Figure 1: Cart-Pole System.....	25
Figure 2: Cart-Pole Actuated Coordinate Tracking Response.....	27
Figure 3: Cart-Pole Actuated Coordinate Force Input.....	28
Figure 4: Cart-Pole Underactuated Coordinate Tracking.....	30
Figure 5: Adaptive Control Pendulum Angle Response.....	31
Figure 6: Adaptive Control X-Coordinate Tracking.....	31
Figure 7: Adaptive Control Estimated Value of Mm	32
Figure 8: Segbot.....	33
Figure 9: Simplified Segbot FBD.....	34
Figure 10: State Space Feedback Gains Block Diagram.....	39
Figure 11: Holonomic Segbot Tracking Response Under Adverse Initial Conditions.....	39
Figure 12: Holonomic Segbot Stabilization Under Adverse Initial Conditions.....	40
Figure 13: Navigational Control.....	43
Figure 14: Open Loop XY Tracking.....	44
Figure 15: Navigational Control Posture Error.....	46
Figure 16: Rolling Wheel Free Body Diagram.....	50
Figure 17: Rolling Wheel Path Without Navigational Control.....	51
Figure 18: Open-Loop XY Tracking.....	57
Figure 19: Navigational Control Gains Test.....	58
Figure 20: Dynamic Response for $k1=1$ $k2=1$	59
Figure 21: Dynamic Response for $k1=1$ $k2=0.1$	59
Figure 22: Rolling Wheel Torque About X-Axis.....	60
Figure 23: Rolling Wheel Torques.....	61

Figure 24: Rolling Wheel Passive Navigational Control Contact Position	62
Figure 25: Rolling Wheel Passive Navigational Gains Response	62
Figure 26: Rolling Wheel Passive Navigational Control Lyapunov Function Plots	63
Figure 27: Segbot Open Loop XY Tracking.....	69
Figure 28: Non-Holonomic Segbot Simulation Response I	70
Figure 29: Non-Holonomic Segbot Simulation Response II	71
Figure 30: Non-Holonomic Segbot.....	74
Figure 31: Segbot XY Coordinate Tracking.....	89
Figure 32: $k_1=0.1$ Position Absolute Value Error.....	90
Figure 33: Steady State Position Variations for k_1	91
Figure 34: Steady State Time vs Absolute Value Distance Error for k_1	92
Figure 35: Steady State Position Variations for k_2	93
Figure 36: Steady State Time vs Absolute Value Distance Error for k_2	94
Figure 37: Steady State Position Variations for k_4 and k_5	95
Figure 38: Steady State Time vs Distance Error for k_4 and k_5	95
Figure 39: Legend for Figure 40-Figure 46	96
Figure 40: Steady State $k_1=0.1, k_2=1.0, k_4=0, k_5=0$	96
Figure 41: Steady State $k_1=0.30, k_2=5.0, k_4=0, k_5=0$	97
Figure 42: Steady State $k_1=1.0, k_2=5.0, k_4=0, k_5=0$	97
Figure 43: Steady State $k_1=0.10, k_2=5.0, k_4=0, k_5=0$	98
Figure 44: Steady State $k_1=0.5, k_2=30.0, k_4=200, k_5=0$	98
Figure 45: Steady State $k_1=1.1, k_2=50, k_4=500, k_5=100$	98
Figure 46: Steady State $k_1=0.5, k_2=30, k_4=0, k_5=0$	99
Figure 47: θ_d Path Tracking Sliding Mode Control Gains	102

Figure 48: Legend for Figure 49-Figure 51	102
Figure 49: Desired Angle Response for Path 1	103
Figure 50: Desired Angle Response for Path 2.....	103
Figure 51: Desired Angle Response for Path 3.....	104

List of Tables

Table 1: Cart-Pole Physical Parameters.....	25
Table 2: Cart-Pole Actuated Coordinate Tracking Control Law Values.....	27
Table 3: Cart-Pole Underactuated Coordinate Control Law Values.....	29
Table 4: Simplified Segbot System Parameters.....	38
Table 5: Rolling Wheel System Parameters.....	55
Table 6: Initial Sliding Mode Control Parameters.....	56
Table 7: Non-Holonomic Segbot Properties.....	68
Table 8: Non-Holonomic Segbot Sliding Mode Control Values.....	68
Table 9: Ra and Kb Calculations.....	81
Table 10: Inertia and Viscous Friction Constants.....	82
Table 11: Segbot Implementation Parameters.....	83
Table 12: Implementation Average Position Error.....	100
Table 13: θ_d Trajectories and Linear Control Gains.....	101
Table 14: Response Characteristics for θ_d Tracking.....	104

Acknowledgements

I like to express my appreciation to Professor Warren White for his patience, time, and guidance throughout the research process. Your time and guidance have proven invaluable throughout the construction of this thesis. Also, I would like to thank Professor Dave Auckly and Professor Dale Schinstock for their support.

I would also like to thank my family and particularly my significant other Kristina for their encouragement and support throughout my academic endeavors. I feel truly blessed.

Chapter 1: Introduction

Linear control is a mature subject that has an array of proven powerful techniques. These have been put to practical use in a variety of industries. Unfortunately, in actual implementation there are very few (if any) totally linear systems. In recent years, major advances have been made in the theory and application of non-linear control. Some of the more common areas of research have been in feedback linearization, back stepping, sliding mode control, robust control, and adaptive control.

New techniques in non-linear control can provide improvements in performance on existing linear control. Some dynamic systems cannot be accurately approximated as linear. Examples of non-linear occurrences include various types of hysteresis, saturation, static friction, non-holonomic coordinates, and mechanical system non-linearities. These dynamic systems might not be able to be controlled by linear methods, while being able to be controlled by non-linear methods.

Stability of nonlinear systems is often analyzed from a Lyapunov point of view (Slotine & Li, 1991). There are varying degrees of stability: Lyapunov stability, local asymptotic stability, and global asymptotic stability. Though Lyapunov is often used as a tool to prove the stability of a given control scheme, the creation of a Lyapunov function for a dynamic system can be quite challenging. This is especially true for underactuated dynamic systems; these are systems that contain more degrees of freedom than actuators.

A textbook underactuated controls problem is the inverted cart-pole (see Chapter 3). Its popularity derives partly because the system is unstable without control. The open loop response of the dynamic system has a right hand plane pole, which makes the system unstable. With feedback the system is able to be controlled and then exhibits non-minimum phase behavior.

A dynamic system that has gained a lot of interest in recent years is the Segbot (see Chapters 3, 4, and 5). If the generalized coordinates did not include the axel's center position the Segbot would still be underactuated and the linearized version would have a right hand plane pole, similar to the inverted cart-pole. However, if the axel center position is not constrained to move in a straight line, the system is also non-holonomic. This added layer of complexity makes

the Segbot an excellent candidate for underactuated tracking research. The Segbot will be used extensively in this thesis.

Underactuated Control

Non-holonomic systems are by definition underactuated systems. Much research has occurred on dynamic systems that contain a roll without slip constraint (which is non-holonomic) that introduces the generalized coordinates X and Y . These coordinates have no applied torques or actuators acting on them.

Non-holonomic trajectory tracking is an area of research by itself. Cheng et al. (2006) and Quan et al. (2008) made use of backstepping to produce non-holonomic trajectory tracking control laws. They made use of kinematic constraints for the system inputs and outputs. In implementation, it would be more beneficial to implement torque inputs. Yang, F., & Wang, C. (2012) consider the tracking problem of a wheeled robot with a camera of unknown parameters. The control presented by Yang, F., & Wang, C. (2012) is able to reject disturbances and provide position/orientation tracking by compensating for parametric uncertainties through adaptive control techniques. Bounded uncertainties can be suppressed by sliding mode characteristics. The key contribution from the paper is that the problem is discussed in the image frame (from the camera) and the inertial frame, which provides some subtle simplifications to the tracking problem. The dynamic system is only underactuated due to the non-holonomic constraints.

One of the more well-known and documented non-linear control techniques is feedback linearization. It is presented in most non-linear control books such as *Applied Non-Linear Control* by Slotine, J. J., & Li, W. (1991). This control maps a nonlinear system into a linear system by feedback that cancels the nonlinear terms. There are two major shortcomings for feedback linearization. First off, the control relies heavily on the system model accuracy. In application, all models have some assumptions that will lead to inaccuracies in the dynamic model. The second major disadvantage is that for full-state feedback linearization, the dynamic system must be fully actuated.

Rigid body link systems that are simply underactuated (torques on some joints, no torques on others) can have the actuated degrees of freedom linearized. Collocated feedback linearization is the process of implementing feedback linearization on the actuated degrees of

freedom. In cases of strong inertial coupling, the unactuated degrees of freedom can be linearized through feedback linearization [Spong (2004)]. This case of control is called non-collocated feedback linearization. Results from Spong (2004) show that these control techniques work well. However, the dynamic systems were all rigid body links with torques applied at the joints. These systems are fairly simple underactuated systems and the control law could become quite cumbersome to apply to more complicated systems.

Backstepping has proven to be an effective method to implement tracking control of underactuated systems. This method was effectively implemented by Olfati-Saber (2001,2002). In his work, nominal forms to present underactuated dynamic systems were also presented. Hongrui, Yantao, Siyan, and Zhen (2008) also used a backstepping scheme to stabilize a plate while tracking a reference trajectory of a ball on the plate. During implementation backstepping control values had to be manually tuned. An automated tuning system would greatly improve the implementation. One shortcoming of backstepping methods is that the control derivation can vary greatly from case to case. Backstepping methods generally require some creativity on the control designers part to account for undesirable terms in the equations of motion. It is hard to provide a systematic method for all types of underactuated systems using these methods.

Another popular technique uses inverse dynamics. In this scheme the dynamic equations are used to calculate the required torque to make a dynamic system follow a given trajectory. The technique often is applied to links as is the case in Isobe, Yamanaka, & Kitamura (2008). They present a parallel solution scheme that uses link trajectories calculated by a successive revision algorithm presented in their paper. A major advantage of the technique is that the use of the parallel solution scheme is not limited by the number of joints in the dynamic system. The technique can also incorporate the elastic deformation of links. Simulation results show the stabilization of an eight-joint serial link dynamic system. The technique requires that the dynamic system contain links. The matrix elements describing each link are successive and depend on the previous link. The equation elements for later links begin to grow quite lengthy.

In general, controllers that involve backstepping or inverse dynamics are cumbersome to obtain. Modeling inaccuracies greatly affect the control. There are no robustness terms built into the control. The control derivation for these techniques is very case by case dependent. A systematic approach that works for all systems is difficult to provide.

One non-linear control method that has proven to be effective and robust is sliding mode control. Sliding mode control has the ability to account for parametric uncertainty in a model. If the system is fully actuated, a sliding mode control law can be derived with little trouble. However, underactuated dynamics presents new problems.

Work by Schkoda (2007) and DiFiore (2009) used a squaring transformation matrix to apply a sliding mode controller to underactuated systems. The key concept was to multiply the input matrix of a linearized state-space system, \mathbf{B} , by a transformation matrix, \mathbf{T} . \mathbf{B} is size $n \times m$, \mathbf{T} is size $m \times n$, and \mathbf{TB} is size $n \times n$. \mathbf{T} is to be chosen so the resulting matrix, \mathbf{TB} , has full rank and is invertible. Choosing the elements of \mathbf{T} would allow the control to select which states were controlled most precisely. The control is fairly simple to derive, and tracking priorities could easily be adjusted between states. An advantage of the work is that it is easily applicable to different systems. Specific feedback related to the dynamics is not required. One short coming of the work was that the approach was only applied to dynamic systems that have 4 states (or two 2nd order differential equations describing the motion of only two generalized coordinates). The tracking actually responds to coordinates that correspond to \mathbf{TB} , while the desired coordinates to track actually correspond to \mathbf{B} . The dynamic equations of motion also had to be completely linear with respect to the generalized coordinates.

Ashrafiuon & Erwin (2004) derive a stabilizing sliding mode controller for a system with n degrees of freedom where m are actuated. The control builds only m sliding surfaces. However, each sliding surface consists of a combination of actuated and unactuated generalized coordinates. This combination of actuated and unactuated coordinates presents a systematic method to derive a particular control for a dynamic system. The proposed controller does guarantee that all system trajectories converge to the sliding surface if certain conditions are met. However, there is no guarantee that once on the surface the trajectory will head to the desired stabilizing point. The control was successfully applied to effectively position the attitude control of a satellite system when the vehicle loses control of a reaction wheel.

Xu et al. (2014) derive an integral sliding mode controller and implement the control on a spin-off of the Segbot (see Chapters 3, 4, 5). In their paper, the Segbot wheels are locked together so that the tracking is holonomic. However, they allow the Segbot to move on an inclined plane. Xu et al. (2014) show that their derived control can handle disturbances and

reach a set point (possibly up or downhill) with both matched and unmatched uncertainties in the dynamic model. Matched uncertainties were able to be perfectly rejected with the inclusion of a switching term. The main contribution of the paper is that for the first time integral sliding mode control is applied to a two wheel mobile robot. In this thesis the dynamics are not constrained so that the Segbot is holonomic; however, the Segbot is constrained to movement on a flat plane.

Current Sliding Mode Control Proposal

The goal of this work is to present a systematic framework to guide the development of a control law for an underactuated dynamic system. An underactuated system has more degrees of freedom than actuators. The presented control law will track a given trajectory of generalized coordinates. The starting point for the control design comes from White et al. (2006, 2007, 2008), Slotine and Li (1991, 1998), and Patenaude (2008).

White et al. (2006, 2007, 2008) used the Direct Lyapunov Method to construct three matching conditions to ensure that a dynamic system could be stabilized. The first candidate Lyapunov function proposed was $V(q, \dot{q}) = \frac{1}{2} \dot{q}^T K_d \dot{q} + \phi(q)$. K_d could be rewritten as PM , where M represented the mass matrix of the dynamic system, and P was an unknown positive definite matrix. $\phi(q)$ was some unknown potential function. For further details, see Chapter 2.

A problem with early implementations of the Direct Lyapunov Approach was that there must be continuous feedback integrated to determine the elements of K_d . It was noticed that the matrix K_d could be approximated as almost equal to a constant value without significantly hindering the dynamic response to the control law by White et al (2008).

During the development of the control law, it was found that local asymptotic stability could be ensured through the use of three matching conditions. Subsequent papers used variations of the three matching conditions. However, similar terms would show up in corresponding matching conditions. The first matching condition contained properties that were pre and post multiplied by the velocity vector and the matching condition was cubic in the generalized velocities. The second matching condition contained viscous damping and was quadratic in the generalized velocities. The third matching condition involved partial differential

equations and it includes a conservative vector field from the dynamic system. The third matching condition was linear in the generalized velocities. It should be noted that these matching conditions were found with the goal of stabilization in mind, not trajectory tracking.

Subsequent work by Patenaude (2008) was focused on the trajectory tracking of underactuated systems. The method uses the matching conditions derived by White et al. to help ensure that the control is underactuated. The method then attempts to fuse the underactuated matching conditions with a fully actuated sliding mode control law presented by Slotine and Li (1988). The end result is a control law that is both underactuated and provides trajectory tracking. The method is explained in detail in Chapter 2.

There were some shortcomings of the work by Patenaude (2008). Some undesirable terms had to be included to fuse the underactuated stabilizing control laws presented by White et al. (2006, 2007, 2008) with the fully actuated tracking control law presented by Slotine and Li (1988). Numeric integration was still required to estimate the matrix K_d . Furthermore the type of underactuation was restricted. For a dynamic system with n degrees of freedom and m actuators, the method required that the first m equations have actuators, while the remaining $n-m$ equations have no actuation.

The most complicated dynamic system that Patenaude (2008) applied his control method to was the rolling wheel, as presented in Chapter 2. Think of the rolling wheel as a penny. The rolling wheel has applied torques on the roll and pitch directions, while the yaw coordinate is unactuated. This forces the wheel to be steered by leaning. The linearized version of this dynamic model is uncontrollable; thus, linear control techniques are not directly applicable to this dynamic system. The control proposed by Patenaude could track a figure-8 trajectory with limited success.

Some questions were still left unanswered. The rolling wheel could track desired wheel angles, but in real world application it would make more sense to track the contact position of the wheel on the ground, namely a desired X and Y coordinate. This deals with the non-holonomic nature of the dynamic system. Also, a method to choose each control law component was not thoroughly developed.

None of the methods discussed so far specifically address non-holonomic coordinate tracking of a dynamic system that is underactuated if the non-holonomic constraints were not

imposed. The Segbot has only three degrees of freedom (if the non-holonomic constraints are not imposed) and only two actuators. Applying the non-holonomic constraints to the system adds another layer of complexity to the problem. One of the main contributions from this thesis is the actual implementation of non-holonomic coordinate tracking on the Segbot.

The goal of this thesis is to derive an underactuated sliding mode control law that can be used for trajectory tracking. Areas of emphasis are:

- Clearly derive the sliding mode control law from a candidate Lyapunov function. This control could also be applied to holonomic systems.
- Keep the control as clean as possible. Don't add unnecessary terms to the control law. Make sure that all terms included in the control law serve a purpose or improve the implementation of the control.
- The control will use matching conditions to ensure that the torques applied to the dynamic system are underactuated. However, the matching conditions will be found with the objective of trajectory tracking in mind, not stabilization.
- Construct the control law so that K_d from White et al. (2006, 2007, 2008) does not need to be integrated to satisfy the first matching condition. This is performed by setting the matrix presented by White et al. (2006, 2007, 2008) equal to the identity matrix.
- The control should not be limited to underactuated dynamic systems that have an applied torque on the first m generalized coordinate equations and no applied torque on the remaining $n-m$ equations.
- Show robustness of the control.
- Allow an emphasis to be placed on different tracking characteristics.
- For the non-holonomic roll without slip condition where torque is applied to a wheel, provide a solution to tracking a desired X and Y coordinate, while measuring wheel coordinate's and heading direction.
- Show that other developed control techniques can be integrated in conjunction with the control. Specifically show that the control can be made adaptive.

- Implement non-holonomic, variable speed, X and Y coordinate trajectory tracking on the Segbot.

Outline

Chapter 2 of this thesis begins with the introduction of Lagrange's Equation. It is presented in non-holonomic form, which may also be applicable to holonomic systems if the Lagrange multipliers and constraints are removed. The fundamentals of Direct Lyapunov Stability are presented. Chapter 2 then derives a fully actuated sliding mode control. The control used by Patenaude (2008) is then presented in detail. Some problems are discussed and a new method for underactuated sliding mode control is presented. The versatility of the method is shown by adding an adaptive term to it.

Chapter 3 contains two holonomic examples. The first is the inverted cart-pole system. It is shown through simulation that a tracking emphasis can be placed on different coordinates. The next example is the holonomic case of the Segbot. The control methods used by White et al. (2006, 2007, 2008) and Patenaude (2008) were not applicable to this type of underactuated system. It is shown that the proposed non-linear control method can help to increase the sphere of attraction. The chapter ends with an implementation of adaptive control.

Chapter 4 begins with the derivation of a navigational control. This is used to address the non-holonomic problem of tracking the X and Y contact position of a rolling wheel. Advantages that the navigational control provides are also discussed. Chapter 4 contains two non-holonomic dynamic systems. The first system discussed is the rolling wheel as presented in Patenaude (2008). The navigational control is combined with the proposed underactuated sliding mode control from Chapter 2 so that the contact point of the rolling wheel can track a figure-8 path. The next non-holonomic system is a less constrained version of the Segbot. This dynamic system would not be controllable by the previous underactuated sliding mode control presented by Patenaude (2008). In combination with the navigational control, the non-holonomic Segbot can track a figure-8 path as well.

Chapter 5 discusses the application of the sliding mode control law in conjunction with the navigational control on the Segbot. Chapter 5 begins by presenting the derivation of the dynamic equations of motion used for the Segbot. Next, the system identification procedure is

presented for dynamic properties that could not be directly measured. The derivation for the non-linear control is presented and figure-8 tracking results for the Segbot are presented.

Chapter 6 recaps the thesis and provides final conclusions and recommendations for future work.

Chapter 2: Theoretical Development

First an introduction to Lagrange's equations is presented. This is followed by presenting Lyapunov's Direct Method. Basic concepts from Lyapunov's Direct Method are essential to the fundamentals of sliding mode control. This is followed by an introduction to sliding mode control. The underactuated tracking method presented by Patenaude (2008) will be discussed. The proposed sliding mode control for underactuated systems will be provided. The versatility of the control is shown by adding an adaptive term to the control.

Lagrange's Equation

The method of Lagrange can be used to derive dynamic equations. Lagrange's equation for holonomic systems in terms of independent generalized coordinates is given in Greenwood (2003) as

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \quad (1)$$

where L represents the Lagrangian of the system, which is simply the difference between the kinetic energy of the system and the potential energy. Q represents the non-conservative forces acting on the system, t is time, q is a vector of the generalized coordinates, and the subscript i represents the corresponding generalized coordinate. For the dynamic models presented in this thesis, Q is written as a vector of the applied torques and is written as τ . Equation (1) can be further reduced to the matrix dynamic equation form of

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \quad (2)$$

where $M(q)$ consists of mass and inertia quantities, $C(q, \dot{q})\dot{q}$ consists of centripetal and Coriolis moments and forces, and $G(q)$ consists of excitations stemming from the gradient of a potential field, quite often from a gravitational field. τ represents the input actuations to the dynamic system. In subsequent sections the arguments associated with the matrices will be dropped. However, they are still functions of the coordinates given. The non-holonomic form of Equation (1) is given by Greenwood (2003) as

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i + \sum_{j=1}^l \lambda_j a_{ji} \quad (3)$$

where λ_j represents the corresponding Lagrange multiplier, l is the number of constraints, a_{ji} represents the constraint component of the i^{th} generalized coordinate, and the j^{th} constraint. If the Lagrange multipliers are set to zero, this is the same form as the holonomic version.

If the τ vector consists of fewer actuators than degrees of freedom, the system is said to be underactuated. In previous work by Patenaude (2008) and White et al (2006, 2007, 2008) the dynamic system was defined so that there were n generalized coordinates and m actuators. The vector τ would contain actuators on the first m equations. The remaining $n-m$ coordinates have no applied torques on the joints. This specific case of underactuation is referred to as simply underactuated. In this work, the τ vector is underactuated; however, it does not have the constraint of simply underactuated.

Direct Lyapunov Stability

The fundamental philosophy of Lyapunov's direct method is that if the total energy (or some other positive metric) in a system is continuously decreased, then the system will come to rest at an equilibrium point. A thorough overview of Lyapunov's direct method is presented in Slotine and Li (1991). An outline of some essential characteristics is presented here.

First a candidate Lyapunov function is defined as the map

$$V(q) : \mathbb{R}^{2n} \rightarrow \mathbb{R}. \quad (4)$$

In general, the Lyapunov function is a function of all the generalized coordinates. In order to be a Lyapunov function, the function must satisfy the conditions of

$$V(q) \geq 0 \quad (5)$$

and

$$\dot{V}(q) = \frac{d}{dt}V(q) \leq 0. \quad (6)$$

Recall that this Lyapunov function is associated with the energy, or some positive definite function of the generalized positions and velocities of the system. If these conditions are met then the function is stable in the sense of Lyapunov. If the function meets these conditions, then the system will dissipate energy, which decreases the positive definite quantity. Eventually the system will come to rest at an equilibrium point.

Sliding Mode Control

Sliding mode control relies on the energy dissipation of the proposed Lyapunov function. It also provides a way to formulate a tracking controller. First some notations need to be introduced. The tilde coordinate is the difference between the generalized coordinate and the desired (d subscripted) generalized coordinate. Examples of \tilde{q} and its time derivatives are

$$\begin{aligned}\tilde{q} &= q - q_d, \\ \dot{\tilde{q}} &= \dot{q} - \dot{q}_d, \\ \text{and} \\ \ddot{\tilde{q}} &= \ddot{q} - \ddot{q}_d.\end{aligned}\tag{7}$$

The reference velocity and its time derivative are defined as

$$\begin{aligned}\dot{q}_r &= \dot{q}_d - \Lambda (q - q_d) \\ \text{and} \\ \ddot{q}_r &= \ddot{q}_d - \Lambda (\dot{q} - \dot{q}_d)\end{aligned}\tag{8}$$

where $\Lambda \in \mathfrak{R}^{n \times n}$ is a positive definite matrix, that is usually diagonal. The sliding mode vector is given as

$$s = (\dot{q} - \dot{q}_d) + \Lambda(q - q_d) = \dot{\tilde{q}} + \Lambda\tilde{q} = \dot{q} - \dot{q}_r.\tag{9}$$

Observe that no subscript on the generalized coordinate denotes the actual position, the subscript d represents the desired quantity, and r signifies the reference quantity. It can be observed that the sliding mode is actually a weighted sum of the position and velocity error. By using the sliding mode vector, the problem of tracking $x = x_d$ is equivalent to remaining on the surface $s = 0$.

With these definitions now in mind, define a candidate Lyapunov function as

$$V = \frac{1}{2} s^T M s.\tag{10}$$

This function contains the mass matrix, which must be positive definite. It is also pre and post multiplied by the s vector. It fits the build of a candidate Lyapunov function because it is positive definite and is equal to zero only when the elements of s are equal to zero.

The candidate Lyapunov function is differentiated with respect to time. After the differentiation a substitution is made for \dot{s} to give

$$\dot{V} = s^T M \dot{s} + \frac{1}{2} s^T \dot{M} s = s^T M [(\ddot{q} - \ddot{q}_d) + \Lambda(\dot{q} - \dot{q}_d)] + \frac{1}{2} s^T \dot{M} s = -s^T K_d s \leq 0 \quad (11)$$

where K_d is a positive definite matrix used for the control. Recall the matrices M and C from Equation (2). Next the term $s^T C s$ is added to the first term in brackets, and subtracted from $\frac{1}{2} s^T \dot{M} s$ in Equation (11). Because the matrix $\frac{\dot{M}}{2} - C$ is skew-symmetric (when using Lagrange's method to determine the equations), the pre and post multiplication by the s vector takes this term to zero. Next \ddot{q} is solved for in Equation (2) and the solution is inserted into Equation (11) to give

$$\dot{V} = s^T M [(M^{-1}(-C(\dot{q} - s) - G + \tau) - \ddot{q}_d + \Lambda(\dot{q} - \dot{q}_d))] = -s^T K_d s \leq 0. \quad (12)$$

The terms are written in terms of the reference coordinates as shown in Equations (7)-(9). This simplification yields

$$\dot{V} = s^T [-M\ddot{q}_r - C\dot{q}_r - G + \tau] = -s^T K_d s \leq 0. \quad (13)$$

The control law for τ is determined so that the right hand side of Equation (13) is achieved where $K_d \in \mathbb{R}^{n \times n} \geq 0$. If the system properties are known exactly, (i.e. $\hat{M} = M, \hat{C} = C, \hat{G} = G$, where \hat{M} is the best known estimate of the system mass matrix etc.) then it is well known that there is a control law for a fully actuated system that can asymptotically track a desired path. The fully actuated control is discussed in the following paragraph.

τ is to be chosen to ensure that the time derivative of the Lyapunov function given in Equation (13) stays negative semi-definite. Often the matrix K_d is symmetric, though this is not a requirement. Only the symmetric part of the matrix must be positive-semi definite as the pre and post multiplication of the sliding mode vector cancels the skew-symmetric part of the matrix. Stripping off the pre-multiplication of the s^T vector and solving for τ yields the control law

$$\tau = M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + G(q) - K_d s. \quad (14)$$

If the parameters of Equation (14) are exactly known, and the control is fully actuated, then the control law ensures global asymptotic stability. Note that the first three terms of the control law ensure that \dot{V} is zero. They are referred to as the feed forward term in Slotine and Li (1988). The last term is analogous to a PD control term. In reality, sometimes parameters in the first three terms of Equation (14) are not exactly known. This is where adaptive and/or robust control

parameters can be built in as a safety measure to ensure stability or enhance performance. Note that direct implementation of Equation (14) is applicable only to a fully actuated system with known parameters.

Former Underactuated Control Law

The previous underactuated sliding mode tracking control given by Patenaude (2008) is

$$\begin{bmatrix} \tau \\ 0 \end{bmatrix} = M\ddot{q}_r + C\dot{q}_r + C_d\dot{q}_r - \bar{K}_d s + \begin{bmatrix} u1 + F \\ u2 \end{bmatrix} + P^{-1}\nabla\phi(q) \quad (15)$$

where $\bar{K}_d \in \mathfrak{R}^{n \times n}$ is a positive definite Hermitian matrix. It has the same form as the K_d matrix from the fully actuated sliding mode control presented in Equation (14). $\phi(q)$ is a real scalar potential function of the generalized coordinates.

P was used to define the candidate Lyapunov function. Its definition is explained shortly. $u2$ from the vector $u = [u1 \ u2]^T$ is used to provide stabilization to the lower $n-m$ rows. C_d contains non-conservative viscous damping terms. F is equal to $F_1 + F_2 + F_3$. These control terms are determined from a set of three matching conditions. The mass matrix, Coriolis and centripetal matrix, reference velocity, reference acceleration, and sliding mode vector are the same as those presented in Equation (14).

The goal is to show that this control yields a candidate Lyapunov function. The non-conservative viscous term is included in the equations of motion given by Equation (2). This result is set equal to Equation (15) to yield

$$\begin{aligned} \begin{bmatrix} \tau \\ 0 \end{bmatrix} &= M(q)\ddot{q} + C(q, \dot{q})\dot{q} + C_d\dot{q} + G(q) \\ &= M\ddot{q}_r + C\dot{q}_r + C_d\dot{q}_r - \bar{K}_d s + \begin{bmatrix} u1 + F \\ u2 \end{bmatrix} + P^{-1}\nabla\Phi(q). \end{aligned} \quad (16)$$

$M\ddot{q}_r + C\dot{q}_r + C_d\dot{q}_r$ is subtracted from the first and second partitioned rows of Equation (16). A substitution is made so that the sliding mode vector and its time derivative are included. After these substitutions the first row is set equal to the second row to yield

$$M\dot{s} + Cs + C_d s + G = -\bar{K}_d s + \begin{bmatrix} u1 + F \\ u2 \end{bmatrix} + P^{-1}\nabla\phi(q). \quad (17)$$

The direct Lyapunov approach of White et al. (2006, 2007, 2008) is used as a guide to design F of the tracking control law in Equation (15). The candidate Lyapunov function is

$$V = \frac{1}{2}s^T K_d s. \quad (18)$$

K_d was defined by White et al. (2006, 2007, 2008) as a positive definite matrix where $K_d = P M$. M is the mass matrix and P is an unknown positive definite matrix with full rank. The time derivative of Equation (18) is taken. Equation (17) is solved for \dot{s} . A substitution is made for \dot{s} to provide

$$\begin{aligned} \dot{V} &= s^T K_d \dot{s} + \frac{1}{2}s^T \dot{K}_d s \\ &= s^T K_d M^{-1} \left(-C s - C_d s - G - \bar{K}_d s + \begin{bmatrix} u_1 + F \\ u_2 \end{bmatrix} + P^{-1} \nabla \phi(q) \right) + \frac{1}{2}s^T \dot{K}_d s. \end{aligned} \quad (19)$$

The components of F in Equation (19) are chosen so that the right hand side of Equation (19) is given by

$$\dot{V} = -s^T K_v s + \Psi(s, u) \leq 0 \quad (20)$$

where K_v is an unknown positive semi-definite matrix and $\Psi(s, u)$ is another control parameter discussed in a later section of this chapter. The matching conditions are chosen to ensure that Equation (19) is equal to Equation (20), and thus the candidate Lyapunov function is formed by the choice of the control law given by Equation (15).

Following the procedure of White et al. (2008) F is rewritten as $F = F_1 + F_2 + F_3$, where F_i corresponds the i^{th} matching condition. The first matching condition contains dynamic terms related to s^2 , the second matching contains terms linear in s , and the third matching condition contains the generalized coordinates. Substituting $F = F_1 + F_2 + F_3$ into Equation (19) yields the three matching conditions

$$\begin{aligned} s^T K_d M^{-1} \left(-C s + \begin{bmatrix} F_1 \\ 0 \end{bmatrix} \right) + \frac{1}{2}s^T \dot{K}_d s &= 0, \\ s^T K_d M^{-1} \left(-C_d s - \bar{K}_d s + \begin{bmatrix} F_2 \\ 0 \end{bmatrix} \right) &= -\frac{1}{2}s^T K_v s, \end{aligned} \quad (21)$$

and

$$s^T K_d M^{-1} \left(-G + \begin{bmatrix} F_3 \\ 0 \end{bmatrix} + P^{-1} \nabla \phi(q) \right) = 0.$$

These matching conditions are derived for a control used for trajectory tracking in a very similar fashion to the matching conditions used for stabilization. However the terms

$$s^T P \begin{bmatrix} u1 \\ u2 \end{bmatrix} = \Psi(s, u) \leq 0 \quad (22)$$

are unaccounted for by the three matching conditions.

Patenaude First Matching Equation

Patenaude (2008) then subtracted two symmetric matrices $C'_d \in \mathfrak{R}^{n \times n}$ and $\bar{C}_d \in \mathfrak{R}^{n \times n}$ from the first matching condition and added them to the second. The sum of the first two matching conditions is still the same. The first matching equation was rewritten as

$$s^T K_d M^{-1} \left(-Cs - C'_d s - \bar{C}_d s + \begin{bmatrix} F_{1m} \\ 0 \end{bmatrix} s \right) + \frac{1}{2} s^T \dot{K}_d s = 0. \quad (23)$$

F_1 from Equation (21) was partitioned so that $F_1 = \begin{bmatrix} F_{1m} \\ 0 \end{bmatrix} s$. Because all the terms are pre and post multiplied by s , only the symmetric part of the resulting matrix equation must be zero.

Following White et al. (2008) this leads to the following condition

$$\begin{aligned} & \dot{K}_d - K_d M (C + C'_d) - (C + C'_d)^T M^{-1} K_d + \\ & K_d M^{-1} \left(-\bar{C}_d + \begin{bmatrix} F_{1m} \\ 0 \end{bmatrix} \right) + \left(-\bar{C}_d + \begin{bmatrix} F_{1m} \\ 0 \end{bmatrix} \right)^T M^{-1} K_d = 0. \end{aligned} \quad (24)$$

The F_{1m} and \bar{C}_d are chosen so that the last two terms of Equation (24) are given by

$$K_d M^{-1} \left(-\bar{C}_d + \begin{bmatrix} F_{1m} \\ 0 \end{bmatrix} \right) + \left(-\bar{C}_d + \begin{bmatrix} F_{1m} \\ 0 \end{bmatrix} \right)^T M^{-1} K_d = -\beta (K_d - K_{df}) \quad (25)$$

where β is a negative constant and K_{df} is the form of the matrix K_d at equilibrium. Combining Equations (24) and (25) yields

$$\dot{K}_d - K_d M (C + C'_d) - (C + C'_d)^T M^{-1} K_d - \beta (K_d - K_{df}) = 0. \quad (26)$$

This set of differential equations must be evaluated numerically as part of the feedback. If K_d is known through this integration, then P can be determined.

Patenaude Second Matching Condition

Adding $C'_d \in \mathfrak{R}^{n \times n}$ and $\bar{C}_d \in \mathfrak{R}^{n \times n}$ to the second matching condition yields

$$s^T K_d M^{-1} \left(-C_d - \bar{K}_d + \begin{bmatrix} F_{2m} \\ 0 \end{bmatrix} + C'_d + \bar{C}_d \right) s = -s^T K_v s. \quad (27)$$

Applying similar simplifications to those performed on the first matching condition yields

$$P \left(-C_d - \bar{K}_d + \bar{C}_d + C'_d \right) + P \begin{bmatrix} F_{2m} \\ 0 \end{bmatrix} = -K_v. \quad (28)$$

Equation (28) is solved in a two-step process. First K_v is written as $K_{v1} + K_{v2}$. The sum of these must be positive definite. The solution that affects the control is given by

$$\begin{bmatrix} F_{2m} \\ 0 \end{bmatrix} = -P^{-1} K_{v1} \quad (29)$$

for which the solution of simply underactuated systems is given by

$$K_{v1} = \sum_{i=1}^m \alpha_i P_i P_i^T \quad (30)$$

where P_i is the i^{th} column of P and α_i are positive constants. This ensures that K_{v1} is positive semi-definite.

Excluding K_{v1} the remaining terms in Equation (28) are

$P \left(-C_d - \bar{K}_d + \bar{C}_d + C'_d \right) = -K_{v2}$. C'_d is then defined as $C'_d = -\bar{C}_d$. Because P, C_d , and \bar{K}_d are all positive definite, K_{v2} must also be positive definite.

Patenaude Third Matching Condition

Stripping off s^T from the third matching condition in Equation (21) yields

$$K_d M^{-1} \left(-G + \begin{bmatrix} F_3 \\ 0 \end{bmatrix} + P^{-1} \nabla \phi(q) \right) = 0. \quad (31)$$

The solution for Equation (31) can be performed in the same manner as shown in White et al. (2008).

Patenaude Ensuring Underactuation

The remaining terms not included in the three matching conditions are given by Equation (22) as

$$s^T P \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \Psi(s, u). \quad (32)$$

In order to satisfy the Lyapunov function the right side must be non-positive. P is partitioned as $P = [P_1 \ P_2]$ and $\Psi(s, u)$ is set to $-s^T s$. Then the Moore-Penrose pseudo-inverse is used to obtain

$$u_1 = -(P_1^T P_1)^{-1} P_1^T s - (P_1^T P_1)^{-1} P_1^T P_2 u_2. \quad (33)$$

Using the pseudoinverse is not an ideal solution process.

There are a vast array of underactuated systems that this control methodology does not apply to. The dynamic system must be simply underactuated. Also the last $n-m$ coordinates must be the unactuated coordinates. Specifically in this thesis the Segbot dynamics would not be applicable. Due to \dot{K}_d appearing in the first matching condition, integration of \dot{K}_d must be performed to apply the control law.

The substitution $G = F_3 + P^{-1} \nabla \phi$ is obtained from the third matching condition in Equation (31) and applied to the control law presented in Equation (15). This effectively removes the potential, ϕ , and F_3 from the control law because it is much easier to include the dynamic gravitational terms. This is not too surprising as this matching condition was derived by White et al. (2008) while obtaining control for stabilization, not trajectory tracking.

u_2 was introduced for feedback linearization so that the lower $n-m$ equations remain stable. However, though introduced for purposes of feedback linearization, u_2 also plays a prominent role in determining the desired accelerations that ensure the control is underactuated. Because of this relationship to the desired accelerations, choosing u_2 to stabilize the lower $n-m$ equations is no longer trivial. It complicates the control law, while not performing the function that it was introduced for in the first place.

Proposed Underactuated Sliding Mode Control Method

The goal of this work is to use the fully actuated control given in Equation (14) as a starting point for developing a systematic framework to derive an underactuated control law for a

dynamic system having n degrees of freedom and m actuators. Though asymptotic tracking for an underactuated system may not be possible given the dynamics and desired trajectories, a trajectory tracking emphasis may be placed on some of the coordinates.

Sliding Mode Derivation Recollection

The candidate Lyapunov function is the same for the proposed underactuated case as the fully actuated case presented in Equation (10), $V = \frac{1}{2} s^T M s$. The time derivative is taken to obtain

$$\dot{V} = s^T M \dot{s} + \frac{1}{2} s^T \dot{M} s = s^T M [(\ddot{q} - \ddot{q}_d) + \Lambda(\dot{q} - \dot{q}_d)] + \frac{1}{2} s^T \dot{M} s = -s^T K_d s \leq 0. \quad (34)$$

Recall from the fully actuated derivation that the term $s^T C s$ is added to the first term in brackets from Equation (34) while being subtracted from the second term in brackets. The second term then becomes $s^T \left(\frac{\dot{M}}{2} - C \right) s$. Recall that the first matching condition from Patenaude (2008) involved the quantities that were pre and post multiplied by the velocity vector, much like $s^T \left(\frac{\dot{M}}{2} - C \right) s$. Because $\frac{\dot{M}}{2} - C$ is skew-symmetric the first matching condition is trivially equal to zero. Applying this simplification yields

$$\dot{V} = s^T M [(M^{-1}(-C(\dot{q} - s) - G + \tau) - \ddot{q}_d + \Lambda(\dot{q} - \dot{q}_d))] = -s^T K_d s \leq 0. \quad (35)$$

This eliminates the need to numerically integrate \dot{K}_d due to the first matching condition. It also eliminates the need to introduce \bar{C}_d , C'_d and F_{1m} . If the matrix P in the Lyapunov function from Patenaude (2008) in Equation (18) is set equal to the identity matrix, then this skew-symmetric simplification for $s^T \left(\frac{\dot{M}}{2} - C \right) s$ could also be applied on the first matching condition for that method. This would also eliminate the need to introduce the extra control terms and numeric integration of \dot{K}_d for that control method. Substitutions are made to write Equation (35) in terms of reference velocities and accelerations

$$\dot{V} = s^T [-M\ddot{q}_r - C\dot{q}_r - G + \tau] = -s^T K_d s \leq 0. \quad (36)$$

Equation (36) is solved for τ and the control is written as

$$\begin{aligned} \tau &= M\ddot{q}_r + C\dot{q}_r + G - K_d s \\ &= M(\ddot{q}_d - \Lambda(\dot{q} - \dot{q}_d)) + C(\dot{q}_d - \Lambda(q - q_d)) + G - K_d((\dot{q} - \dot{q}_d) + \Lambda(q - q_d)). \end{aligned} \quad (37)$$

The final steps are defining the remaining $n-m$ desired accelerations to ensure the control is underactuated, ensuring the system stays stable while tracking, and emphasizing tracking characteristics. These are not performed independent of each other, as they have profound impacts on one another.

How to ensure the control is underactuated varies drastically from case to case depending on the dynamics of the system. However, the basic method is the same. For the dynamic system there will be specified position, velocity, and acceleration trajectories for m coordinates. In general these are the coordinates that require more precise tracking.

The desired trajectories for the remaining $n-m$ coordinates need not be explicitly given. The remaining coordinates may need to stay in a bounded region to ensure system stability, or their values may have an influence on the first m equations. However, precision tracking of these coordinates is generally not the objective. The remaining $n-m$ desired accelerations are found to ensure the system is underactuated. If a generalized coordinate had no applied torque, this could be accomplished by setting a row of the control in Equation (37) equal to zero and solving for a desired acceleration. In other cases there may be a reaction torque.

For example the first row and second row of the control law given by Equation (37) could have separate torques acting on them. The third row could have reaction torques that are equal in magnitude and in the opposite direction of the torques applied on the first and second row. The sum of the first, second, and third row could then be solved to equal zero for a desired acceleration. There are an array of possibilities, but these two examples demonstrate simple cases of ensuring the control is underactuated by solving for a single desired acceleration.

This leaves the remaining $n-m$ desired positions and velocities. The solution to these will vary depending on the dynamic system and the coordinates that require precision tracking. In this thesis they are determined by two methods. First they are simply set equal to a constant. This is the case when precision tracking is emphasized on the other coordinates, but the remaining $n-m$ position and/or velocity trajectories need to stay within a certain bounds. The other method used in this thesis is integrating the desired velocity of the remaining $n-m$ coordinates. This is used when the remaining $n-m$ coordinates are used to directly influence the first m coordinates for steering the rolling wheel in Chapter 4. Variations of these methods could easily be implemented

depending on the desired tracking performance. An example would be to use inverse dynamics to determine a suitable desired path to follow.

The final step is to choosing control law components for K_d and Λ in the control law of Equation (37). The choice of these parameters has a few consequences. First they are chosen so that the sliding surface equations in Equation (9) remain stable. This could be trivially performed if the system was fully actuated, as all of the corresponding gains simply need to be positive. The next purpose of the control gains is to place an emphasis on the desired tracking characteristics. How to choose K_d and Λ is best shown through demonstration in Chapters 3, 4, and 5.

Asymptotic tracking convergence can be guaranteed for a fully actuated system. However, in an underactuated system if all the generalized coordinates are given a desired trajectory, generally all the desired trajectories may not be asymptotically tracked. Most often the control places an emphasis on specific coordinates, so that these specific coordinates track asymptotic or near asymptotic. Oftentimes in order to emphasize tracking on the specified coordinates, the remaining coordinates must stay in a stable region.

The second and third matching conditions were originally derived for stabilization, not for tracking in White et al. (2006,2007,2008). Patenaude's tracking control forces these stabilization matching conditions to fit into a Lyapunov control used for tracking. However, the second and third matching conditions can be omitted entirely. These matching conditions don't help provide insight to control gains that will keep the desired coordinates stable. For tracking, they add unnecessary calculations.

It should be noted that dynamic parameters were assumed to be exactly known. However, pre-existing robust or adaptive control strategies could be included in the control law given by Equation (37) for parameter identification. A combination of the presented control law and adaptive control is presented to show the versatility of the proposed control method.

Adaptive Underactuated Sliding Mode Control

Adaptive control differs from other controllers in that the values used in the controller are variable or uncertain. The parameters are adjusted in such a manner that the system is stable. This method often involves the implementation of a Lyapunov function. For the model-reference

adaptive control method there are four main parts: the plant, controller, reference model, and adaptation law.

The plant is assumed to have a specific structure, though the exact parameters are initially uncertain. It is what needs to be controlled.

The controller can come in many forms. It could be calculated by classical linear or state-space methods. It could also come in the form of a non-linear control such as the sliding mode control presented in Equation (37).

The estimates used for the physical values of the controller come from the adaptation law. This is the key idea of adaptive control. The physical estimates are usually linearly related to the generalized coordinates of the plant. If this condition is met, the system is said to be linearly parameterized. Though this is not a necessary condition, most adaptive control designs require linear parameterization to ensure stability and tracking convergence.

A reference model is used to adjust values used in the control law. This is the part of the control that adapts. An unknown mass could be used as a reference model. The adaptation mechanism is used to adjust the reference model parameters used in the control law. The objective of the adaptation is to sync the estimate of the physical values with the control law to ensure the tracking error decays to zero. The adaptive control used by Slotine & Li (1988) is used as a guide in the following derivation. Start with the Lyapunov function

$$V = \frac{1}{2}s^T Ms + \frac{1}{2}\tilde{a}^T \Gamma \tilde{a} \quad (38)$$

where terms similar to those in Equation (10) are the same, \tilde{a} is the difference between the reference model value and the actual value, and Γ is a positive definite matrix of currently unknown values. Taking the time derivative and applying similar simplifications to those used to acquire Equation (36) yields

$$\dot{V} = s^T (\tau - M\ddot{q}_r - C\dot{q}_r - G) + \tilde{a}^T \Gamma \dot{\tilde{a}} = -s^T K_d s. \quad (39)$$

$\dot{\tilde{a}}$ can be written as $\dot{\hat{a}}$, because the actual parameter values are constant. The control law is then chosen as $\tau = \hat{M}\ddot{q}_r + \hat{C}\dot{q}_r + \hat{G} - K_d s$. Again the hat accent stands for the best approximation. The tilde accent stands for the current approximation minus the actual value. Γ is a diagonal positive definite matrix of unknown constant values used for the adaptive part of the control.

The underactuated sliding mode control given in Equation (37) is changed to the estimate of the parameters, instead of the actual parameter so that the control is given by

$$\tau = \hat{M}(q)\ddot{q}_r + \hat{C}(q, \dot{q})\dot{q}_r + \hat{G}(q) - K_d s. \quad (40)$$

The matrix $Y(q, \dot{q}, \ddot{q}_r)$ is introduced so that

$$Y\tilde{a} = \tilde{M}\ddot{q}_r + \tilde{C}\dot{q}_r + \tilde{G}. \quad (41)$$

Now since Y is linear in terms of \tilde{a} , a simplification can be made to find the adaptation law. The adaptation law is

$$\dot{\tilde{a}} = -\Gamma^{-1}Y^T s. \quad (42)$$

Applying Equations (40)-(42) to Equation (39) yields

$$\begin{aligned} \dot{V} &= s^T \left(\tilde{M}\ddot{q}_r + \tilde{C}\dot{q}_r + \tilde{G} - K_v s \right) + \tilde{a}^T \Gamma \dot{\tilde{a}} \\ &= s^T (Y\tilde{a} - K_v s) + \tilde{a}^T \Gamma \dot{\tilde{a}} \\ &= -s^T K_v s. \end{aligned} \quad (43)$$

This shows that the given underactuated sliding mode control presented in this thesis can be made adaptive with little effort.

Chapter Summary

This chapter first presents Lagrange's equation that can be used to derive equations of motion. It then provides the essentials to Direct Lyapunov Stability. A derivation for fully actuated sliding mode control is presented. An underactuated sliding mode control by Patenaude (2008) is presented. The terms are explained and some limitations and difficulties of the method presented by Patenaude (2008) are discussed. With these shortcomings in mind, a new underactuated sliding mode control is derived. Terms are explained and the general solution process is presented. The derivation of the control varies from case to case. Chapters 3, 4, and 5 should be referred to as a guide for choosing the elements of K_d and Λ . Adaptive control laws are often desirable. It was shown that the presented underactuated sliding mode control could have an adaptive term added with little effort. Future research on robustness and other adaptive control strategies is an area of interest.

Chapter 3: Holonomic Examples

This chapter implements the underactuated sliding mode control law on two separate holonomic systems. The sliding mode control will be compared to collocated and non-collocated control as presented by Tedrake (2009). Collocated control is used to asymptotically track the actuated coordinates of a dynamic system, while the path of the unactuated coordinates is not important. Non-collocated control is used to asymptotically track the unactuated coordinates, while the path of the actuated coordinates is not important. An implementation of adaptive control is also presented. Including the adaptive term was shown to increase the sphere of attraction.

The first dynamic system is the cart-pole, which is a simply underactuated system (no applied torque on some generalized coordinates, while there are independent torques applied to the remaining generalized coordinates). Next, the method is applied to a holonomic case of the Segbot.

Inverted-Cart Pole

The inverted pendulum cart system is shown in Figure 1. First a control law will be implemented to stabilize the pendulum around $\theta = 0^\circ$, while precision tracking will be applied so that the x coordinate of the cart tracks a sin wave.

Next, a control law will be presented for precision tracking of a sin wave by the θ coordinate while keeping the x coordinate close to 0. Modifications to the control law will be shown to prioritize tracking performance on a desired coordinate.

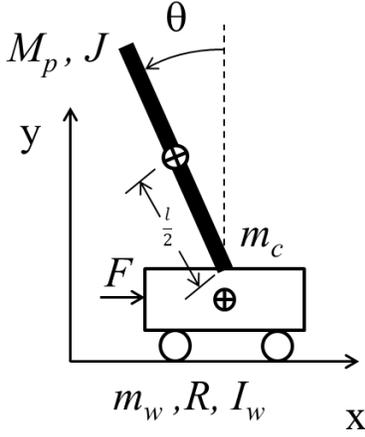


Figure 1: Cart-Pole System

The physical parameters used for the cart-pole system are summarized in Table 1.

Table 1: Cart-Pole Physical Parameters

Variable	Description	Value Used
m_c	mass of the cart	NA
m_p	mass of the pendulum	1.0 [kg]
m_w	mass of the wheel	NA
Mm	Effective mass contribution from cart, pendulum, and wheel masses as well as inertia contribution from the wheels	5.0 [kg]
l	length of the pendulum	0.7 [m]
J	moment of inertia of pendulum about the base	0.2 [kg-m ²]
\oplus	center of mass location	NA

The equations of motion for the cart-pole system are

$$\begin{bmatrix} Mm & -\frac{1}{2}m_p l \cos(\theta) \\ -\frac{1}{2}m_p l \cos(\theta) & J \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{2}m_p l \cos(\theta) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{2}m_p g l \sin(\theta) \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}. \quad (44)$$

Documentation of the derivation for Equation (44) can be found at

http://www.mne.ksu.edu/static/nlc/tiki-index.php?page=Stabilization_Holonomic.

Actuated Coordinate Tracking

For precision tracking of the actuated coordinate, the desired position for the x coordinate was chosen as $x_d = -2 \cos(2t)$ [m]. \dot{x} is calculated by the time derivative. If perfect tracking of the actuated axis was the primary objective, collocated partial feedback linearization could be applied to the dynamics given in Equation (44) Tedrake (2009). The first step is to solve the

second row of Equation (44) for $\ddot{\theta}$. Then insert this value into the first row of Equation (44). Substituting \ddot{x}_d for \ddot{x} will perfectly track the x coordinates. Keep in mind that the collocated tracking does not ensure that the unactuated axis stays stable as will be seen in the results. The control is open loop with respect to θ and its time derivatives. This operation provides

$$\tau = 1/4 \frac{\ddot{x}_d (4 M m J - m_p^2 l^2 (\cos(\theta))^2) - m_p^2 l^2 \cos(\theta) g \sin(\theta) + 2 m_p l \sin(\theta) \dot{\theta}^2 J}{J},$$

$$\ddot{\theta} = \frac{\frac{1}{2} m_p l \cos(\theta) \ddot{x}_d + \frac{1}{2} m_p g l \sin(\theta)}{J}, \quad (45)$$

and

$$\ddot{x} = \ddot{x}_d.$$

The proposed sliding mode control starts off similar to collocated feedback. The second row of the control law given by Equation (37) is solved for the desired unactuated acceleration. Note that the second row of the control law is solved for $\ddot{\theta}_d$ instead of the second row of the dynamic equations for $\ddot{\theta}$. The solution gives

$$\ddot{\theta}_d = \frac{\frac{1}{2} m_p l \cos(\theta) (\ddot{x}_d - \lambda_1 (\dot{x} - \dot{x}_d)) + \frac{1}{2} m_p g l \sin(\theta) + Kd12 (\dot{x} - \dot{x}_d + \lambda_1 (x - x_d)) + Kd22 (\dot{\theta} - \dot{\theta}_d + \lambda_2 (\theta - \theta_d))}{J} + \lambda_2 (\dot{\theta} - \dot{\theta}_d). \quad (46)$$

The final step is to choose values for λ_i as well as the K_d matrix from Equation (37). The values need to be chosen to ensure that θ stays in a stable region, $\pm \frac{\pi}{2} [rad]$. From the control law, it is known that K_d should be positive definite. It is also known that the θ coordinate needs to be stabilized. This suggests that an emphasis on the θ coordinate would be provided by choosing a larger value for λ_2 than for λ_1 .

Linear quadratic regression (*lqr*) or other linear schemes can also be used on the linearized system. One way to choose the parameters in the non-linear control law is to linearize the non-linear control law presented in Equation (37). Then choose the control law components so the linearized control law matches gains that can be found through *lqr* or another linear controller method. The control law components used are shown in Table 2.

Table 2: Cart-Pole Actuated Coordinate Tracking Control Law Values

Constant	Value
λ_1	0.57
λ_2	4.14
Kd_{11}	9.06
Kd_{12}	-15
Kd_{22}	28.93

The results of the collocated control and the underactuated sliding mode control are compared in Figure 2 and Figure 3 for all initial conditions set to 0 with the exception of $x(0) = -2 [m]$.

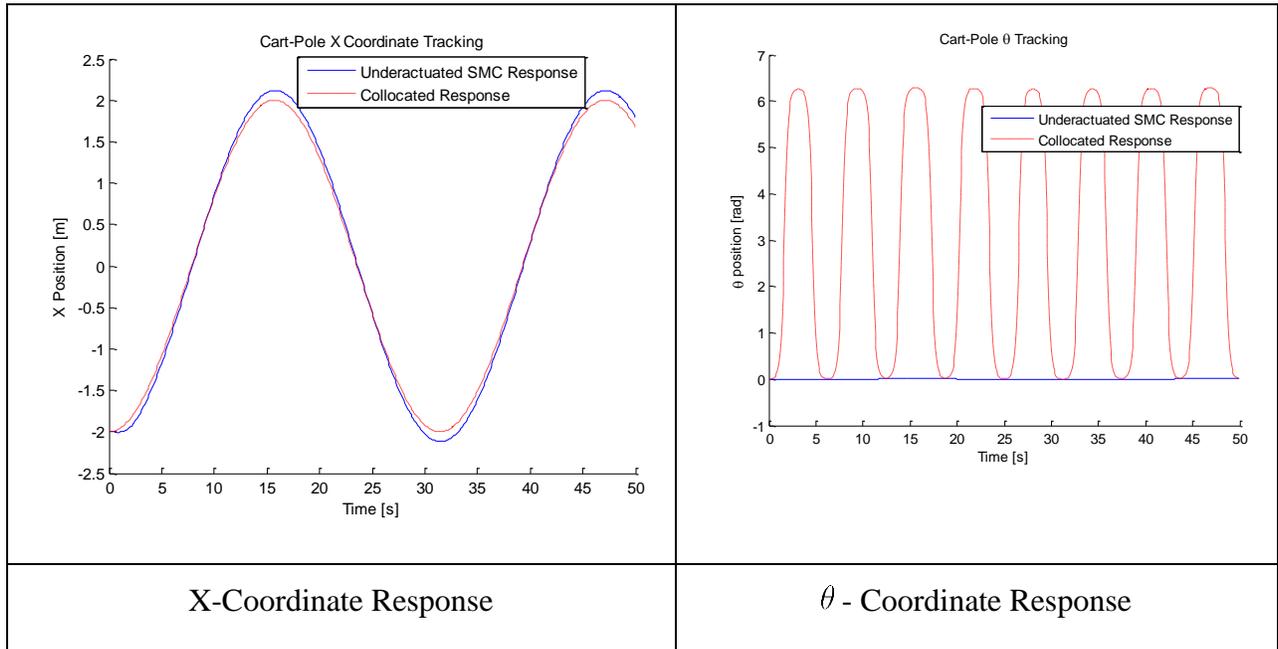


Figure 2: Cart-Pole Actuated Coordinate Tracking Response

The response is as expected. The underactuated sliding mode control law tracks the x -axis reasonably well while still ensuring that θ stays close to zero. The collocated control response perfectly tracks the x -coordinate, however, θ reaches a value that would be too large for actual implementation. The collocated control is open loop with respect to θ and its time derivatives, while the underactuated sliding mode control law is closed loop.

In actual implementation θ can't surpass $\frac{\pi}{2} [rad]$ as the pendulum would strike the ground. If the initial conditions were set up just perfectly, then the θ coordinate may appear to be stable for collocated control. However, since θ and its time derivative are not used for the collocated

control, even a slight deviation between actual and estimated system parameters would yield θ becoming unstable, even with perfectly chosen initial conditions.

Figure 3 shows that unreasonable forces are required for the actual implementation of collocated feedback control. It should be noted that in the underactuated sliding mode control law $\dot{\theta}_d$ and θ_d are set to zero, as this is the stable region that we want the system to stay in.

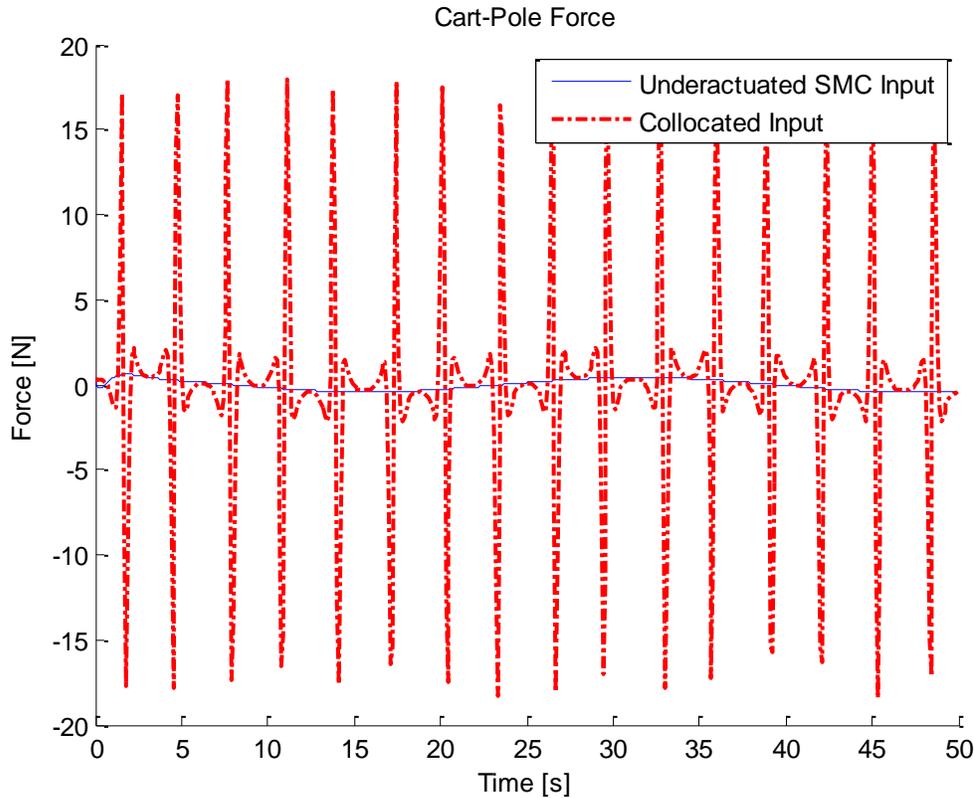


Figure 3: Cart-Pole Actuated Coordinate Force Input

Unactuated Coordinate Tracking

The path $\theta_d = \frac{5\pi}{180} \sin(\omega t)$ [rad] was chosen as the path for the unactuated coordinate to track. $\dot{\theta}_d$ was calculated as the time derivative. The initial conditions were set to match for θ_d and $\dot{\theta}_d$ at $t = 0$. To ensure perfect tracking of the unactuated axis, non-collocated partial feedback linearization can be performed [Tedrake (2009)]. However, implementation of this control makes no promises for the performance of the actuated coordinate. The non-collocated control method yields

$$\begin{aligned}\ddot{x} &= -2 \frac{-J\ddot{\theta}_d + \frac{1}{2} mgl \sin(\theta)}{ml \cos(\theta)}, \\ \ddot{\theta} &= \ddot{\theta}_d, \\ \text{and}\end{aligned}\tag{47}$$

$$\tau = \left(2 \frac{MmJ}{ml \cos(\theta)} - \frac{1}{2} ml \cos(\theta) \right) \ddot{\theta}_d - \frac{Mm g \sin(\theta)}{\cos(\theta)} + \frac{1}{2} ml \sin(\theta) \dot{\theta} \dot{x}.$$

Note, that in order to track θ , the desired path must not contain any values where $\cos(\theta)$ is equal to zero.

To implement the underactuated sliding mode control, the dynamic matrices from Equation (44) are inserted into the sliding mode control law given by Equation (37) and the second row is solved for \ddot{x}_d to yield

$$\ddot{x}_d = -2 \frac{-J \left(\ddot{\theta}_d - \lambda_2 \left(\dot{\theta} - \dot{\theta}_d \right) \right) + 1/2 m_p g l \sin(\theta) + Kd21 s1 + Kd22 s2}{m_p l \cos(\theta)} + \lambda_1 (\dot{x} - \dot{x}_d).\tag{48}$$

The chosen gains are listed in Table 3.

Table 3: Cart-Pole Underactuated Coordinate Control Law Values

Constant	Value
λ_1	0.26
λ_2	4.18
$Kd11$	16.55
$Kd12$	-1.42
$Kd22$	1.2

The results for the non-collocated control law and the underactuated sliding mode control law are compared in Figure 4. All initial conditions match the ideal conditions at $t = 0$.

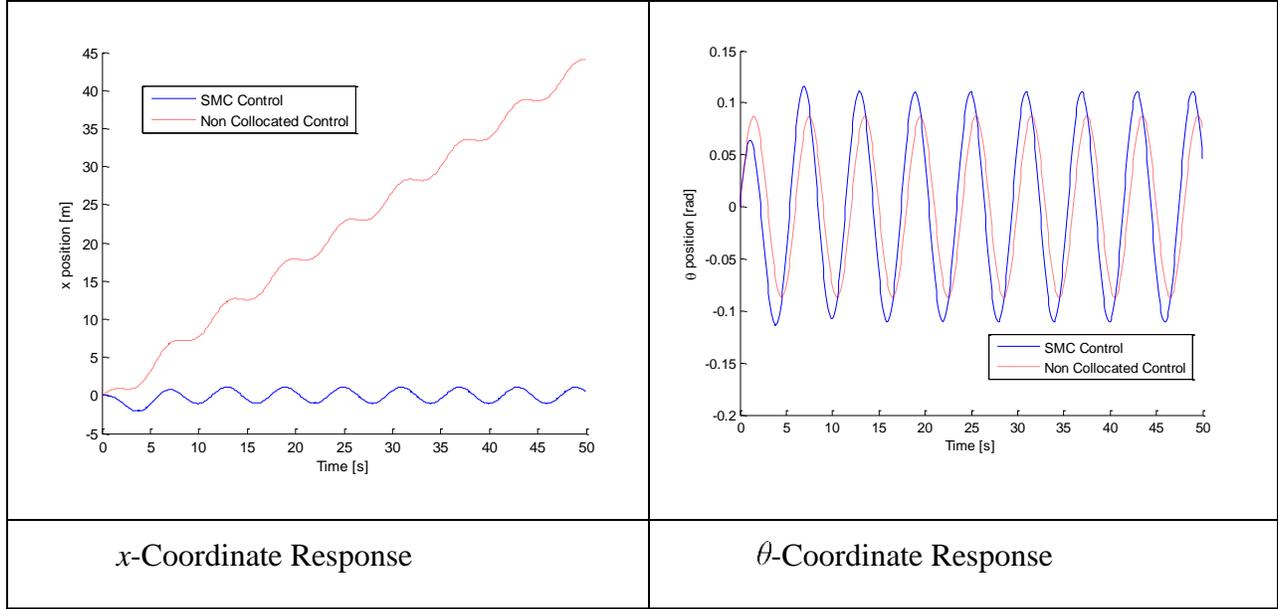


Figure 4: Cart-Pole Underactuated Coordinate Tracking

The non-collocated control tracks θ coordinates perfectly. However, the x -coordinate increases throughout time with these initial conditions. Figure 4 shows that the sliding mode control tracks the desired θ coordinates reasonably well, while also keeping x reasonably close to 0. This demonstrates that the proposed control method can be implemented on a 2 DOF simply underactuated dynamic system with little effort.

Cart-Pole Adaptive Control Implementation

This section shows the advantages provided by including the adaptive term given by Equation (42). The parameters are left the same as the previous ***Actuated Coordinate Tracking*** section with one exception. The value used for Mm in the control from Table 1 was equal to 28 [kg] instead of 5 [kg], while the actual dynamics retained Mm at 5 [kg]. This simulation was compared to an adaptive control simulation. Γ from Equation (42) was set equal to 0.1 for the adaptive control. The initial guess for Mm was set to 28[kg] for the adaptive control, while the actual value used by the dynamics was again left at 5 [kg].

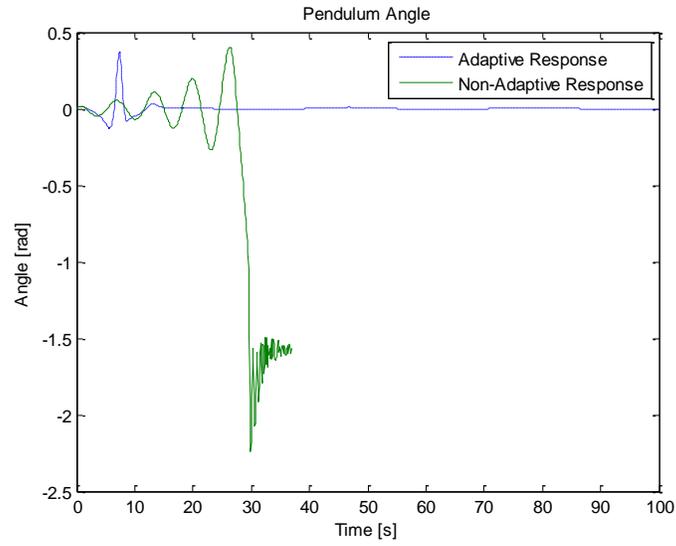


Figure 5: Adaptive Control Pendulum Angle Response

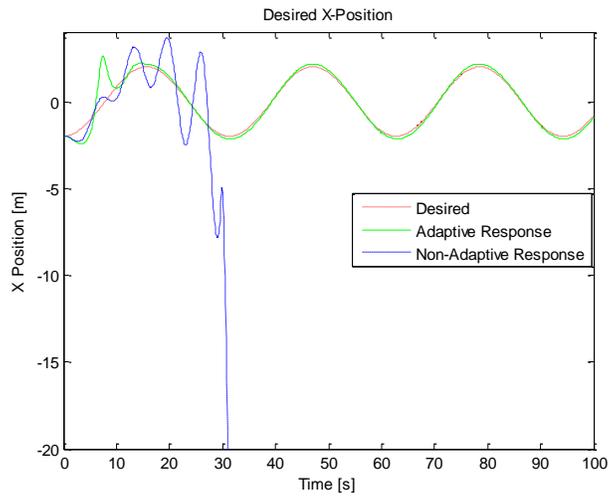


Figure 6: Adaptive Control X-Coordinate Tracking

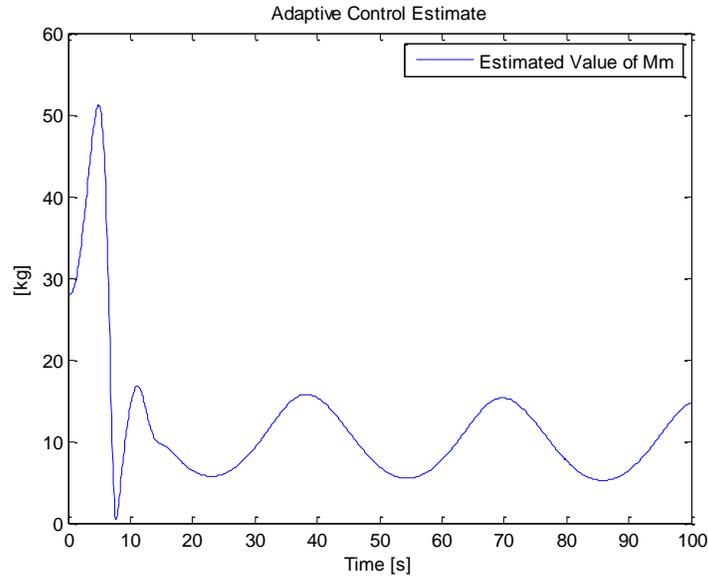


Figure 7: Adaptive Control Estimated Value of Mm

The results confirm that controller performance can be increased by including the adaptive control term. The non-adaptive control response goes unstable. The simulation that includes the adaptive control term has a little trouble finding an estimate for Mm at first. However, after about 15 seconds Mm begins to reach its final value and the performance is very similar to as if the parameter Mm was exactly known. For a detailed explanation on why Mm does not converge to the actual value refer to Slotine & Li (1991). The shows the underactuated sliding mode control method can accommodate an adaptive control term that may help increase performance.

Holonomic Segbot

The Segbot is a self-balancing two wheeled robot that draws many similarities to the cart-pole system. The Segbot is shown in Figure 8.

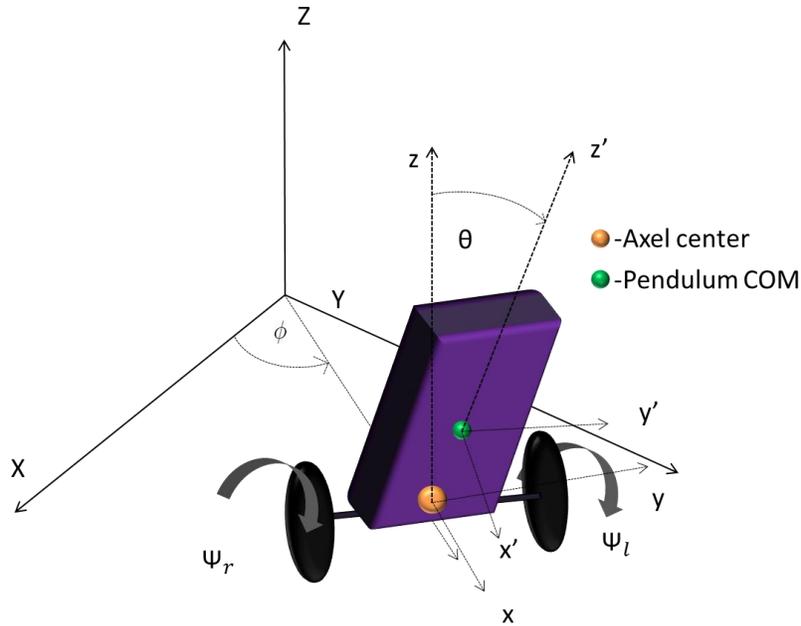


Figure 8: Segbot

To derive the equations of motion, the Segbot was modeled as 3 rigid bodies: these being the left wheel, the right wheel, and the pendulum body. For the holonomic Segbot model, the left wheel and right wheel angles are set equal to each other. This simplifies the dynamics so that there are only 2 rigid bodies are used for the simplified version of the Segbot. This also means that ψ_r and ψ_l and their time derivatives are equal to each other. The Segbot cannot steer in the global XY plane. This makes the system even more similar to the cart-pole system. There is one major difference, a reaction torque on the pendulum. The equations of motion were derived using Newton-Euler and then confirmed using Lagrange's Equation. A reiteration that for the holonomic case the Segbot dynamics are constrained so that it can only move straight forward and straight backwards.

Equations of Motion Using Newton Euler

The free body diagram for the simplified Segbot is shown in Figure 9.

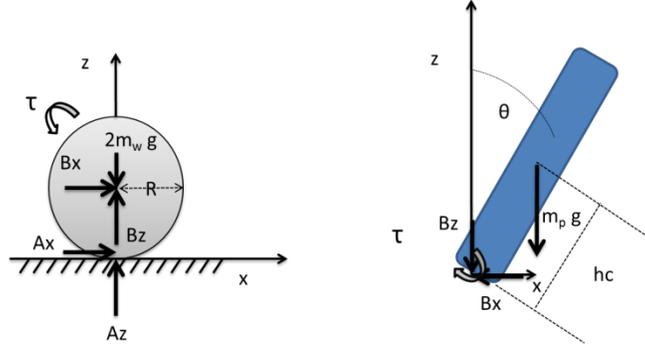


Figure 9: Simplified Segbot FBD

Newton-Euler applied to the wheel provides the relations of

$$\begin{aligned}
 \sum F_x = ma_x &\Rightarrow A_x + B_x = 2m_w \ddot{x}, \\
 \sum F_z = ma_z &\Rightarrow A_z + B_z - 2m_w g = 0, \\
 &\text{and} \\
 \sum \tau = I\alpha &\Rightarrow m_w R^2 \ddot{\psi} = \tau - A_x R.
 \end{aligned} \tag{49}$$

Doing the same for the Segbot body yields

$$\begin{aligned}
 \sum F_x = ma_x &\Rightarrow -B_x = m_p \ddot{x}_c, \\
 \sum F_z = ma_z &\Rightarrow m_p \ddot{z}_c = -m_p g - B_z, \\
 &\text{and} \\
 \sum \tau = I\alpha &\Rightarrow I_\theta \ddot{\theta} = -\tau + B_x \cos(\theta) h_c - B_z \sin(\theta) h_c.
 \end{aligned} \tag{50}$$

\ddot{x}_c and \ddot{z}_c are the pendulum center of mass accelerations. The equations given by Equations (49) and (50) need to be simplified. Notice that the mass of the wheels is assumed to be the same. Wheel masses were included separately and assumed to be the same. Also, the pendulum body is modeled so that when $\theta = 0$ the center of mass is exactly above the axel. \ddot{x}_c and \ddot{z}_c are calculated at the pendulum body center of mass. These can be written as the sum of the axel center acceleration and the relative acceleration of the pendulum center of mass with respect to the axel center. Writing the center of mass accelerations relative to the axel accelerations yields

$$\begin{aligned}
 \ddot{x}_p &= \ddot{x} + \ddot{\theta} h_c \cos(\theta) - \dot{\theta}^2 h_c \sin(\theta) \\
 &\text{and} \\
 \ddot{z}_p &= -\ddot{\theta} h_c \sin(\theta) - \dot{\theta}^2 h_c \cos(\theta).
 \end{aligned} \tag{51}$$

There is also a holonomic kinematic relation between x and ψ . Because of the holonomic nature, the same holds true for the time derivatives of the constraint. The holonomic constraint is given by

$$x = R\psi. \quad (52)$$

The time derivatives are given by

$$\dot{x} = R\dot{\psi} \text{ and } \ddot{x} = R\ddot{\psi}. \quad (53)$$

Because integration of the constraint equation provides a unique solution to the Segbot position, the system is holonomic. All of the terms that involve time derivatives of x are converted to the corresponding time derivative of ψ through the holonomic constraint. There are a few reasons for choosing ψ instead of x . First, ψ is the generalized coordinate on which the torque is applied. Also, in implementation encoders can be used to measure angular displacement of the wheels.

The simplifications from Equations (51) – (53) are applied to the moment sum of the pendulum in Equation (50). This yields the second equation of motion in Equation (54).

Apply the simplifications given by Equations (51) – (53) are applied to the sum of the horizontal forces on the wheel in Equation (49). The result is solved for the reaction force Ax . This value for Ax is inserted into the moment sum for the wheel in Equation (49) to obtain the top row of Equation (54). The combined equations of motions are then given by

$$\begin{bmatrix} 3m_w R^2 + m_p R^2 & m_p \cos(\theta) h_c R \\ m_p \cos(\theta) h_c R & m_p h_c^2 + I_{theta} \end{bmatrix} \begin{bmatrix} \ddot{\psi} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & -\dot{\theta} m_p \sin(\theta) h_c R \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ -h_c \sin(\theta) m_p g \end{bmatrix} = \begin{bmatrix} \tau \\ -\tau \end{bmatrix}. \quad (54)$$

Equations of Motion Using Lagrange

The kinetic energy of the wheels can be written as the sum of the translational and rotational energies of the wheels or the parallel axis theorem can be used to find the moment of the wheel about its contact point with the ground. Either method gives the same result. The total kinetic energy of the wheel is

$$KE_w = \frac{3}{2} m_w R^2 \dot{\psi}^2. \quad (55)$$

The position of the pendulum center of mass is written as the sum of the position of the axel center and the position of the pendulum center of mass with respect to the axel center. This is differentiated with respect to time to find the velocities of the pendulum center of mass. The pendulum center of mass position and velocities are given by

$$r_p = r + r_{p/o} = \begin{bmatrix} x \\ R \end{bmatrix} + \begin{bmatrix} \sin(\theta) h_c \\ \cos(\theta) h_c \end{bmatrix}$$

and

(56)

$$\frac{d}{dt}(r_p) = \frac{d}{dt}(r) + \frac{d}{dt}(r_{p/o}) = \begin{bmatrix} \dot{x} + \dot{\theta} \cos(\theta) h_c \\ -\dot{\theta} \sin(\theta) h_c \end{bmatrix}.$$

The kinetic energy of the pendulum is found as a sum of the translational and rotational components. Because Lagrange's Method uses the rate of energy, the potential energies of the wheels are omitted as they remain constant. The kinetic and potential energy of the pendulum are given by

$$KE_p = KE_t + KE_r = \frac{1}{2} m_p \left(\left(\dot{x} + \dot{\theta} \cos(\theta) h_c \right)^2 + \left(\dot{\theta} \sin(\theta) h_c \right)^2 \right) + \frac{1}{2} I_{theta} \dot{\theta}^2$$
(57)

and

$$PE = h_c \cos(\theta) m_p g, \text{ respectively.}$$
(58)

The Lagrangian is constructed as presented in Equation (1). The holonomic relation between x and ψ is used to eliminate x and its time derivatives. The Lagrangian is

$$L = \frac{3}{2} m_w R^2 \dot{\psi}^2 + \frac{1}{2} m_p \left(\left(R\dot{\psi} + \dot{\theta} \cos(\theta) h_c \right)^2 + \dot{\theta}^2 (\sin(\theta))^2 h_c^2 \right) + 1/2 I_{theta} \dot{\theta}^2 - h_c \cos(\theta) m_p g.$$
(59)

Applying Lagrange's equation given by the left hand side of Equation (1) yields

$$\begin{aligned} \frac{d}{dt} \left(\frac{dL}{d\dot{\psi}} \right) - \frac{dL}{d\psi} &= (3 m_w R^2 + m_p R^2) \ddot{\psi} + m_p \cos(\theta) h_c R \ddot{\theta} - m_p \dot{\theta}^2 \sin(\theta) h_c R m_p \cos(\theta) h_c R \dot{\psi} \\ \frac{d}{dt} \left(\frac{dL}{d\dot{\theta}} \right) - \frac{dL}{d\theta} &= \left(\frac{1}{2} m_p (2 (\cos(\theta))^2 h_c^2 + 2 (\sin(\theta))^2 h_c^2) + I_{theta} \right) \ddot{\theta} + \frac{1}{2} m_p \left(2 \dot{\theta} \sin(\theta) h_c^2 \cos(\theta) - 2 (R\dot{\psi} + \dot{\theta} \cos(\theta) h_c) \sin(\theta) h_c \right) \dot{\theta} \\ &\quad - \frac{1}{2} m_p \left(-2 (R\dot{\psi} + \dot{\theta} \cos(\theta) h_c) \dot{\theta} \sin(\theta) h_c + 2 \dot{\theta}^2 \sin(\theta) h_c^2 \cos(\theta) \right) - h_c \sin(\theta) m_p g. \end{aligned}$$
(60)

Equation (60) can be shown to be the same as Equation (54). As shown in Equation (1), the non-conservative force terms are placed into Q . For this example viscous friction was omitted. It

should be noted that the non-conservative forcing terms must be applied in the direction of a generalized coordinate.

It is assumed that a motor applies a torque in the positive ψ direction. There will also be a torque in the opposite direction on the pendulum body. The torque applied on the pendulum body may or may not be the same magnitude as the torque applied on the wheel. For this case it was assumed that the gear ratio is also equal to the applied torque ratio. After inserting the torque elements into Q , the equations of motion match those found by using Newton-Euler.

Tracking Control

The ψ -coordinate was chosen to precisely track because in implementation the desired trajectory for the Segbot to follow would most likely be global XY coordinates. These are related to ψ through non-holonomic roll without slip constraints and will be looked at in more detail in Chapter 4. The steps to solve for $\ddot{\theta}_d$ so that the applied torque meets the underactuated constraints imposed by the dynamics of the Segbot are outlined as:

- Insert the dynamic terms from Equation (54) into the control law given by the proposed sliding mode control in Equation (37).
- From the ratio of the applied torques in the Segbot equations of motion, observe that the first row is equal in magnitude but in the opposite direction of the second row.
- The first row of the control law is added to the second row of the control law. The dynamic equations show this sum must be equal to zero.

The sum is then solved for $\ddot{\theta}_d$ to yield

$$\begin{aligned} \ddot{\theta}_d = & \frac{-1}{m_p h_c^2 + m_p \cos(\theta) h_c R + I_{theta}} ((3 m_w R^2 + m_p R^2) (\ddot{\psi}_d - \lambda I (\dot{\psi} - \dot{\psi}_d)) + \dot{\theta} m_p \sin(\theta) h_c R (\dot{\theta}_d - \lambda 2 (\theta - \theta_d)) + Kd11 (\dot{\psi} - \dot{\psi}_d + \lambda I (\psi - \psi_d)) \\ & + Kd12 (\dot{\theta} - \dot{\theta}_d + \lambda 2 (\theta - \theta_d)) + m_p \cos(\theta) h_c R \lambda 2 \dot{\theta} - m_p \cos(\theta) h_c R \lambda 2 \dot{\theta}_d - m_p \cos(\theta) h_c R \ddot{\psi}_d + m_p \cos(\theta) h_c R \lambda I \dot{\psi} - m_p \cos(\theta) h_c R \lambda I \dot{\psi}_d \\ & + m_p h_c^2 \lambda 2 \dot{\theta} - m_p h_c^2 \lambda 2 \dot{\theta}_d + I_{theta} \lambda 2 \dot{\theta} - I_{theta} \lambda 2 \dot{\theta}_d + h_c \sin(\theta) m_p g + Kd21 \dot{\psi} - Kd21 \dot{\psi}_d + Kd21 \lambda I \psi - Kd21 \lambda I \psi_d \\ & + Kd22 \dot{\theta} - Kd22 \dot{\theta}_d + Kd22 \lambda 2 \theta - Kd22 \lambda 2 \theta_d) \end{aligned} \quad (61)$$

where λ_i is the i^{th} diagonal element of Λ , $Kdij$ is the corresponding element of the Kd control matrix from the proposed control of Equation (37), I_{theta} is the moment of inertia from the Segbot center of mass about the body centered y' axis, and I_{phi} is the moment of inertia from the Segbot center of mass about the body centered z' axis.

Equation (61) provides $\ddot{\theta}_d$ to ensure the sum of the first and second row of the control law given by Equation (37) is always zero. This step is taken to ensure the control law fits the underactuation given by the dynamic system. The Segbot has a reaction torque. For the cart-pole system this step ensured the θ coordinate equation was unactuated.

The control law in Equation (37) is then evaluated at this desired acceleration and the given plant dynamics. The remaining steps are to choose the desired position and velocity for the θ coordinate and to choose the control constants given by Λ and K_d in Equation (37).

The dynamic reaction to the control law must stay in a stable region for θ . Since the actual position of θ tracks the position of θ_d , the differential equation describing θ_d also needs to stay stable. The desired position and velocity for θ are set to zero as this is the center of the region where the Segbot is stable. The system parameters were chosen as shown in Table 4. The control law values were chosen so that Λ and K_d were positive definite.

Table 4: Simplified Segbot System Parameters

Control	Value	System Paramter	Value
λ_1	3	m_p	1[kg]
λ_2	5	W	0.15[m]
Kd_{11}	10	R	0.1016[m]
Kd_{12}	-5	h_c	0.3 [m]
Kd_{21}	50	g	9.81[m/s ²]
Kd_{22}	120	N	10
		I_{theta}	0.0027 [kg-m ²]

The sliding mode control was then linearized about $\theta = 0, \dot{\theta} = 0$. The linear feedback gains that result from this linearization were calculated as

$$K = [18.70 \quad 216.14 \quad 6.21 \quad 43.04] \quad (62)$$

where the first gain corresponds to ψ , the second corresponds to θ , the third corresponds to $\dot{\psi}$, and the fourth corresponds to $\dot{\theta}$. The block diagram for trajectory tracking while using linear feedback gains on the linearized dynamics is shown in Figure 10.

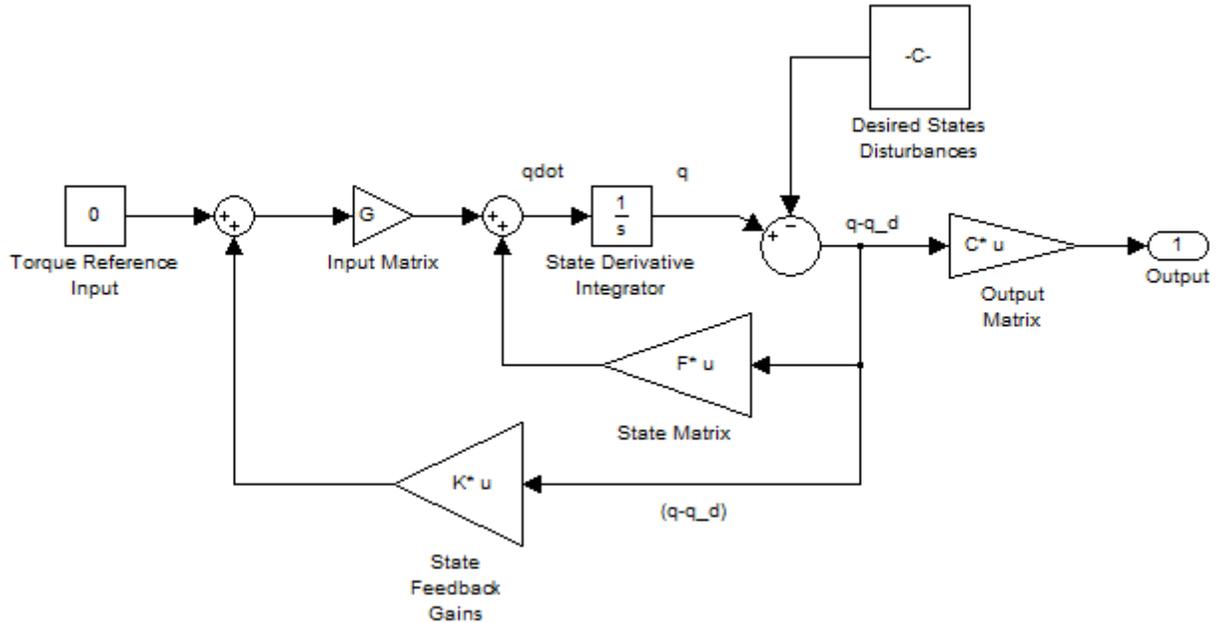


Figure 10: State Space Feedback Gains Block Diagram

The specified path to track was defined as $\psi_d = 10(1 - \cos(t))$, while keeping θ stabilized. The initial pendulum angle was defined at a very aggressive $\frac{70\pi}{180}$ [rad], while $\dot{\psi}$ and $\dot{\theta}$ were set to zero. $\frac{70\pi}{180}$ [rad] is an initial condition well out of the range where the linearized model about 0 [rad] is an appropriate approximation. If the non-linear control is linearized and applied at these initial conditions, the system will go unstable. Simulations are provided in Figure 11.

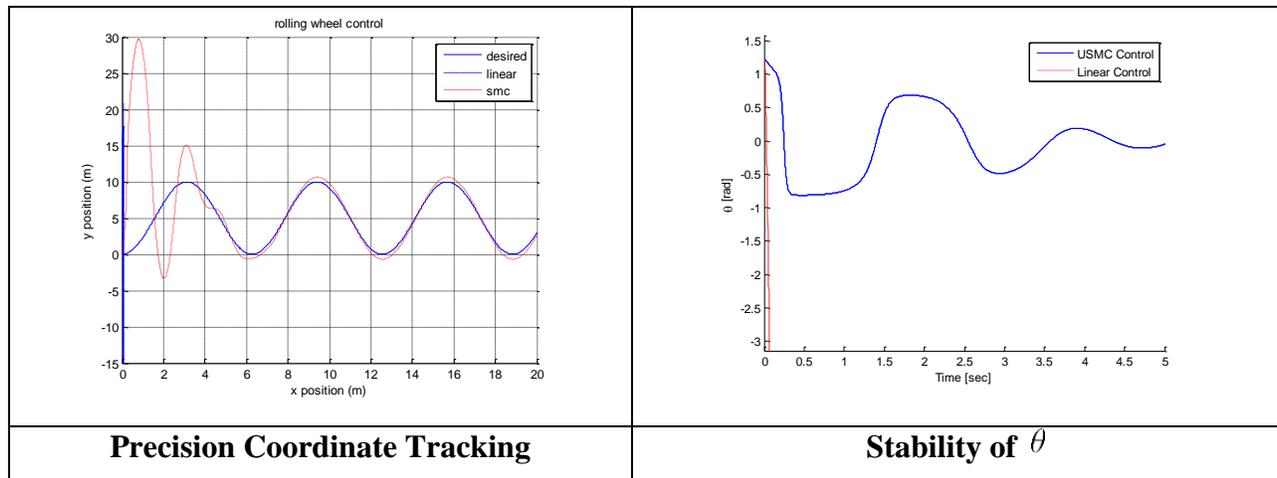


Figure 11: Holonomic Segbot Tracking Response Under Adverse Initial Conditions

The linear response for ψ in Figure 11 reaches a magnitude of 10^4 in less than 0.1 seconds and the response is unstable. The y-scale was set so the response for the USMC control could be viewed, so the y position response for the linear control appears nearly on top of the y-axis in Figure 11.

The aggressive initial condition for θ was chosen to demonstrate that the non-linear sliding mode control increases the sphere of attraction for the dynamic system. If the corresponding linear control is used on the system it immediately goes unstable.

Utilizing the Proposed Control for Stabilization

It was then decided to test the same control derived in the previous *Tracking Control* section to stabilize the dynamics at $\psi_d = 0$ instead of tracking $\psi_d = 10(1 - \cos(t))$. All the parameters remained the same except ψ_d and its time derivatives were set to zero. The results are shown in Figure 12.

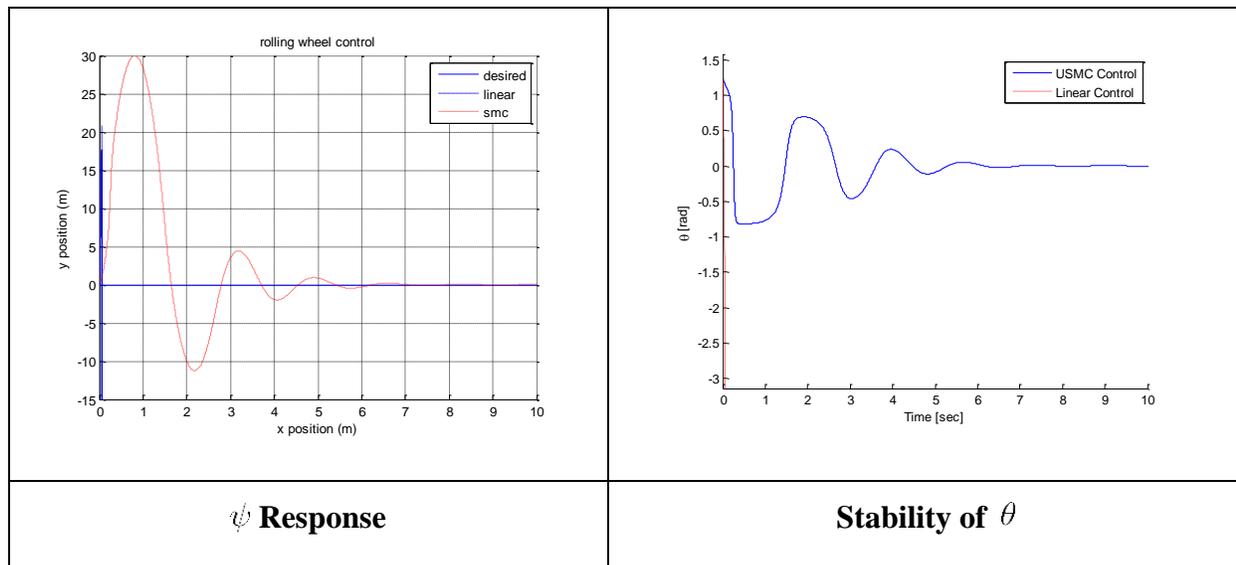


Figure 12: Holonomic Segbot Stabilization Under Adverse Initial Conditions

Again the proposed control stays stable, while the corresponding linear controller immediately falls into an unstable region. The y-position dynamics response for the linear control is nearly on top of the y-axis in Figure 12. The y-axis scale was set to best view the system's dynamic response to the non-linear control law.

Chapter Summary

This chapter demonstrated that the presented underactuated sliding mode control law can be implemented on simple holonomic mechanical systems with little effort. It showed that the control law is applicable on a simply underactuated system in the cart-pole system. This example demonstrated that precision tracking can be applied to either the actuated or unactuated coordinates. It is also shown that the control law is versatile and can include an adaptive term. This chapter also showed the applicability of the control law on a reactionary torque system in the Segbot. It was observed that adverse conditions could be overcome by the control law.

Just because the control law components make the matrices K_d and Λ positive definite, there is no guarantee that the system dynamics will stay stable. The energy may begin to dissipate, but if the θ coordinate from the examples in this chapter leaves the sphere of attraction of the control law, then control is lost.

The presented sliding mode control law is implemented for coordinate trajectory tracking on two underactuated dynamic systems with minimal calculations. The first step is to find control gains that will stabilize the system. These gains for K_d and Λ can often be found by matching a linearized version of the sliding mode control to linear gains. The linear feedback gains can be found through *lqr* or other linear methods. However, the presented sliding mode control was shown to have an increased sphere of attraction and exhibited better trajectory tracking capabilities when compared to the linear control from which the non-linear gains were derived.

Chapter 4: Non-Holonomic Systems

This chapter covers the implementation of the proposed underactuated tracking control law on two separate non-holonomic examples. The presentation of a navigational control is presented to address how to kinematically track a given trajectory that is related to the dynamic system through non-holonomic constraints. The implementation includes a closed loop X and Y coordinate control scheme. The dynamic system needs to be able to track the path even under disturbances. The control is tested on two dynamics systems through simulation.

The first dynamic system is the rolling wheel as presented by Patenaude (2008). The linearized version of this dynamic system is uncontrollable. This presents a new challenge while choosing the control law components. In previous work the wheel was given a desired XY path. Note that the XY path refers to the contact location of the wheel as it rolls in the XY plane. If the initial conditions were perfect, the previous controller provided by Patenaude (2008) in Equation (15) could precisely control the wheel. However, the controller actually was open loop for the X and Y coordinates. This thesis uses a navigational control to help the rolling wheel track the contact position as precisely as possible.

The Segbot is the next dynamic system studied. Note that XY refers to the projection of the axel center in Figure 8 onto the global XY plane. In this chapter the generalized coordinates to track are chosen as X and Y to make the problem non-holonomic. Some problems choosing the control parameters are presented and accounted for.

Navigational Control

Typically it is easier to design underactuated control laws where $n-m$ is as small as possible, where n refers to the number of generalized coordinates, and m refers to the number of actuators. While deriving the equations of motion for the non-holonomic problems, there is a point where a substitution can be made to reduce the value of $n-m$ by two. This is accomplished by removing X and Y from the dynamic equations. This is performed to make the inner loop tracking control as simple as possible. The sliding mode control presented in Chapter 2 is used as the inner loop control. An alternative to introducing the navigational control would be to have two extra equations of motion (the equations of motion that come from the generalized

coordinates X and Y and Lagrange's method to determine the equations of motion) as well as the corresponding generalized coordinates.

Removing the XY coordinates from the inner loop control does not make it impossible to track a given trajectory for the X and Y coordinates. A navigational control computes desired values of generalized coordinates in the inner control loop to compensate for X and Y error. The generalized coordinates in the inner loop are related to X and Y through a roll without slip constraint. A visual representation of the process is presented in Figure 13.

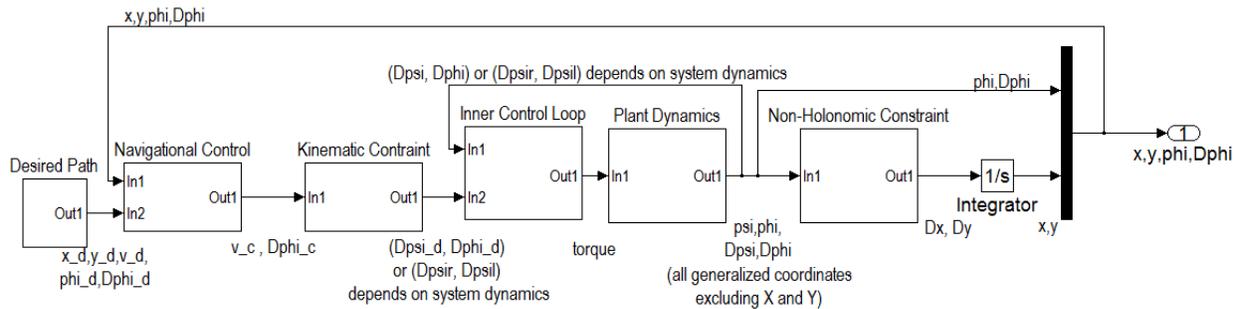


Figure 13: Navigational Control

The input to the navigational control is $X_d, Y_d, \dot{\phi}_d, \dot{\phi}_d$ the actual X position, the actual Y position, and ϕ . All of the desired trajectories are provided by the desired path subsystem in Figure 13. For the Segbot, ϕ can be observed as the right-hand-rule positive rotation about the global Z -axis in Figure 8.

The navigational control outputs a translational and rotational velocity for the dynamic system that compensates for X and Y errors. This is the desired translational and desired rotational velocity that is used by the inner control loop. It is labeled with v_c (for compensational) in Figure 13. These compensational velocities are used to find desired coordinate velocities in the inner control loop. The velocities can often be related to the generalized coordinates of the dynamic system through non-holonomic roll without slip constraints. The translational and angular velocity must undergo a kinematic constraint to fit the dynamic system. For the rolling wheel example (as shown in Figure 16) in this chapter, the desired translational velocity is converted to $\dot{\psi}$ while the rotational velocity is converted to $\dot{\phi}$. For the Segbot example, the desired translational and angular velocities are converted to $\dot{\psi}_r$ and $\dot{\psi}_l$ as can be viewed later in Figure 30.

These desired coordinates are then used in conjunction with the actual coordinates of the dynamic system by the inner control loop. The sliding mode control derived in Chapter 2 of this thesis is used as the inner control loop. Alternatively, the inner control loop could be the previously proposed control by Patenaude (2008) or linear feedback gains.

Depending on the hardware available it may be much simpler to implement encoders to calculate wheel positions and velocities to estimate \dot{X} and \dot{Y} than directly measuring them. Often the desired coordinates to track are X and Y for non-holonomic systems that involve roll without slip constraints. These coordinates are related to the dynamics of the system through non-holonomic constraints that takes the form

$$\dot{X} = R\dot{\psi} \cos(\phi) \text{ and } \dot{Y} = R\dot{\psi} \sin(\phi) \quad (63)$$

where R is the wheel radius, $\dot{\psi}$ is an angular wheel velocity, and ϕ is the heading direction.

Figure 13 shows a closed-loop control scheme to track the desired X and Y contact position.

Open loop (X, Y) position control is represented in Figure 14. A kinematic constraint is imposed directly after the X_d and Y_d coordinates are specified to give the desired coordinates used by the dynamic system. The output of the kinematic constraint does not get fed to a navigational control. Instead the output is fed directly to the inner control loop.

Also fed into the inner control loop are the actual generalized coordinates. State measurements must be obtained from the system dynamics to obtain these. Sometimes X and Y cannot be measured directly and a kinematic constraint is used to find \dot{X} and \dot{Y} , which can be integrated to calculate X and Y .

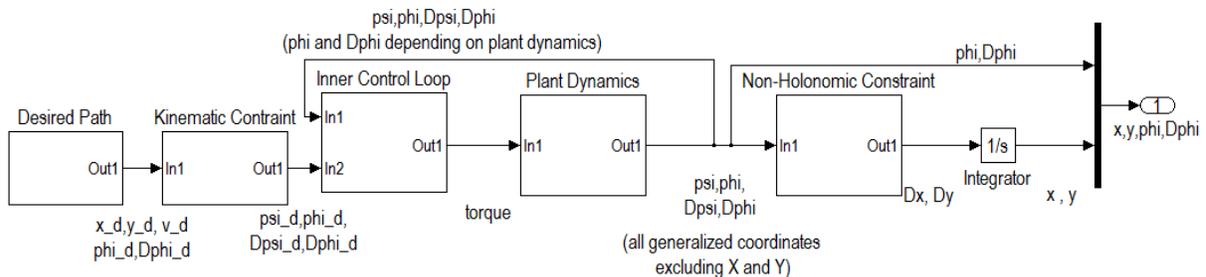


Figure 14: Open Loop XY Tracking

The main difference between Figure 13 and Figure 14 is that Figure 13 uses feedback of the current X and Y position for the navigational control, and $\dot{\phi}$ and $\dot{\psi}$ for the inner control loop. Figure 14 uses feedback of $(\psi, \phi, \dot{\psi}, \dot{\phi})$ for the inner control loop and has no feedback of X or Y .

Integrating non-holonomic constraints does not provide a unique solution to the location of all the generalized coordinates. For visualization consider the Segbot. Imagine that the left wheel stays in place, while the right wheel makes the Segbot rotate 360° . Now imagine that the right wheel stays in place while the left wheel makes the Segbot rotate 360° in the other direction. In either case the axel center ends up in the same position, while the wheel angles have changed.

Now if the wheel angles both change at an equal angular rate for an amount of time so that the overall angular displacements match those in the previous paragraph, the Segbot would move forward in a straight line. This results in a new XY coordinate, but with the same angles for ψ_l and ψ_r , as the previous case which resulted in the Segbot axel center ending in the same place as it started.

The purpose of the navigational control is to accept the current position given by (X, Y, ϕ) , as well as the desired coordinates $(X_d, Y_d, \phi_d, \dot{\phi}_d)$, and output a compensating translational velocity and a compensating rotational velocity. This allows tracking of X_d and Y_d without implicitly using X and Y values in the inner control loop.

The dynamic equations of motion that are found using X and Y as generalized coordinates, the non-holonomic constraints given by Equation (63), and the time derivatives of the non-holonomic constraints provide a unique solution for $\dot{X}, \dot{Y}, \ddot{X}, \ddot{Y}$, and the Lagrange multipliers from Equation (3). This simplification reduces the number of equations of motion by two. It also removes X and Y as well as their time derivatives from the remaining equations of motion and makes control laws easier to derive. This step is performed because in general the smaller $n-m$ is in an underactuated system, the easier the inner loop control law is to evaluate. Because of the navigational control, tracking of X_d and Y_d can still occur.

Navigational Control Derivation

The derivation of the navigational control begins with an object that is subject to the non-holonomic constraints

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \dot{\phi} \end{bmatrix}. \quad (64)$$

In the constraints, v is the translational velocity and $\dot{\phi}$ is the angular velocity. Figure 15 shows the actual position of the object as a triangle.

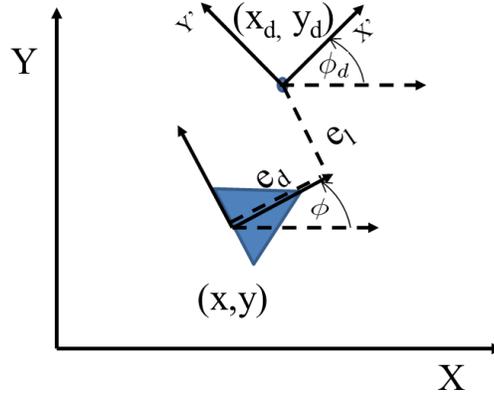


Figure 15: Navigational Control Posture Error

The actual position of the object is given as (x, y, ϕ) , while the desired position of the object is given by (x_d, y_d, ϕ_d) . x and y are the position of the object, while ϕ is the orientation.

Kanayama (1990) suggests defining the posture error as the error between the current coordinates and the desired coordinates in the local body fixed coordinate system of the object being controlled shown in Figure 15 as

$$P = \begin{bmatrix} e_d \\ e_l \\ \phi_e \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \phi_d - \phi \end{bmatrix}. \quad (65)$$

The time derivative of the posture error can be written as

$$\dot{P} = \begin{bmatrix} \dot{e}_d \\ \dot{e}_l \\ \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} \dot{\phi} e_l - v + v_d \cos(\phi_e) \\ -e_d \dot{\phi} + v_d \sin(\phi_e) \\ \dot{\phi}_d - \dot{\phi} \end{bmatrix} \quad (66)$$

where e_d is the position error measured along the local drive direction, e_l is the position error measured in the local lateral direction, ϕ_e is the error in the heading direction, v is the compensational translational velocity found by the navigational control, v_d is the desired translational velocity, and the subscript e stands for the desired position minus the actual

position. Note that e_d and e_l are orthogonal directions. Equation (66) may not be intuitive at first; the simplification for \dot{e}_d and \dot{e}_l are given as

$$\begin{aligned}
\dot{e}_d &= (\dot{x}_d - \dot{x}) \cos(\phi) + (\dot{y}_d - \dot{y}) \sin(\phi) - (x_d - x) \dot{\phi} \sin(\phi) + (y_d - y) \cos(\phi) \dot{\phi} \\
&= \dot{x}_d \cos(\phi) + \dot{y}_d \sin(\phi) - v + \dot{\phi} e_l \\
&= \dot{\phi} e_l - v + \cos(\phi_d - \phi_e) \dot{x}_d + \dot{y}_d \sin(\phi_d - \phi_e) \\
&= \dot{\phi} e_l - v + \dot{x}_d (\cos(\phi_d) \cos(\phi_e) + \sin(\phi_d) \sin(\phi_e)) + \dot{y}_d (\sin(\phi_d) \cos(\phi_e) - \cos(\phi_d) \sin(\phi_e)) \\
&= \dot{\phi} e_l - v + (\dot{x}_d \cos(\phi_d) + \dot{y}_d \sin(\phi_d)) \cos(\phi_e) + (\dot{x}_d \sin(\phi_d) - \dot{y}_d \cos(\phi_d)) \sin(\phi_e) \\
&= \dot{\phi} e_l - v + v_d \cos \phi_e
\end{aligned}$$

and

$$\begin{aligned}
\dot{e}_l &= -e_d \dot{\phi} - \sin(\phi) \dot{x}_d + \sin(\phi) \dot{x} + \cos(\phi) \dot{y}_d - \cos(\phi) \dot{y} \\
&= -e_d \dot{\phi} + \dot{x} \sin(\phi) - \dot{y} \cos(\phi) - \dot{x}_d \sin(\phi) + \dot{y}_d \cos(\phi) \\
&= -e_d \dot{\phi} + v \cos(\phi) \sin(\phi) - v \cos(\phi) \sin(\phi) - \dot{x}_d \sin(\phi_d - \phi_e) + \dot{y}_d \cos(\phi_d - \phi_e) \\
&= -e_d \dot{\phi} - \dot{x}_d (\sin(\phi_d) \cos(\phi_e) - \cos(\phi_d) \sin(\phi_e)) + \dot{y}_d (\cos(\phi_d) \cos(\phi_e) + \sin(\phi_d) \sin(\phi_e)) \\
&= -e_d \dot{\phi} + (\dot{x}_d \cos(\phi_d) + \dot{y}_d \sin(\phi_d)) \sin(\phi_e) + (\dot{y}_d \cos(\phi_d) - \dot{x}_d \sin(\phi_d)) \cos(\phi_e) \\
&= -e_d \dot{\phi} + v \sin(\phi_e).
\end{aligned} \tag{67}$$

The navigational control law is developed through a Lyapunov argument. The candidate Lyapunov function is chosen as

$$V = \frac{1}{2} e_d^2 + \frac{1}{2} e_l^2 + \frac{1 - \cos(\phi_e)}{k^2}. \tag{68}$$

This can be observed to be non-negative in the errors and V describes a positive definite system. The time derivative of the Lyapunov function can be found in conjunction with Equation (67) to be

$$\begin{aligned}
\dot{V} &= e_d \dot{e}_d + e_l \dot{e}_l + \frac{\sin(\phi_e) \dot{\phi}_e}{k^2} \\
&= e_d \left(\dot{\phi} e_l - v + v_d \cos(\phi_e) \right) + e_l \left(-e_d \dot{\phi} + v_d \sin(\phi_e) \right) + \frac{\sin(\phi_e) \left(\dot{\phi}_d - \dot{\phi} \right)}{k^2}.
\end{aligned} \tag{69}$$

Kanayama (1990) proposes the control law

$$\begin{aligned}
v &= v_d \cos(\phi_e) + k_1 e_d \\
&\text{and} \\
\dot{\phi} &= \dot{\phi}_d + v_d (k_2 e_l + k_3 \sin(\phi_e))
\end{aligned} \tag{70}$$

where $k1, k2,$ and $k3$ are positive constants used as the control terms. This control yields the Lyapunov time derivative as

$$\dot{V} = -k1 e_d^2 - \frac{k3}{k2} v_d \sin(\phi_e)^2. \quad (71)$$

The controller proposed by Kanayama in Equation (70) is robust and asymptotically tracks the desired path when the desired velocity remains positive. The control gains must all be positive. However, it can be observed that the angular velocity in Equation (70) does not contain e_d while the translational velocity does not contain e_l .

Kanayama did not implement the navigational control in conjunction with inner control loop. When using the navigational control to compensate desired generalized velocities used by the inner control loop simulations have indicated that including e_d in the navigational control term for $\dot{\phi}$ and e_l in the navigational control term for v may yield more desirable results.

The navigational control proposed for use in this thesis is

$$\begin{aligned} v &= v_d \cos(\phi_e) + k1 e_d + k4 e_d e_l^2 \\ \text{and} \\ \dot{\phi} &= \dot{\phi}_d + v_d (k2 e_l + k3 \sin(\phi_e)) + k5 \sin(\phi_e) e_l^2. \end{aligned} \quad (72)$$

The time derivative of the Lyapunov function is then given by

$$\dot{V} = -k1 e_d^2 - \frac{k3}{k2} v_d \sin(\phi_e)^2 - k4 e_d^2 e_l^2 - \frac{k5 \sin(\phi_e) e_l^2}{k2}. \quad (73)$$

Clearly $V \geq 0, \dot{V} \leq 0$. Furthermore to prove stability, if Equation (72) is substituted into Equation (66) and linearized about $P = 0$, then

$$\dot{P} = A P = \begin{bmatrix} -k1 & \dot{\phi}_d & 0 \\ -\dot{\phi}_d & 0 & v_d \\ 0 & -v_d k2 & -v_d k3 \end{bmatrix} P. \quad (74)$$

The characteristic polynomial of the matrix in Equation (74) is then described by

$$|S I - A| = a_3 s^3 + a_2 s^2 + a_1 s + a_0 \quad (75)$$

where $a_3 = 1, a_2 = v_d k3 + v_d k1, a_1 = v_d^2 k2 + v_d^2 k1 k3 + \dot{\phi}_d^2$, and $a_0 = v_d^3 k1 k2 + \dot{\phi}_d^2 v_d k3$.

All a_i are positive and $a_1 a_2 - a_0 a_3 > 0$. Therefore the real parts of all roots are negative through the Routh-Hurwitz Criterion and the linearized system is stable for small errors.

Navigational Control Recap

When the actual dynamic response is described by either Equation (70) or Equation (72), the path converges asymptotically to the desired trajectory. However, this is not necessarily the case when the navigational control is used to calculate desired velocities used by the inner control loop that compensate for X and Y coordinate errors. Adequate gains must be chosen to accurately and precisely track the path as will be observed later in this chapter.

The modified navigational control law presented in Equation (72) can yield faster and more precise convergence than the former navigational control proposed by Kanayama (1990) and presented in Equation (70). However in practice it may be best to start with the original control from Equation (70) and then add the additional components. As an ad-hoc thought, this is analogous to how a PID controller often yields more desirable tracking results, though a simpler proportional controller may more easily be chosen to make the system stable. Techniques to choose navigational gains that yield the most precise results when used in conjunction with an inner control loop is an area of interest for future research.

Rolling Wheel

The rolling wheel configuration is shown in Figure 16. In the set-up, both the θ direction (wheel tilt) and ψ direction (wheel displacement angle) are actuated while ϕ (heading direction) is left unactuated. A similar system was first used by Xu and Au (2004). Their approach used a linearized model to calculate two control components. The velocity control component was designed to ensure path curvature continuity. The second control ensured that the robot followed a lean angle trajectory. The lean angle trajectory was calculated through inverse dynamics.

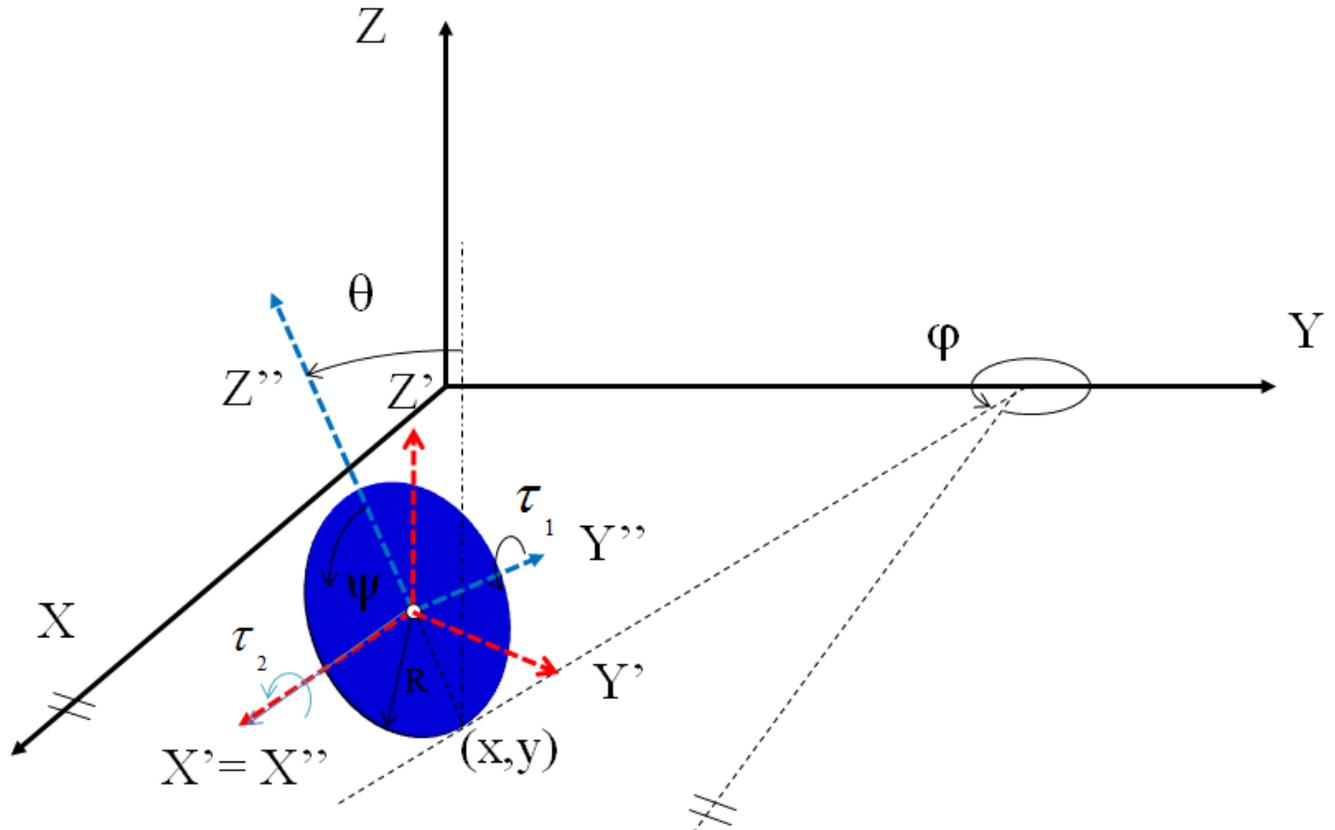


Figure 16: Rolling Wheel Free Body Diagram

Previous Work

An underactuated sliding mode control was also implemented on the system by Patenaude (2008). In his work, inverse dynamics were used to determine all the desired coordinate histories. When the system was modeled perfectly, and with ideal initial conditions, the controller tracks the desired trajectories with little error. It was also shown that a small disturbance on feedback could be overcome. However a small disturbance on feed forward control resulted in large trajectory errors.

The reason for this behavior was that the X and Y coordinates were removed from the dynamic equations and were not used in the control. The desired X and Y trajectory was converted to a ψ and ϕ trajectory through a non-holonomic constraint. Because the constraint is not integrable, the controller would track the ψ and ϕ trajectories, but not the X and Y trajectories. Figure 17 shows the results from Patenaude (2008) when the rolling wheel started without ideal initial conditions. The wheel would continue to loop on the actual X and Y path in Figure 17 and

never converge to the desired X and Y path. Furthermore, the control required integration \dot{K}_d to calculate the control law. The matrix K_d is nearly constant and the advantages that it provides can be implemented by the sliding mode tracking controller in this thesis without the additional calculations to integrate \dot{K}_d .

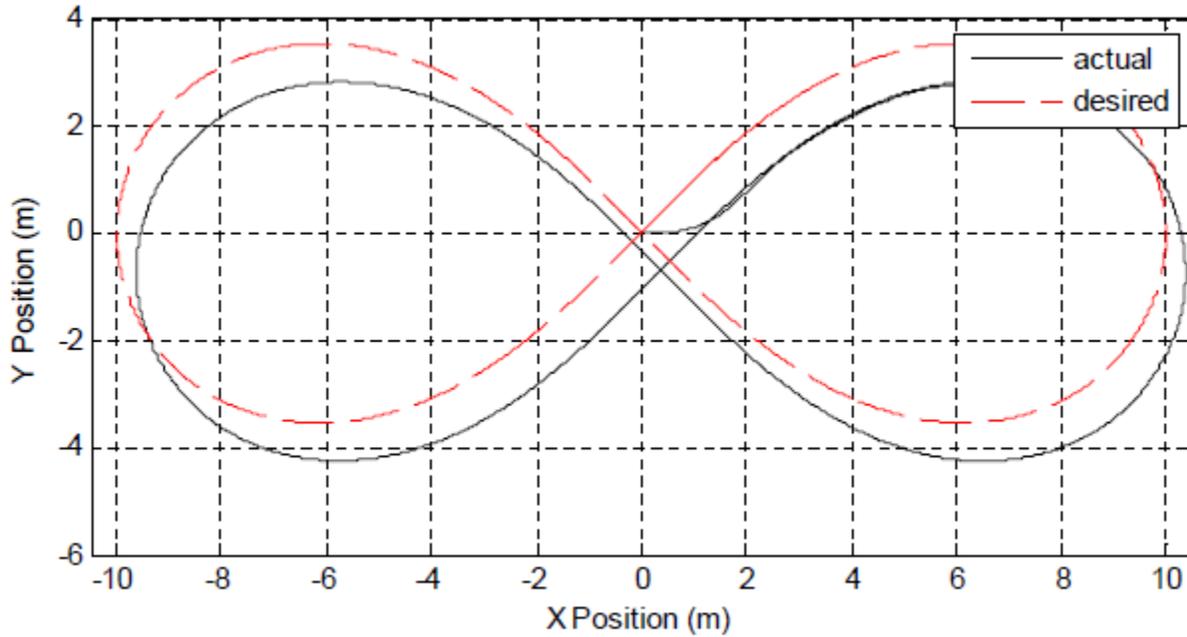


Figure 17: Rolling Wheel Path Without Navigational Control

The goal for the currently proposed control is to provide input torques that will roll the wheel in a figure-8 shape in the X and Y plane and overcome disturbances. The control does not require integration of \dot{K}_d and will converge to the desired X and Y coordinates.

Rolling Wheel Dynamics

The rolling wheel dynamics were found using the non-holonomic form of Lagrange's Equation as presented in Equation (3). This was carried out in the symbolic math software Maple 16. The derivation is presented in Appendix A. The dynamic system is described by

$$\begin{aligned}
M &= \begin{bmatrix} \frac{3}{2}mR^2 & 0 & \frac{3}{2}mR^2 \sin(\theta) \\ 0 & \frac{5}{2}mR^2 & 0 \\ \frac{3}{2}mR^2 \sin(\theta) & 0 & \frac{3}{2}mR^2 - \frac{5}{4}mR^2 \cos(\theta)^2 \end{bmatrix}, \\
C &= \begin{bmatrix} 0 & \frac{5}{2}mR^2 \cos(\theta)\dot{\phi} & 0 \\ 0 & 0 & -\frac{3}{2}mR^2 \cos(\theta)\dot{\psi} - \frac{5}{4}mR^2 \cos(\theta) \sin(\theta)\dot{\phi} \\ 0 & \frac{1}{2}mR^2 \cos(\theta) (\dot{\psi} + 5 \sin(\theta)\dot{\phi}) & 0 \end{bmatrix}, \\
G &= \begin{bmatrix} 0 \\ -mgR \sin(\theta) \\ 0 \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ 0 \end{bmatrix}, \text{ and } q = \begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix}.
\end{aligned} \tag{76}$$

The rolling wheel has the non-holonomic constraints

$$\dot{X} = R \cos(\phi)\dot{\psi} \text{ and } \dot{Y} = R \sin(\phi)\dot{\psi}. \tag{77}$$

The non-holonomic constraints given in Equation (77) are already integrated into the matrices given by Equation (76) to minimize $n-m$.

Rolling Wheel Controllability

If the system dynamics are linearized about zero for all generalized coordinate positions and velocities, then the linearized state matrices have the form

$$\begin{aligned}
\dot{x} &= \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = Ax + Bu \\
\dot{x} &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ c2 & 0 \\ 0 & c3 \\ c4 & 0 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}.
\end{aligned} \tag{78}$$

In Equation (78) c_i are constants related to the physical quantities of the system. The controllability matrix of this linearized system has a rank of only 4. If all generalized coordinates are linearized about zero, except $\dot{\psi}$ is linearized about a constant angular velocity then the linearized state space system is represented by

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c1 & 0 & 0 & 0 & c5 \\ 0 & 0 & 0 & 0 & c6 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ c2 & 0 \\ 0 & c3 \\ c4 & 0 \end{bmatrix} \begin{bmatrix} \tau 1 \\ \tau 2 \end{bmatrix}. \quad (79)$$

Now the constants depend upon the physical quantities of the system as well as the constant speed that ψ is linearized about. The controllability matrix for this linearized system increases in rank to 5, but still does not have full rank. Though linear control methods are uncontrollable about this point, it does not necessarily indicate that non-linear control will be uncontrollable about this same point. If it can be proved that a non-linear system is controllable about a linearized point in the dynamic system, the control is said to be small-time locally controllable at the linearized point Slotine & Li (1991).

A non-linear system is in control affine if

$$\dot{x} = f(x) + \sum_{i=1}^m g_i(x)u_i. \quad (80)$$

where x_i is the i^{th} generalized coordinate and u_i is the i^{th} input. g is a function of the states and the inputs are multiplied by it. The system is said to be locally accessible about x_0 if the accessibility distribution C spans n space, Hedrick (2005). C is defined by

$$C = [g_1, g_2, \dots, g_m, [g_i, g_j], \dots, [ad_{g_i}^k, g_j], \dots, [f, g_i], \dots, [ad_f^k, g_i], \dots] \quad (81)$$

and is analogous to the controllability matrix of a linear system.

The Lie bracket is defined as

$$[f, g] = \frac{dg}{dx}f - \frac{df}{dx}g. \quad (82)$$

The adjoint is defined as

$$\begin{aligned} [ad_f^1, g] &= [f, g], \\ [ad_f^2, g] &= [f, [f, g]], \\ &\dots \\ [ad_f^k, g] &= [f, ad_f^{k-1}, g]. \end{aligned} \quad (83)$$

When the system is linearized about all positions and velocities set to zero, except $\dot{\psi}$ is set to a positive constant, the control affine form of the rolling wheel is given by

$$\ddot{x} = \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \\ 0 \\ \frac{6}{5}\dot{\psi}_{con}\dot{\phi} + \frac{4}{5}\frac{g\theta}{R} \\ -2\dot{\theta}\dot{\psi}_{con} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{2}{3} \\ 0 \\ -4\theta \end{bmatrix} \tau_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{4}{5} \\ 0 \end{bmatrix} \tau_2. \quad (84)$$

In Equation (84), R is the wheel radius from Figure 16, g is the gravitational constant, and $\dot{\psi}_{con}$ is the constant angular speed that the system is linearized about.

An accessibility distribution is described by

$$C = [g_1, g_2, [f, g_1], [f, g_2], [f, [f, g_2]], [g_1, [f, g_2]]] \\ = \begin{bmatrix} 0 & 0 & -\frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{4}{5} & 0 & 0 \\ 0 & 0 & 4\theta & 0 & -\frac{8}{5}\dot{\psi}_{con} & 0 \\ \frac{2}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{24}{5}\theta\dot{\psi}_{con} & 0 & -\frac{16}{25}\frac{-g+3\dot{\psi}_{con}^2 R}{R} & 0 \\ -4\theta & 0 & -4\dot{\theta} & \frac{8}{5}\dot{\psi}_{con} & 0 & -\frac{16}{5} \end{bmatrix}, \quad (85)$$

and shows full rank. Note that the rank is full regardless of values for $(\theta, \dot{\theta}, \dot{\phi})$, so long as $\dot{\psi}_{con} \neq 0$. Thus the non-linear system is small-time locally controllable about the linearized point.

Sliding Mode Control Parameters

Precise tracking will be implemented on the ψ and ϕ coordinates, while θ simply needs to be stabilized. The first step for finding the underactuated sliding mode control is to start with the control law given by Equation (37),

$$\begin{aligned} \tau &= M\ddot{q}_r + C\dot{q}_r + G - K_d s \\ &= M(\ddot{q}_d - \Lambda(\dot{q} - \dot{q}_d)) + C(\dot{q}_d - \Lambda(q - q_d)) + G - K_d((\dot{q} - \dot{q}_d) + \Lambda(q - q_d)) \end{aligned}$$

where M , C , and G are the matrices that describe the dynamic system in Equation (76).

For this dynamic model the third dynamic equation has no applied torque, therefore the third row of the control law needs to cancel out without a torque applied to this row. Because $\ddot{\theta}_d$

does not show up in the third line of the control law, the third line is solved for $\dot{\theta}_d$ to ensure that no torque is applied on the third dynamic equation. This process provides

$$\begin{aligned} \dot{\theta}_d &= \frac{Num}{Den}, \text{ where} \\ Num &= -\frac{3}{2} mR^2 \sin(\theta) \left(\ddot{\psi}_d - \lambda_1 (\dot{\psi} - \dot{\psi}_d) \right) - \left(\frac{3}{2} mR^2 (\sin(\theta))^2 + \frac{1}{4} mR^2 (\cos(\theta))^2 \right) \left(\ddot{\phi}_d - \lambda_3 (\dot{\phi} - \dot{\phi}_d) \right) + \frac{1}{4} \dot{\theta} mR^2 \cos(\theta) \left(\dot{\psi}_d - \lambda_1 (\dot{\psi} - \dot{\psi}_d) \right) \\ &\quad - \frac{5}{4} \dot{\theta} mR^2 \cos(\theta) \sin(\theta) \left(\dot{\phi}_d - \lambda_3 (\dot{\phi} - \dot{\phi}_d) \right) + K_d31 \left(\dot{\psi} - \dot{\psi}_d + \lambda_1 (\psi - \psi_d) \right) + K_d33 \left(\dot{\phi} - \dot{\phi}_d + \lambda_3 (\phi - \phi_d) \right) + \frac{3}{4} mR^2 \cos(\theta) \dot{\psi} \lambda_2 \theta \\ &\quad - \frac{3}{4} mR^2 \cos(\theta) \dot{\psi} \lambda_2 \theta_d + \frac{5}{4} \dot{\phi} \cos(\theta) mR^2 \sin(\theta) \lambda_2 \theta - \frac{5}{4} \dot{\phi} \cos(\theta) mR^2 \sin(\theta) \lambda_2 \theta_d + K_d32 \dot{\theta} + K_d32 \lambda_2 \theta - K_d32 \lambda_2 \theta_d \\ \text{and} \\ Den &= \frac{5}{4} \dot{\phi} \cos(\theta) mR^2 \sin(\theta) + K_d32 + \frac{3}{4} mR^2 \cos(\theta) \dot{\psi}. \end{aligned} \tag{86}$$

The solution process can be found in Appendix A. The system parameters and aggressive navigational control values used are given in Table 5. The figure-8 path to track was given by

$$X_d = r \sin(\omega t) \text{ and } Y_d = \pm \frac{1}{2} \sqrt{-4x^2 - 2r^2 + 2\sqrt{8r^2x^2 + r^4}}. \tag{87}$$

The correct sign to make the figure-8 continuous was found through a series of if statements that involved the period of the path X_d . One figure-8 cycle takes 2π seconds to complete. The code can be viewed in Appendix A. The remaining parameters are shown in Table 5.

Table 5: Rolling Wheel System Parameters

Variable	Description	Value Used
m	mass of the wheel	5.0 [kg]
R	wheel radius	0.5 [m]
g	gravitational constant	9.81 [m/s ²]
r	figure 8 width	10 [m]
ω	figure 8 frequency	0.5 [rad/s]

Because the previous control by Patenaude (2008) provides results to compare to, these are the same dynamic system parameters and desired path used in Chapter 4 from Patenaude (2008).

For the closed loop tracking problem refer to Figure 13. v from the navigational control can be converted to $\dot{\psi}_d$ by solving

$$v = R \dot{\psi}_d \tag{88}$$

for $\dot{\psi}_d$ where v is the compensating linear velocity calculated by the navigational control, R is the wheel radius, and $\dot{\psi}_d$ is the desired angular rotation of the wheel used by the inner control loop. The compensating angular velocity from the navigational control is equal to $\dot{\phi}_d$ used by the inner

control loop. For detailed views of the subsystems and user defined Matlab functions see Appendix A.

$\dot{\theta}_d$ from Equation (86), θ_d from integrating $\dot{\theta}_d$, the system dynamic matrices from Equation (76), and the system's dynamic properties from Table 5 were inserted into the non-linear control law given in Equation (37). The first step was to find suitable values for the elements of the matrices K_d and Λ .

Somewhat arbitrary values were initially picked for the elements of K_d and Λ . The first guess that successfully followed the path is shown in Table 6.

Table 6: Initial Sliding Mode Control Parameters

Control Variable	Value
Kd11	10
Kd12	-1
Kd13	-1
Kd21	-1
Kd22	10
Kd23	-1
Kd31	-1
Kd32	-1
Kd33	10
λ_1	1
λ_2	10
λ_3	1

The diagonal values for K_d were all initially chosen to be 10, while the non-diagonal terms were all initially set to be -1. This makes the matrix K_d positive definite. λ_2 was chosen to be an order of magnitude larger than λ_1 and λ_3 to place an emphasis on keeping the θ coordinate stabilized. All initial conditions were set to match the ideal initial condition with the following exceptions: $X(0) = -2$, $Y(0) = \frac{4}{5}$, $\phi(0) = \frac{\pi}{3}$.

Because the first step is only intended to find control law values for K_d and Λ , open loop tracking of the X and Y coordinates will be implemented first. Refer to Figure 14 for the open loop tracking schematic. The actual Simulink file used is included in Appendix A. For best case results, the rolling wheel will make the correct shape of the figure-8 path, however the figure-8 contact position may be translated and rotated in the global XY plane. The dynamic response for two laps around the figure-8 path is shown in Figure 18.

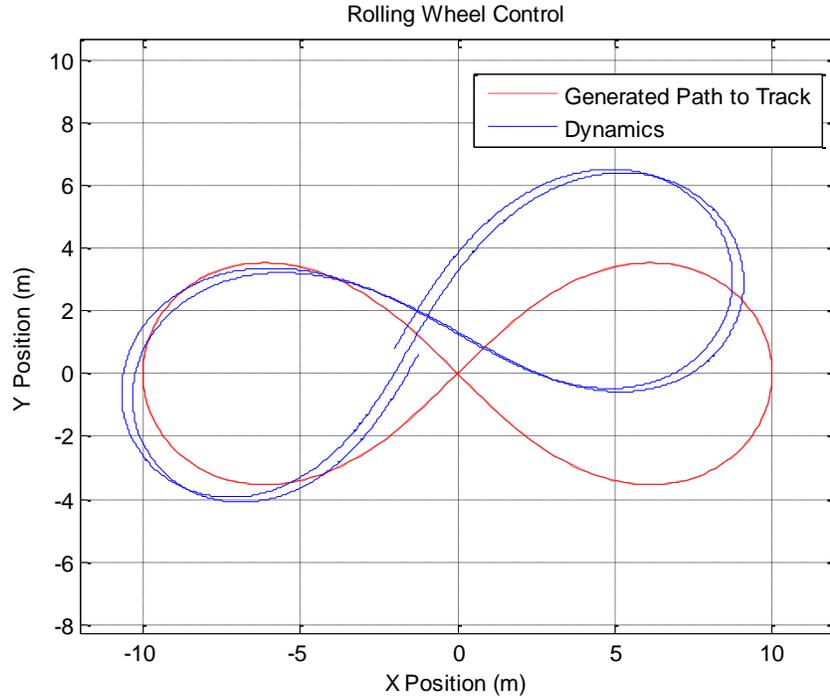


Figure 18: Open-Loop XY Tracking

From this plot it can be inferred the coordinates (ψ, θ, ϕ) track well. The figure-8 has the nearly the correct shape and magnitude. However, it is not aligned with the desired figure-8. As expected the figure-8 starts at an angle because the initial angle of the wheel was set to $\frac{\pi}{3}$. These initial guesses for K_d and Λ were simply chosen to make K_d positive definite and place an emphasis on keeping θ stabilized.

Navigational Control Parameters

The next step is to find gains for the navigational control. The first step is to set the newly proposed navigational control terms ($k4$ and $k5$ from Equation (72)) equal to zero. Following the advice of Kanayama et al. (1990), $k3$ is set to $2\sqrt{k2}$. The navigational control is initially not used in conjunction with the proposed sliding mode control or the dynamics of the system to find reasonable values for $k1$ and $k2$. The outputs from the navigational control are used as the actual translational and rotational velocities. Initial conditions are set to match the desired rotational and translational velocities. The initial position is set to $X(0) = -2, Y(0) = \frac{4}{5}, \phi(0) = \frac{\pi}{3}$. The following trajectories were obtained.

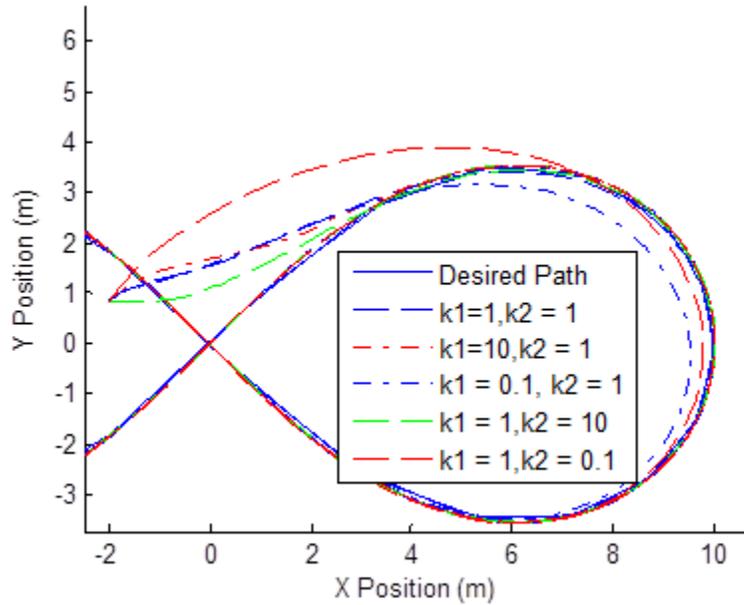


Figure 19: Navigational Control Gains Test

Figure 19 shows that a large value for $k1$ places an emphasis on the translational velocity, while $k2$ places an emphasis on the heading direction. Intuitively it is known that the time derivative of the heading direction for the rolling wheel should be small in magnitude. Figure 19 suggests that when blending the navigational control with the sliding mode control for the rolling wheel, a small value for $k2$ should be expected.

Blending SMC and Navigational Control Values

The control law values from Table 6 for the sliding mode control was tested with various values for the navigational control. The dynamic response for the first guess of the navigational control gains is shown in Figure 20.

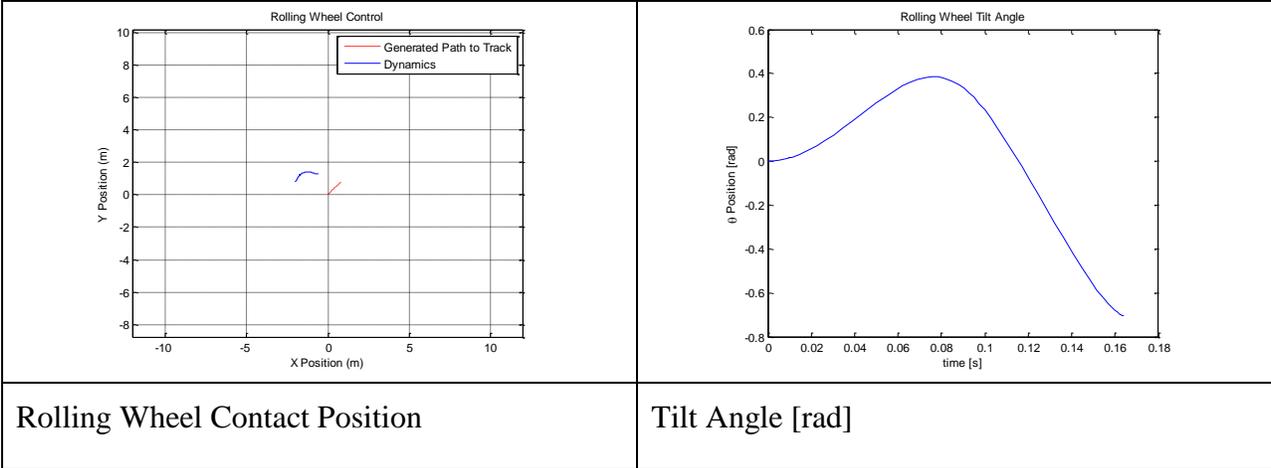


Figure 20: Dynamic Response for $k1=1$ $k2=1$

The first response for the navigational control gains $k1=1$ and $k2=1$ did not work very well. The rate at which the wheel changed its heading direction was too fast, which in turn makes the angle at which the wheel needs to lean to steer the rolling wheel along the trajectory too extreme. It was decided to use $k1=1$ and $k2=0.1$ by inspecting Figure 19 and knowing that the rate at which the wheel changes heading direction needs to decrease.

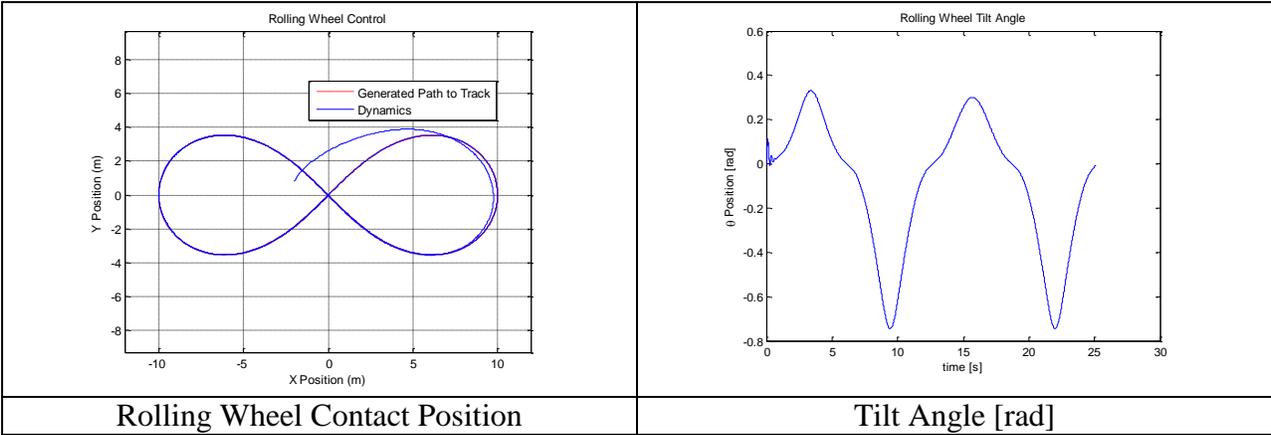


Figure 21: Dynamic Response for $k1=1$ $k2=0.1$

The simulation results in Figure 21 show two cycles of the figure-8 path. It shows that the actual contact position does in fact converge with the desired contact position. It is interesting that the wheel lean angle is not symmetric for the left and right half of the figure-8. It actually leans further when the rolling wheel is on the left hand side of the figure-8 path. Consider a free body diagram that represents a simplified version of the rolling wheel's lean angle as shown in Figure 22.

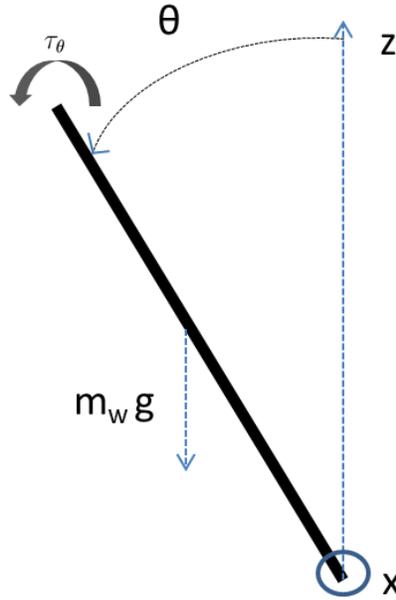


Figure 22: Rolling Wheel Torque About X-Axis

In Figure 22 the force applied at the wheel's contact position is neglected. The sum of the torques about the local x-axis is then given by

$$\sum \tau_x = \tau_\theta + m_w g r \sin(\theta). \quad (89)$$

A plot of τ_θ , the gravitational torque, and the sum of these torques is shown in Figure 23 for $k1=1$ and $k2=0.1$. Figure 23 shows that the gravitational torque and τ_θ are both greater on one side of the figure-8 than the other. However, the sum of the two torques is more symmetric. This is because for an underactuated system there is more than one path that the θ coordinate can take to track the desired X and Y path. However, τ_θ applied will also be different. The sum of the two torques is not exactly the same for the left and right sides of the figure-8 path because the frictional force from the ground on the wheel to maintain the non-holonomic constraint and $\dot{\psi}$ were not considered for this simple study. However, this does help provide some insight as to why angles are greater on one side of the figure-8 path.

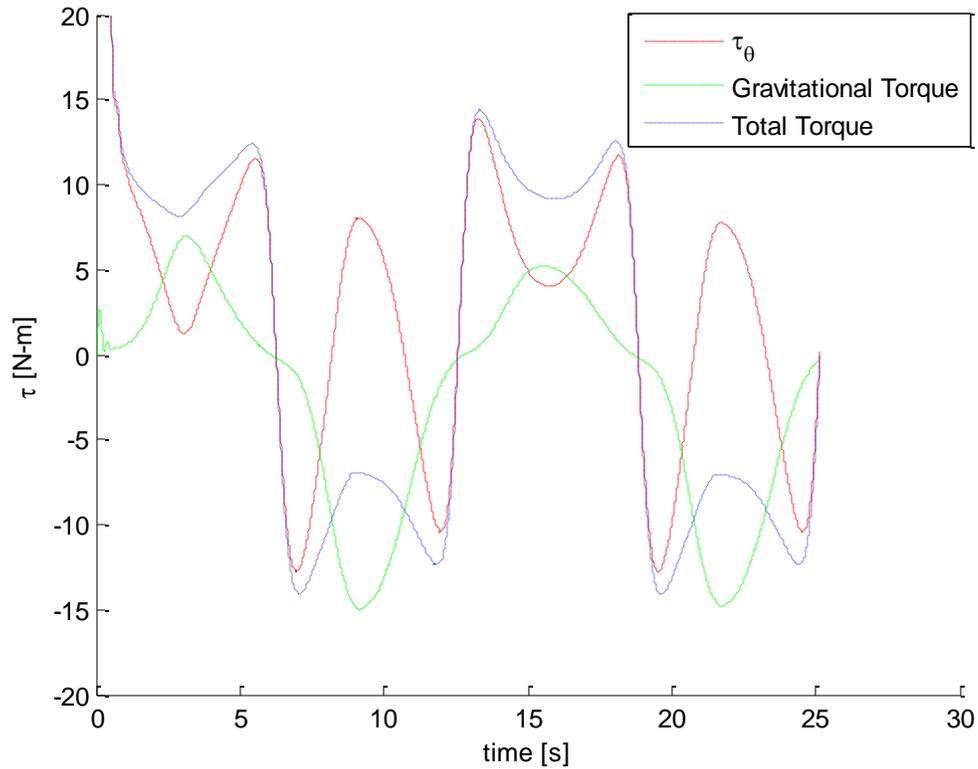


Figure 23: Rolling Wheel Torques

In implementation it may be desirable to limit the torque that is applied by the motors. The gains in the navigational control were set aggressively. Simply changing the gains in the navigational control can yield a path with a more passive convergence rate, thus reducing the applied torques. In order to provide a more passive convergence to the desired trajectory the navigational control values were changed to

$$\left(k_1 = 1, k_2 = 0.01, k_3 = 2\sqrt{k_2}\right) \quad (90)$$

which should result in a more passive convergence to the desired XY plane trajectory. Simulations for the passive navigational control values are shown in Figure 24, Figure 25, and Figure 26.

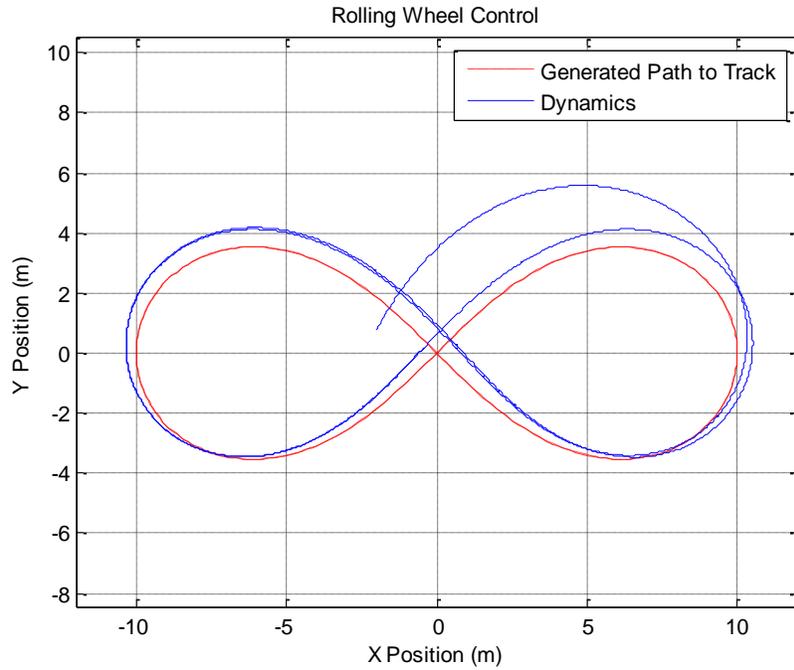


Figure 24: Rolling Wheel Passive Navigational Control Contact Position

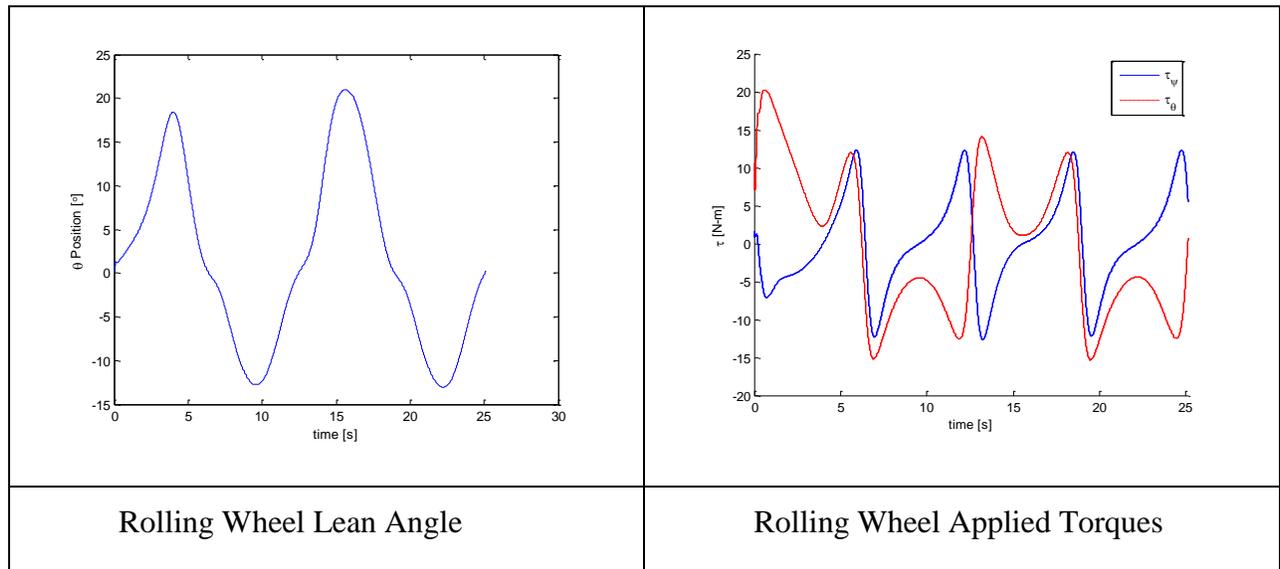


Figure 25: Rolling Wheel Passive Navigational Gains Response

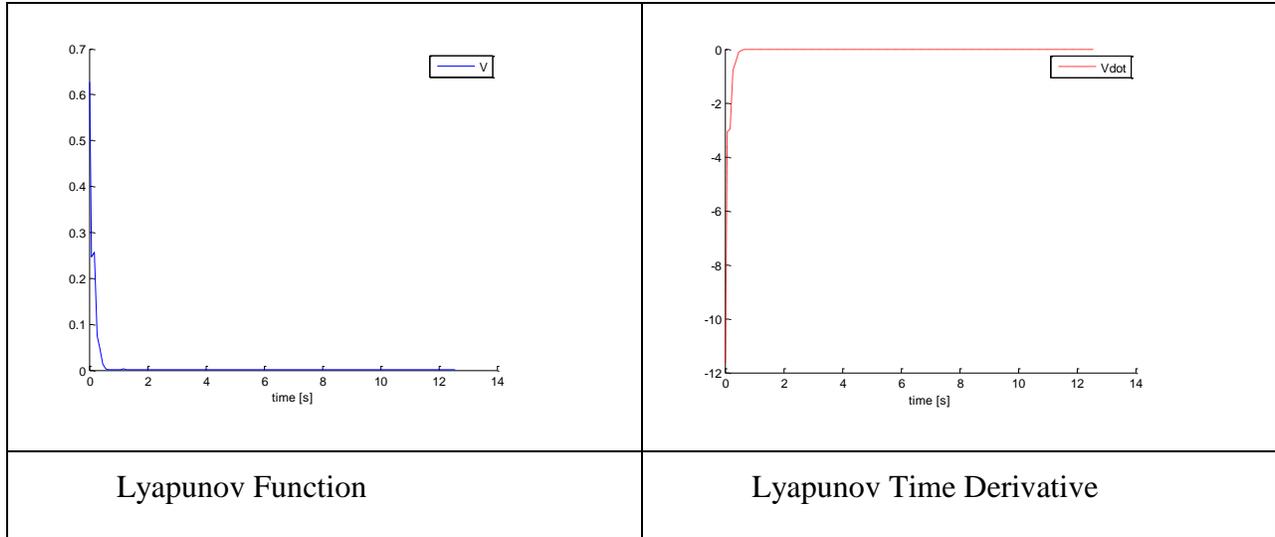


Figure 26: Rolling Wheel Passive Navigational Control Lyapunov Function Plots

These passive gains yield a response with less chatter when the system first begins tracking. Responses are very similar between the aggressive and passive control after about three seconds. This is because the navigational control yields similar results to the open loop response when the actual position is very close to the desired position.

It can be observed from Figure 26 that there are small periods of time where the Lyapunov Function does not decrease. The likely reason for this is that $\dot{\theta}_d$ is found in Equation (86) to ensure that the control is underactuated. $\ddot{\theta}_d$ is not calculated by taking the derivative of $\dot{\theta}_d$. $\ddot{\theta}_d$ is set to 0, as this is the region the tilt of the wheel should stay in. This is an area of interest for future work.

Non-Holonomic Segbot

This section first derives the dynamic equations of motion that describe the motion of the Segbot in the global XY plane. For this configuration two non-holonomic constraints must be imposed. The Segbot was modeled as 3 rigid bodies: these being the left wheel, right wheel, and the pendulum body. ψ_l and ψ_r as well as their time derivatives, are not necessarily equal to each other for the non-holonomic case.

Equations of Motion Using Lagrange

To simplify the derivation of the equations of motion for the non-holonomic case of the Segbot, the projection of the axel center onto the XY plane needs to be related to the rotation of the Segbot wheels. The axel center translational and angular velocities can be related to the wheel velocities through the constraints

$$v = \frac{R(\dot{\psi}_r + \dot{\psi}_l)}{2}, \text{ and } \dot{\phi} = \frac{R}{w}(\dot{\psi}_r - \dot{\psi}_l) \quad (91)$$

where w is the distance between the center of each wheel, v is the axel center translational speed, and $\dot{\phi}$ is the angular velocity at which the Segbot changes heading direction. Refer to Figure 8.

It is assumed that both wheels are the same distance from the axel center. Notice how ψ_l and ψ_r are defined in Figure 8. A positive angular rotation corresponds to the right hand rule and the body centered y -axis. ψ_r falls on the negative side of the y coordinate and ψ_l falls on the positive side. The wheels are both assumed to maintain contact with the ground. They are also modeled as thin circular disks, with identical mass. ϕ is an angle that describes the heading direction of the Segbot. Notice that if the second part of Equation (91) is written in terms of position instead of velocity, then it becomes a holonomic constraint. This simplification is applied later in the derivation.

The constraints from Equation (91) could also be written in terms of wheel velocity with respect to the axel center speed or

$$\dot{\psi}_r R = v + \frac{w}{2}\dot{\phi}, \text{ and } \dot{\psi}_l R = v - \frac{w}{2}\dot{\phi}. \quad (92)$$

The translational velocity of the axel center in the global XY plane is given by the non-holonomic constraints

$$\dot{X} = \frac{R(\dot{\psi}_r + \dot{\psi}_l)}{2} \cos(\phi) \text{ and } \dot{Y} = \frac{R(\dot{\psi}_r + \dot{\psi}_l)}{2} \sin(\phi). \quad (93)$$

The dynamic system is described by the six generalized coordinates $(X, Y, \psi_l, \psi_r, \theta, \phi)$. The dynamic system has only two actuators to work with. To simplify the inner loop control it is desirable to reduce $n-m$ as low as possible.

ϕ can be eliminated easily since it shows up as a holonomic constraint. This leaves the Lagrangian in terms of 5 generalized coordinates, $(X, Y, \psi_r, \psi_l, \theta)$. The X and Y coordinates cannot be eliminated before applying Lagrange's Equation with Lagrange multipliers given by Equation (3) due to the non-holonomic constraints. Two Lagrange multipliers must be added to the equations of motions. Then the two non-holonomic constraint equations can be differentiated to give a total of seven equations to describe the dynamics.

The two constraint equations can be solved for \ddot{X} and \ddot{Y} . The Lagrange equations found with respect to X and Y can be solved for the Lagrange multipliers. This reduces the system to three second order differential equations to describe the Segbot's motion. The derivation is completed in detail in Appendix B. The equations of motion are described by

$$\begin{aligned}
 M = & \begin{bmatrix} \frac{1}{2} \frac{R^2 \left((2I3 - 2m_p h_c^2 - 2II) \cos(\theta) \right)^2 + \left(\frac{1}{2} W^2 + 2h_c^2 \right) m_p + 2II + mv R^2 + 3mv W^2}{W^2} & -\frac{1}{2} \frac{\left((2I3 - 2m_p h_c^2 - 2II) \cos(\theta) \right)^2 + \left(2h_c^2 - \frac{1}{2} W^2 \right) m_p + 2II + mv R^2}{W^2} R^2 & \frac{1}{2} R m_p h_c \cos(\theta) \\ -\frac{1}{2} \frac{\left((2I3 - 2m_p h_c^2 - 2II) \cos(\theta) \right)^2 + \left(2h_c^2 - \frac{1}{2} W^2 \right) m_p + 2II + mv R^2}{W^2} R^2 & \frac{1}{2} \frac{R^2 \left((2I3 - 2m_p h_c^2 - 2II) \cos(\theta) \right)^2 + \left(\frac{1}{2} W^2 + 2h_c^2 \right) m_p + 2II + mv R^2 + 3mv W^2}{W^2} & \frac{1}{2} R m_p h_c \cos(\theta) \\ \frac{1}{2} R m_p h_c \cos(\theta) & \frac{1}{2} R m_p h_c \cos(\theta) & II + m_p h_c^2 \end{bmatrix}, \\
 C = & \begin{bmatrix} \frac{Dtheta R^2 \sin(\theta) \cos(\theta) (m_p h_c^2 + II - I3)}{W^2} & \frac{\sin(\theta) R^2 h_c m_p (-Dpsir + Dpsil)}{W^2} & \frac{2 \sin(\theta) \left(R \left(-\frac{1}{2} Dpsir + Dpsil \right) (m_p h_c^2 + II - I3) \cos(\theta) + \frac{1}{4} Dtheta m_p h_c W^2 \right) R}{W^2} \\ \frac{\sin(\theta) R^2 \left(-2 Dtheta (m_p h_c^2 + II - I3) \cos(\theta) + R m_p h_c (-Dpsir + Dpsil) \right)}{W^2} & \frac{Dtheta R^2 \sin(\theta) \cos(\theta) (m_p h_c^2 + II - I3)}{W^2} & \frac{R \sin(\theta) \left(R Dpsil (m_p h_c^2 + II - I3) \cos(\theta) - \frac{1}{2} Dtheta m_p h_c W^2 \right)}{W^2} \\ \frac{R^2 \sin(\theta) \cos(\theta) (m_p h_c^2 + II - I3) (-Dpsir + 2 Dpsil)}{W^2} & \frac{R^2 Dpsil (m_p h_c^2 + II - I3) \cos(\theta) \sin(\theta)}{W^2} & 0 \end{bmatrix}, \\
 G = & \begin{bmatrix} 0 \\ 0 \\ -\dot{h}_c \sin(\theta) m_p g \end{bmatrix}, \text{ and } \begin{bmatrix} \tau_r \\ \tau_l \\ -\tau_r - \tau_l \end{bmatrix}.
 \end{aligned} \tag{94}$$

The individual parameters are further described in Table 7.

Control Derivation and Implementation

The system's dynamic properties from Equation (94) are then inserted into the control law given by Equation (37). The sum of the first row, the second row, and the third row is then solved for $\ddot{\theta}_d$ to yield

$$\frac{d^2}{dt^2} \theta_d(t) = \frac{1}{2} \frac{1}{W^2 (R m_p h_c \cos(\theta) + N (I_2 + m_p h_c^2))} \left(2 R \left(R (m_p h_c^2 + I_1 - I_3) \left((-2 Dpsir_d + Dpsil_d) Dpsil + Dpsir_d Dpsir \right) N - (\theta - \theta_d) \left(-Dpsir + Dpsil \right) \lambda_3 + Dpsil Dtheta_d - Dpsir Dtheta_d - Dtheta (Dpsil_d - Dpsir_d) \sin(\theta) + \frac{1}{2} h_c W^2 \left((\lambda_2 Dpsil - \lambda_2 Dpsil_d - DDpsir_d - DDpsil_d - \lambda_1 Dpsir_d + \lambda_1 Dpsir) N + 2 \lambda_3 (Dtheta - Dtheta_d) m_p \right) \cos(\theta) + 2 \left((Dpsil_d - Dpsir_d) (-Dpsir + Dpsil) R^3 - Dtheta (\lambda_3 (\theta - \theta_d) - Dtheta_d) W^2 R + g W^2 N \right) h_c m_p \sin(\theta) + 2 W^2 \left(\frac{1}{2} (3 m v + m_p) (\lambda_2 Dpsil - \lambda_2 Dpsil_d - DDpsir_d - DDpsil_d - \lambda_1 Dpsir_d + \lambda_1 Dpsir) R^2 + N h_c^2 \lambda_3 (Dtheta - Dtheta_d) m_p + ((I_2 Dtheta - I_2 Dtheta_d + Km33 (\theta - \theta_d)) \lambda_3 + Km31 Dpsir - Km31 Dpsir_d + Km32 Dpsil - Km32 Dpsil_d + Km33 Dtheta - Km33 Dtheta_d) N + (\theta - \theta_d) (Km23 + Km13) \lambda_3 + (Km22 + Km12) Dpsil + (Km21 + Km11) Dpsir + (-Km21 - Km11) Dpsir_d + (Km23 + Km13) Dtheta + (-Km22 - Km12) Dpsil_d - Dtheta_d (Km23 + Km13) \right) \right) \quad (95)$$

This constraint is imposed so that the system is underactuated. The next step to complete the non-linear control law is given by substituting Equations (94) and (95) into Equation (37). Then the elements of Λ and K_d need to be chosen from Equation (37).

The dynamics given by Equation (94) are linearized about $(\theta = 0, \dot{\theta} = 0, \dot{\psi}_r = \dot{\psi}_l)$. The system parameters given in Table 7 are substituted. The system is then put in state-space form to yield

$$\dot{q} = \begin{bmatrix} \dot{\psi}_r \\ \dot{\psi}_l \\ \dot{\theta} \\ \ddot{\psi}_r \\ \ddot{\psi}_l \\ \ddot{\theta} \end{bmatrix} = Aq + Bu \quad (96)$$

$$= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -70.77 & 0 & 0 & 0 \\ 0 & 0 & -70.77 & 0 & 0 & 0 \\ 0 & 0 & 83.88 & 0 & 0 & 0 \end{bmatrix} q + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 170.51 & 30.95 \\ 30.95 & 170.51 \\ -53.79 & -53.79 \end{bmatrix} \begin{bmatrix} \tau_r & 0 \\ 0 & \tau_l \end{bmatrix}.$$

The linearized A and B matrices are used with the weighting matrices

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix} \text{ and } R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (97)$$

and the Matlab command $K = -lqr(A, B, Q, R)$ is used to find the state feedback gain matrix

$$K = \begin{bmatrix} 0 & 1 & 32.92 & 0.16 & 1.16 & 5.04 \\ 1 & 0 & 32.92 & 1.16 & 0.16 & 5.04 \end{bmatrix}. \quad (98)$$

The negative sign from the Matlab commad *lqr* is used so that the control is calculated by $K(q - q_d)$.

ψ_{rd} is set equal to ψ_r and ψ_{ld} is set equal to ψ_l for the sliding mode control law. Tracking of ψ_r and ψ_l is not important for the given XY path. The position XY is important. This essentially removes the four left-most feedback gains from Equation (98). These gains are accounted for in the navigational control. The navigational control essentially compensates for the removal of these positional gains for ψ_r and ψ_l by calculating a new $\dot{\psi}_{dr}$ and $\dot{\psi}_{dl}$ that ensure the dynamic system follows the desired XY path.

This leaves the remaining eight linear gains in Equation (98). There are 12 control constants given by Λ and K_d from Equation (37). So there are 12 control values to solve for and only 8 remaining unknown linear control gains to account for.

The elements of K_d and Λ are chosen so that when the dynamics from Equation (94) and the desired acceleration from Equation (95) are inserted into the nonlinear control given by Equation (37) and the resulting control is linearized about $(\theta = 0, \dot{\theta} = 0, \dot{\psi}_r = \dot{\psi}_l)$, the eight right most feedback gains match those in Equation (98).

Again a visual representation of the inner and outer control loop interaction is provided in Figure 13. A diagram of the actual Simulink file used for the simulations and the derivation in Maple for the control values used for K_d and Λ is provided in Appendix B. The dynamic system and inner loop control parameters used are presented in Table 7 and Table 8.

Table 7: Non-Holonomic Segbot Properties

Variable	Description	Value Used
m_w	mass of each wheel	0.25 [kg]
m_p	mass of the pendulum	1.0 [kg]
R	wheel radius	0.1016 [m]
g	gravitational constant	9.81 [m/s ²]
h_c	distance to pendulum center of mass	0.15 [m]
I_1	inertia of pendulum c.o.m. about local x-axis	0.0094 [kg-m ²]
I_2	inertia of pendulum c.o.m. about local y-axis	0.0079 [kg-m ²]
I_3	inertia of pendulum c.o.m. about local z-axis	0.0023 [kg-m ²]
W	width between wheel centers	0.15 [m]
r	figure 8 width	10 [m]
ω	figure 8 frequency	0.5 [1/sec]

Table 8: Non-Holonomic Segbot Sliding Mode Control Values

Inner Control Loop			
$Kd11$	2.31	$Kd31$	10
$Kd12$	2.73	$Kd32$	15.64
$Kd13$	0	$Kd33$	21.11
$Kd21$	1.31	$\lambda_1(\psi_r)$	1
$Kd22$	3.72	$\lambda_2(\psi_l)$	1
$Kd23$	0	$\lambda_3(\theta)$	6.46

ψ_r, ψ_l, θ , and their time derivatives were set to zero for the initial conditions. The initial position of the axel center started at $(X, Y, \phi) = (-1, 0, \frac{\pi}{3})$. Figure 27 shows the open loop XY plane wheel contact position (a visual representation of the open loop control is given by Figure 14).

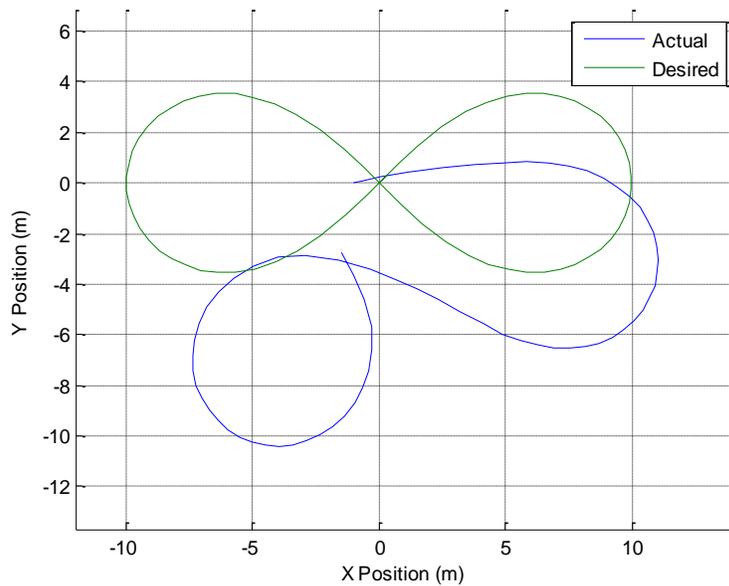


Figure 27: Segbot Open Loop XY Tracking

The dynamics of the Segbot do not allow a “nice” figure-8 path when the navigational control is not used. By that it is meant that the figure-8 path is not simply translated or rotated in the XY plane as was the case for the rolling wheel. This is due to the fact that there are now three rigid bodies, and the pendulum body can become unstable. The Segbot is an example of a dynamic system that exhibits non minimum phase. If the goal was to move the dynamic system forward, the wheels must first rotate backwards to get the pendulum leaning at the correct angle before beginning to move forward.

Figure 28 and Figure 29 show the simulated axel center position for various navigational gains when used with the inner loop control gains from Table 8.

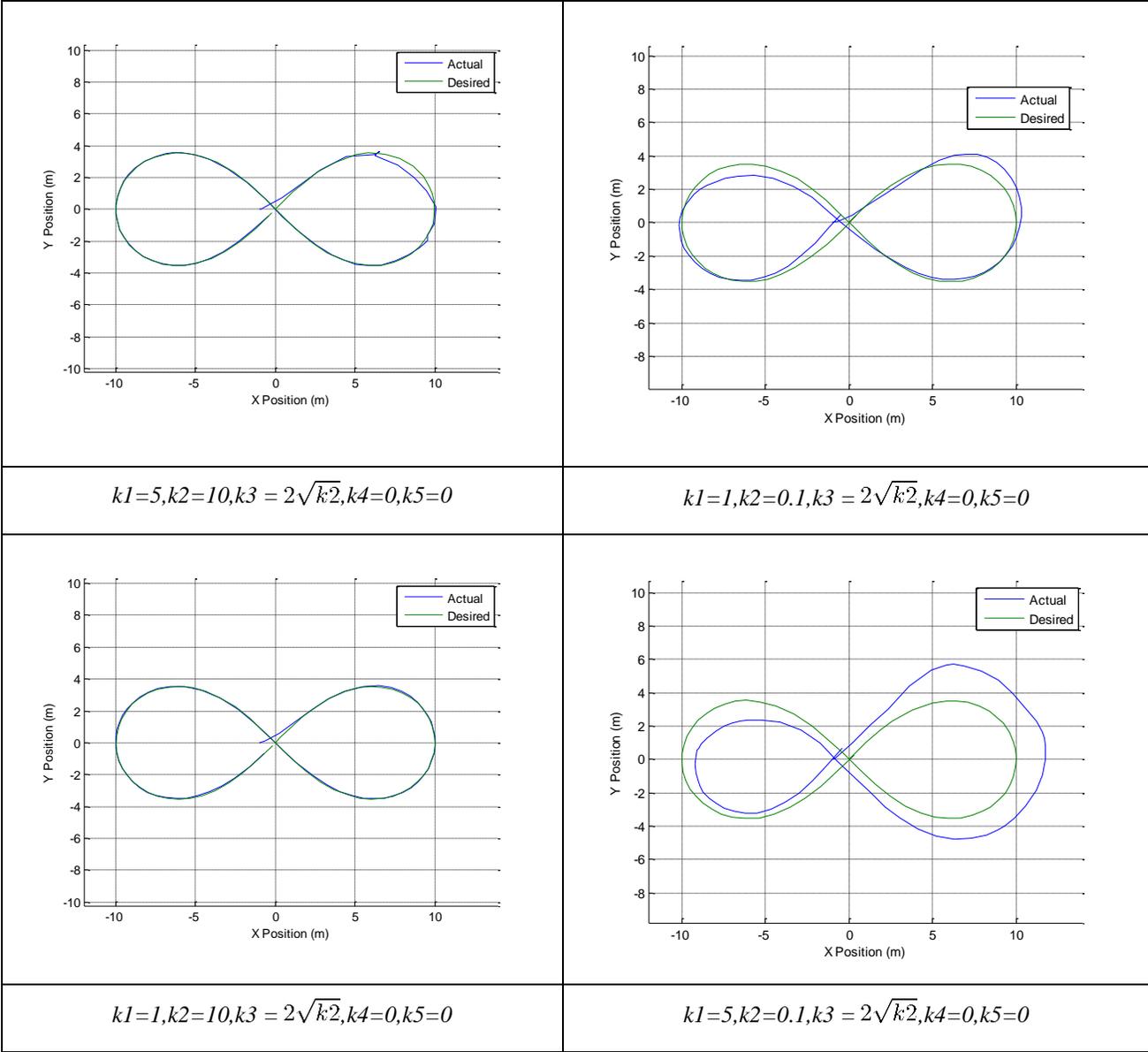


Figure 28: Non-Holonomic Segbot Simulation Response I

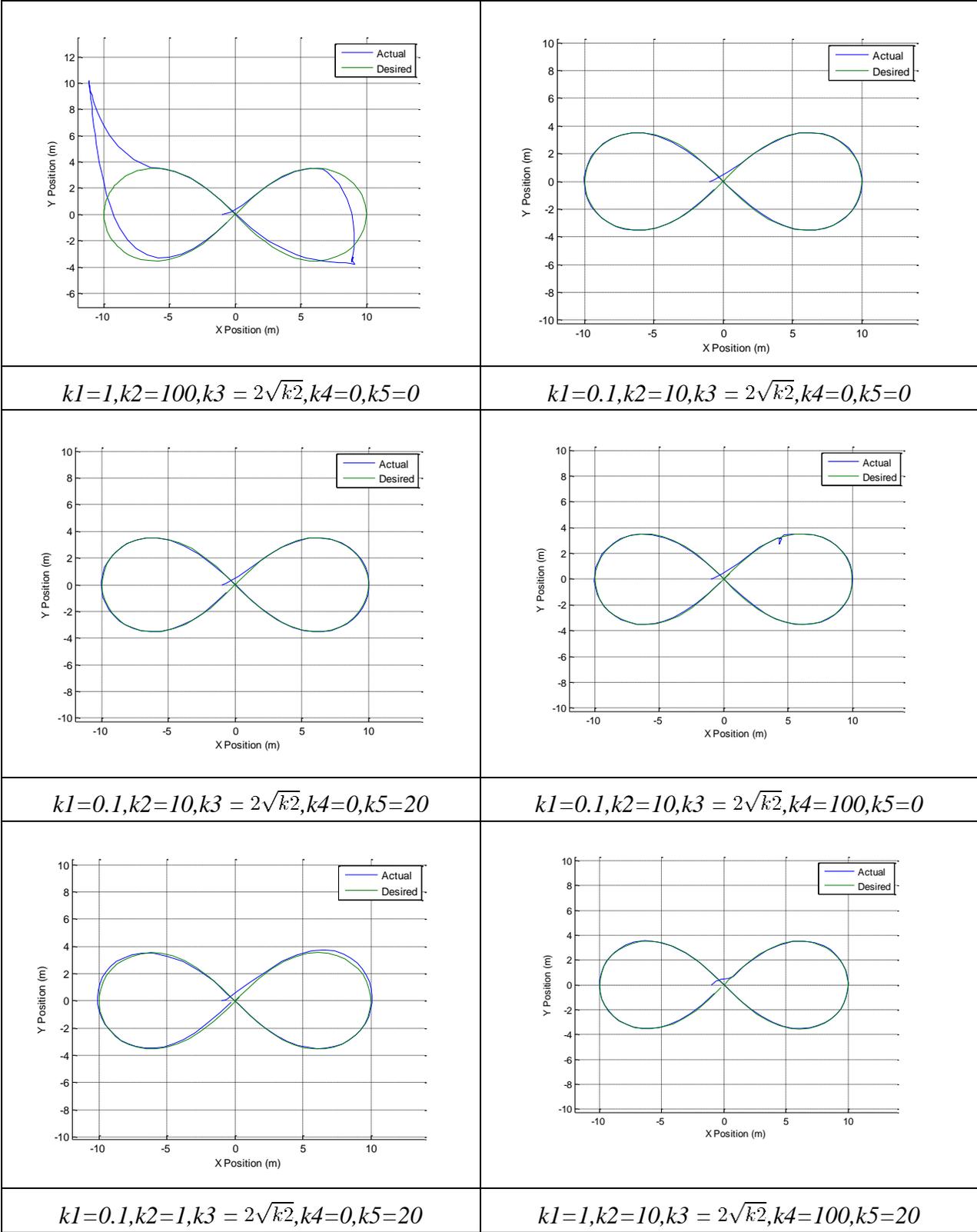


Figure 29: Non-Holonomic Segbot Simulation Response II

The plots in Figure 28 and Figure 29 show a few general trends that the navigational control gains exhibit when implemented on the Segbot. k_1 again more directly affects the e_d direction in Figure 15. If k_1 is picked too large, the system does not track as well. This is because of the non-minimum phase. k_2 has a more profound impact on the angular heading. If it is too small, the system does not track well. If it is too large, the system can go unstable. k_4 seems to affect the translational velocity more. k_5 in general affects the rotational component more. In general, k_4 and k_5 are larger than k_1 and k_2 . This is not too surprising as they are cubically related to the errors. However, it is desirable to first set k_4 and k_5 to zero to find an acceptable range for k_1 and k_2 .

Chapter Summary

This chapter first showed the derivation of the navigational control. Some additional control components were added so that the control proposed by Kanayama (1990) could be better implemented in conjunction with the presented sliding mode control. In order to track global XY coordinates, the navigational control is used to output desired values for the generalized coordinates used in the inner loop control. The underactuated sliding mode control presented in Chapter 2 is used as the inner loop control law.

X and Y are removed from the dynamic equations of motion before the sliding mode control is applied to the system. This is performed because calculating the control values used in the inner loop control is in general the most difficult part of the entire control law. In general, the inner loop calculation is simpler when $n-m$ is as small as possible. Removing X and Y from the equations of motion reduces $n-m$ by two for the inner control loop.

The inner loop control law values must first satisfy the conditions of a Lyapunov function. The inner loop control law values must also ensure that generalized coordinates stay in a stable region. Often some good assumptions such as system symmetry, dynamic relations, kinematic relations, and constraint relations can be used along with a linearized version of the dynamic system to find appropriate control gains for the inner control loop. It is often advisable to find a stable inner control loop before moving onto the global tracking problem.

The navigational control is a simple map that inputs the desired and actual position and heading direction as well as their time derivatives. From these inputs the navigational control

outputs a compensating translational and angular velocity. This control could be done entirely in a microprocessor with adequate state measurements and estimation. The navigational control is a path generation step that directs the current XY position to the desired XY trajectory.

Implementation of the navigational control simplifies the inner control loop by removing the X and Y generalized coordinates as well as two dynamic equations of motion from the system of equations that the inner control loop must be applied to, while still enabling global tracking of an XY trajectory.

If XY coordinate tracking is desired and no navigational control is used, then the system can attempt to track the desired XY trajectory through an open-loop control. Kinematic constraints are imposed so that in actuality the system tracks angular wheel positions and velocities, not X and Y coordinates. Because of the non-holonomic relationship between \dot{X}, \dot{Y} , and the system dynamics of the Segbot and rolling wheel, very poor tracking of XY occurs. If disturbances are added they cannot be compensated.

If the navigational control is used, the actual angular displacements of ψ_r and ψ_l are not important throughout the response. The position of X and Y is important. The system is driven to this trajectory by the navigational control providing the sliding mode control with the necessary desired angular velocities.

The rolling wheel example started out with the generalized coordinates $(\psi, \theta, \phi, X, Y)$ and two Lagrange multipliers through non-holonomic constraints. The Segbot started out with the generalized coordinates $(\psi_r, \psi_l, \theta, \phi, X, Y)$ as well as two Lagrange multipliers. Each dynamic system had only two actuators. There are five generalized coordinates for the rolling wheel and six for the Segbot before any simplifications are applied to the non-holonomic form of Lagrange's Equation. A systematic method was provided to find a control methodology to provide stability and tracking on two complex underactuated non-holonomic systems that show diverse dynamic characteristics from each other. The rolling wheel has strong coupling between the time derivative of the heading direction and the lean angle of the rolling wheel. The Segbot has strong coupling between the translational velocity of the axel center and the lean of the pendulum body.

Chapter 5: Segbot Implementation

This chapter shows the procedure to implement the navigational control and sliding mode control on an actual Segbot. The dynamic equations are derived similarly to that presented in Chapter 4, though some slight deviations occur. Some system identification procedures are presented. The dynamic response to a desired XY path is presented and the implementation of the sliding mode controller as the inner control loop is compared to using linear feedback gains for the inner control. Next a desired path for θ is given. The response of using the sliding mode control is compared to using linear feedback gains. The results are then summarized and suggestions for future implementations are presented.

Dynamic Equation Derivation

The kinetic energy of the right and left wheel were derived similarly to in Chapter 4. However, to determine some properties of the dynamic system, some simplifications and assumptions were used in the actual implementation. The procedure used to find the dynamic equations of motion follow. For convenience another picture of the Segbot is provided in Figure 30.

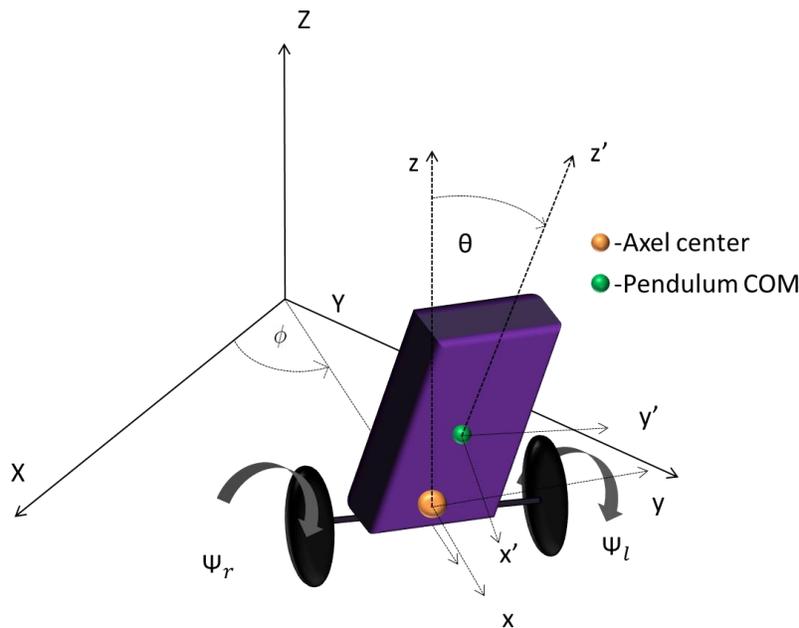


Figure 30: Non-Holonomic Segbot

ψ_r and ψ_l are the right and left wheel angles, respectively. θ is the angle of the Segbot body relative to vertical. The coordinates of a projection of the axel center onto the global XY plane are given by (X, Y) . The axel centered coordinates system is described by (x, y, z) . If this coordinate system is also rotated by θ , and the origin is translated to the pendulum body center of mass, the body centered frame (x', y', z') is given. It is assumed that the Segbot mass and inertial properties are symmetric.

X and Y describe the location of the axel center, ϕ describes the heading direction, and θ describes the angle of the pendulum with respect to vertical. ϕ is the heading direction, defined as right hand rule positive about the global Z axis and measured from the global X axis.

The rotational and translational kinetic energy components of the right and left wheels are given as

$$\begin{aligned} \text{KE}_{rw} &= \frac{1}{2}J_w\dot{\psi}_r^2 + \frac{1}{2}m_wR^2\dot{\psi}_r^2 \\ &\text{and} \\ \text{KE}_{lw} &= \frac{1}{2}J_w\dot{\psi}_l^2 + \frac{1}{2}m_wR^2\dot{\psi}_l^2 \end{aligned} \quad (99)$$

where m_w is the mass of the wheels and R is the radius of the wheels. Both wheels were assumed to be identical. The inertial component of the wheel about the z -axis is included in the inertia of the pendulum about the x' and z' axis. J_w is the combined inertial contribution from the wheel, the wheel's corresponding gearbox, and its corresponding motor.

The position of the pendulum body center of mass is given by the vector

$$r = \begin{bmatrix} X_{com} \\ Y_{com} \\ Z_{com} \end{bmatrix} = \begin{bmatrix} X + h_c \sin(\theta) \cos(\phi) \\ Y + h_c \sin(\theta) \sin(\phi) \\ R + h_c \cos(\theta) \end{bmatrix} \quad (100)$$

where h_c is the distance between the center of the axel and the pendulum body center of mass along the z' axis. The time derivative for the pendulum center of mass is then given by

$$\frac{d}{dt}(r) = \begin{bmatrix} \dot{X} + \dot{\theta}h_c \cos(\theta) \cos(\phi) - \dot{\phi}h_c \sin(\theta) \sin(\phi) \\ \dot{Y} + \dot{\theta}h_c \cos(\theta) \sin(\phi) + \dot{\phi}h_c \sin(\theta) \cos(\phi) \\ -\dot{\theta}h_c \sin(\theta) \end{bmatrix} \quad (101)$$

The rotational kinetic energy of the pendulum base is given by

$$\text{KE}_{rot} = \frac{1}{2} \left(I_1 \sin(\theta)^2 \dot{\phi}^2 + I_2 \dot{\theta}^2 + I_3 \cos(\theta)^2 \dot{\phi}^2 \right) \quad (102)$$

where I_1 , I_2 , and I_3 are the moments of inertia about the body fixed coordinates. Recall that I_1 and I_3 also include a small contribution from the inertia of the wheels about the z coordinate.

The total kinetic energy of the pendulum is given as the sum of the translational and rotational components as

$$\begin{aligned} \text{KE}_{pen} = & \frac{1}{2} m_p \left(\left(\dot{X} + \dot{\theta} h_c \cos(\theta) \cos(\phi) - \dot{\phi} h_c \sin(\theta) \sin(\phi) \right)^2 + \left(\dot{Y} + \dot{\theta} h_c \cos(\theta) \sin(\phi) + \dot{\phi} h_c \sin(\theta) \cos(\phi) \right)^2 + \dot{\theta}^2 h_c^2 (\sin(\theta))^2 \right) \\ & + \frac{1}{2} \left(I_1 \sin(\theta)^2 \dot{\phi}^2 + I_2 \dot{\theta}^2 + I_3 \cos(\theta)^2 \dot{\phi}^2 \right). \end{aligned} \quad (103)$$

The potential energy of the pendulum is given as

$$PE = h_c \cos(\theta) m_p g. \quad (104)$$

The Lagrangian is taken as the difference between the total kinetic energy and the potential energy of the pendulum. The holonomic constraint $\phi = \frac{R}{W} (\psi_r - \psi_l)$ and its time derivative

$\dot{\phi} = \frac{R}{W} (\dot{\psi}_r - \dot{\psi}_l)$ are substituted for ϕ and $\dot{\phi}$ to give the Lagrangian

$$\begin{aligned} L = & \frac{1}{2} J_w \dot{\psi}_l^2 + \frac{1}{2} m_w R^2 \dot{\psi}_l^2 + \frac{1}{2} J_w \dot{\psi}_r^2 + \frac{1}{2} m_w R^2 \dot{\psi}_r^2 + \frac{1}{2} m_p \left(\left(\dot{X} + \dot{\theta} h_c \cos(\theta) \cos\left(\frac{R(\psi_r - \psi_l)}{W}\right) - R(\dot{\psi}_r - \dot{\psi}_l) h_c \sin(\theta) \sin\left(\frac{R(\psi_r - \psi_l)}{W}\right) W^{-1} \right)^2 \right. \\ & + \left. \left(\dot{Y} + \dot{\theta} h_c \cos(\theta) \sin\left(\frac{R(\psi_r - \psi_l)}{W}\right) + R(\dot{\psi}_r - \dot{\psi}_l) h_c \sin(\theta) \cos\left(\frac{R(\psi_r - \psi_l)}{W}\right) W^{-1} \right)^2 + \dot{\theta}^2 h_c^2 (\sin(\theta))^2 \right) + \frac{1}{2} \left(I_1 \sin(\theta)^2 \dot{\phi}^2 + I_2 \dot{\theta}^2 + I_3 \cos(\theta)^2 \dot{\phi}^2 \right) \\ & - h_c \cos(\theta) m_p g. \end{aligned} \quad (105)$$

The Lagrangian, as well as the non-holonomic constraints

$$\begin{aligned} \lambda_1 = & \dot{X} - \frac{R}{2} (\dot{\psi}_r + \dot{\psi}_l) \cos(\phi) = 0 \\ \text{and} & \\ \lambda_2 = & \dot{Y} - \frac{R}{2} (\dot{\psi}_r + \dot{\psi}_l) \sin(\phi) = 0 \end{aligned} \quad (106)$$

are substituted into the non-holonomic form of Lagrange's equation given by Equation (3). This procedure yields the following five equations of motion:

$$\begin{aligned}
eq(X) &= \frac{1}{W^2} \left(-\dot{h}_c m_p (-\ddot{\theta}(t) W^2 \cos(\theta) + (W^2 Dtheta^2 + R^2 (-\dot{\psi}_r(t) + \dot{\psi}_l(t))^2) \sin(\theta)) \cos\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) + W \left(\dot{h}_c m_p ((-\ddot{\psi}_r(t) + \ddot{\psi}_l(t)) \sin(\theta) + 2 Dtheta \cos(\theta) (-\dot{\psi}_r(t) + \dot{\psi}_l(t))) R \sin\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) - (-m_p DDx + \lambda_1) W \right) \right) \\
eq(Y) &= \frac{1}{W^2} \left(-\dot{h}_c m_p (-\ddot{\theta}(t) W^2 \cos(\theta) + (W^2 Dtheta^2 + R^2 (-\dot{\psi}_r(t) + \dot{\psi}_l(t))^2) \sin(\theta)) \sin\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) - W \left(\dot{h}_c m_p ((-\ddot{\psi}_r(t) + \ddot{\psi}_l(t)) \sin(\theta) + 2 Dtheta \cos(\theta) (-\dot{\psi}_r(t) + \dot{\psi}_l(t))) R \cos\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) + (-m_p DDy + \lambda_2) W \right) \right) \\
eq(\Psi_r) &= \frac{1}{2 W^2} \left(-4 R^2 m_p \sin(\theta) \left(\left(-\frac{\ddot{\psi}_r(t)}{2} + \frac{\ddot{\psi}_l(t)}{2} \right) \sin(\theta) + Dtheta \cos(\theta) (-\dot{\psi}_r(t) + \dot{\psi}_l(t)) \right) \dot{h}_c^2 \cos\left(\frac{R(\Psi_r - \Psi_l)}{W}\right)^2 \right. \\
&\quad + 2 m_p R \dot{h}_c \sin(\theta) \cos\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) DDy W - 4 R^2 m_p \sin(\theta) \left(\left(-\frac{\ddot{\psi}_r(t)}{2} + \frac{\ddot{\psi}_l(t)}{2} \right) \sin(\theta) + Dtheta \cos(\theta) (-\dot{\psi}_r(t) + \dot{\psi}_l(t)) \right) \\
&\quad \dot{h}_c^2 \sin\left(\frac{R(\Psi_r - \Psi_l)}{W}\right)^2 - 2 m_p R \dot{h}_c \sin(\theta) \sin\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) DDx W - 2 R^2 II (-\ddot{\psi}_r(t) + \ddot{\psi}_l(t)) \sin(\theta)^2 - 4 R^2 Dtheta \cos(\theta) (-I3 \\
&\quad + II) (-\dot{\psi}_r(t) + \dot{\psi}_l(t)) \sin(\theta) - 2 R^2 I3 (-\ddot{\psi}_r(t) + \ddot{\psi}_l(t)) \cos(\theta)^2 + 2 \left(\ddot{\psi}_r(t) m_w R^2 + \frac{\cos(\phi) \lambda_1 R}{2} - \tau_r + J_w \ddot{\psi}_r(t) + \frac{\sin(\phi) \lambda_2 R}{2} \right. \\
&\quad \left. + b_w \dot{\psi}_r(t) \right) W^2 \Big) \\
eq(\Psi_l) &= \frac{1}{2 W^2} \left(4 R^2 m_p \sin(\theta) \left(\left(-\frac{\ddot{\psi}_r(t)}{2} + \frac{\ddot{\psi}_l(t)}{2} \right) \sin(\theta) + Dtheta \cos(\theta) (-\dot{\psi}_r(t) + \dot{\psi}_l(t)) \right) \dot{h}_c^2 \cos\left(\frac{R(\Psi_r - \Psi_l)}{W}\right)^2 \right. \\
&\quad - 2 m_p R \dot{h}_c \sin(\theta) \cos\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) DDy W + 4 R^2 m_p \sin(\theta) \left(\left(-\frac{\ddot{\psi}_r(t)}{2} + \frac{\ddot{\psi}_l(t)}{2} \right) \sin(\theta) + Dtheta \cos(\theta) (-\dot{\psi}_r(t) + \dot{\psi}_l(t)) \right) \\
&\quad \dot{h}_c^2 \sin\left(\frac{R(\Psi_r - \Psi_l)}{W}\right)^2 + 2 m_p R \dot{h}_c \sin(\theta) \sin\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) DDx W + 2 R^2 II (-\ddot{\psi}_r(t) + \ddot{\psi}_l(t)) \sin(\theta)^2 + 4 R^2 Dtheta \cos(\theta) (-I3 \\
&\quad + II) (-\dot{\psi}_r(t) + \dot{\psi}_l(t)) \sin(\theta) + 2 R^2 I3 (-\ddot{\psi}_r(t) + \ddot{\psi}_l(t)) \cos(\theta)^2 + 2 W^2 \left(\ddot{\psi}_l(t) m_w R^2 + \frac{\sin(\phi) \lambda_2 R}{2} + b_w \dot{\psi}_l(t) + J_w \ddot{\psi}_l(t) \right. \\
&\quad \left. + \frac{\cos(\phi) \lambda_1 R}{2} - \tau_l \right) \Big) \\
eq(\theta(t)) &= \frac{1}{N W^2} \left(- \left(\frac{d^2}{dt^2} \theta(t) \right) W^2 \cos(\theta) + \sin(\theta) \left(W^2 \left(\frac{d}{dt} \theta(t) \right)^2 + R^2 (Dpsil - Dpsir)^2 \right) \right) m_p N \dot{h}_c^2 \cos(\theta) \cos\left(\frac{R(\Psi_r - \Psi_l)}{W}\right)^2 \\
&\quad + m_p \dot{h}_c \cos(\theta) \cos\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) \left(\frac{d^2}{dt^2} X(t) \right) W^2 N - \left(\frac{d^2}{dt^2} \theta(t) \right) W^2 \cos(\theta) + \sin(\theta) \left(W^2 \left(\frac{d}{dt} \theta(t) \right)^2 + R^2 (Dpsil - Dpsir)^2 \right) \right) m_p N \dot{h}_c^2 \cos(\theta) \sin\left(\frac{R(\Psi_r - \Psi_l)}{W}\right)^2 \\
&\quad + m_p \dot{h}_c \cos(\theta) \sin\left(\frac{R(\Psi_r - \Psi_l)}{W}\right) \left(\frac{d^2}{dt^2} Y(t) \right) W^2 N + N W^2 (I2 + m_p \dot{h}_c^2 \sin(\theta)^2) \left(\frac{d^2}{dt^2} \theta(t) \right) + m_p \dot{h}_c^2 \sin(\theta) W^2 \left(\frac{d}{dt} \theta(t) \right)^2 N \cos(\theta) - N R^2 \sin(\theta) (Dpsil - Dpsir)^2 (II \\
&\quad - I3) \cos(\theta) - W^2 (\dot{h}_c \sin(\theta) m_p g N - \tau_l - \tau_r) \Big)
\end{aligned} \tag{107}$$

Notice that τ_r and τ_l are applied in the positive direction on $eq(\psi_r)$ and $eq(\psi_l)$, while they are applied in an equal and opposite direction on the Segbot body for $eq(\theta)$ in Equation (107). τ_r and τ_l represent the torque that accelerates the system applied by the left and right motors.

The other non-conservative force added is viscous friction due the rotation of the gears and motor. The coefficient b_w is assumed to be the same for each wheel. The viscous friction term used for b_w combines viscous friction terms from the motor and the gear box. The time derivatives of the non-holonomic constraints are given by

$$\ddot{X} + \frac{R}{2} \left(-(\ddot{\psi}_r + \ddot{\psi}_l) \cos(\phi) + (\dot{\psi}_r + \dot{\psi}_l) \sin(\phi) \dot{\phi} \right) = 0$$

and

$$\ddot{Y} + \frac{R}{2} \left(-(\ddot{\psi}_r + \ddot{\psi}_l) \sin(\phi) - (\dot{\psi}_r + \dot{\psi}_l) \cos(\phi) \dot{\phi} \right) = 0. \quad (108)$$

The non-holonomic constraints given by Equation (106), their time derivatives given by Equation (108), and $eq(X)$ and $eq(Y)$ from Equation (107) are solved for λ_1 , λ_2 , \dot{X} , \ddot{X} , \dot{Y} , and \ddot{Y} . This solution is then inserted into $eq(\psi_r)$, $eq(\psi_l)$, and $eq(\theta)$ from Equation (107). This is performed to reduce $n-m$ to as small of a number as possible. In matrix form the final three equations of motion can be written as

$$M\ddot{q} + C\dot{q} + G = -C_d\dot{q} + \tau_{accel} = \tau. \quad (109)$$

The matrices of Equation (109) are

$$M = \begin{bmatrix} \frac{-4R^2(-I_3 + mp\,hc^2 + II)\cos(\theta)^2 + ((4mv + mp)W^2 + 4II + 4mp\,hc^2)R^2 + 4JwmW^2}{4W^2} & \frac{R^2((-4I_3 + 4mp\,hc^2 + 4II)\cos(\theta)^2 + (W^2 - 4hc^2)mp - 4II)}{4W^2} & \frac{Rmp\,hc\cos(\theta)}{2} \\ \frac{R^2((-4I_3 + 4mp\,hc^2 + 4II)\cos(\theta)^2 + (W^2 - 4hc^2)mp - 4II)}{4W^2} & \frac{-4R^2(-I_3 + mp\,hc^2 + II)\cos(\theta)^2 + ((4mv + mp)W^2 + 4II + 4mp\,hc^2)R^2 + 4JwmW^2}{4W^2} & \frac{Rmp\,hc\cos(\theta)}{2} \\ \frac{Rmp\,hc\cos(\theta)}{2} & \frac{Rmp\,hc\cos(\theta)}{2} & II + mp\,hc^2 \end{bmatrix},$$

$$C = \begin{bmatrix} \frac{Dtheta R^2(-I_3 + mp\,hc^2 + II)\cos(\theta)\sin(\theta)}{W^2} & \frac{R^2\sin(\theta)hc\,mp(-Dpsir + Dpsil)}{W^2} & \frac{2\sin(\theta)R\left(\frac{Dpsil - Dpsir}{2}\right)R(-I_3 + mp\,hc^2 + II)\cos(\theta) + \frac{Dtheta\,mp\,hc\,W^2}{4}}{W^2} \\ \frac{(-2Dtheta(-I_3 + mp\,hc^2 + II)\cos(\theta) + Rmp\,hc(-Dpsir + Dpsil))\sin(\theta)R^2}{W^2} & \frac{Dtheta R^2(-I_3 + mp\,hc^2 + II)\cos(\theta)\sin(\theta)}{W^2} & \frac{\sin(\theta)\left(RDpsil(-I_3 + mp\,hc^2 + II)\cos(\theta) - \frac{Dtheta\,mp\,hc\,W^2}{2}\right)R}{W^2} \\ \frac{R^2\sin(\theta)(-I_3 + mp\,hc^2 + II)\cos(\theta)(-Dpsir + 2Dpsil)}{W^2} & \frac{R^2\sin(\theta)Dpsil(-I_3 + mp\,hc^2 + II)\cos(\theta)}{W^2} & 0 \end{bmatrix},$$

$$C_d = \begin{bmatrix} b_w & 0 & 0 \\ 0 & b_w & 0 \\ -b_w & -b_w & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 0 \\ -h_c \sin(\theta) m_p g \end{bmatrix}, \dot{q} = \begin{bmatrix} \dot{\psi}_r \\ \dot{\psi}_l \\ \dot{\theta} \end{bmatrix}, \ddot{q} = \begin{bmatrix} \ddot{\psi}_r \\ \ddot{\psi}_l \\ \ddot{\theta} \end{bmatrix}, \text{ and } \tau = \begin{bmatrix} \tau_r \\ \tau_l \\ -(\tau_r + \tau_l) \end{bmatrix}$$

where τ_{accel} is the torque that causes the dynamic system to accelerate and τ is the sum of τ_{accel} as well as the torque generated by viscous friction. This is performed because the viscous friction can

be removed from the dynamic model if its contribution is compensated by the back emf constant as explained in the following section.

DC Motor and Motor Driver Modeling

The torque of a DC motor can be approximated by

$$\tau = K_t i_a \quad (110)$$

where τ is the total torque, K_t is the torque constant, and i_a is the armature current. Using this model includes the viscous friction component of torque as well as the torque that causes acceleration. This assumption can be observed upon examining Equation (109). The sliding mode control law calculates a desired torque for each motor. This torque can be converted to a desired current using Equation (110).

The voltage of a DC motor can be found through the differential equation

$$V = i_a R_a + K_b \dot{\psi} + L \frac{di}{dt} \quad (111)$$

where V is the voltage, i_a is the armature current, R_a is the armature resistance, K_b is the back emf constant, $\dot{\psi}$ is the angular velocity of the motor, L is the armature inductance, and $\frac{di}{dt}$ is the rate of change of current with respect to time. A reasonable assumption that will be made is that the time constant for the electrical properties of the system is much smaller than the time constant for the mechanical properties. This simplifies Equation (111) to

$$V = i_a R_a + K_b \dot{\psi}. \quad (112)$$

To simplify the cost of implementation, the desired voltage will be back calculated from the torque calculation from the inner control loop. The inner control loop calculates a desired torque. This torque is converted to a voltage to apply to the motor terminals. The Segbot motor driver is then used to apply the voltage to the motor. The motor driver was configured for a drive-brake voltage supply to the motor terminals. In this configuration the current is allowed to flow through the DC motor when the pulse width modulation (PWM) input to the motor driver is high, and the leads to the DC motor are shorted together when the PWM input to the motor driver is low. This allows the linear relationship

$$V_{out} = V_s D_{in} \quad (113)$$

where D_{in} is the duty cycle of the motor driver PWM input from 0-1.0, V_{out} is the average voltage supplied to the DC motor terminals, and V_s is the supply voltage to the motor driver. The magnitude of the V_{out} pulses can be approximated as the magnitude of V_s . The duty cycle of V_{out} can be approximated as the duty cycle of D_{in} . Due to the induction of the DC motor the current continues to flow during the low side of the PWM applied to the motor terminals and V_{out} can be approximated by Equation (113). Another option to control the torque would be to implement an amplifier or op amp to directly control a current supply. It should be noted that static friction is not included in this model.

System Identification: R_a and K_b

A power supply provided a constant 12.08 [volts] to the motor driver. The duty cycle of the PWM input to the motor driver was then varied from 0 to 1. The wheel was allowed to spin freely, while voltage [volts] and current [amps] measurements were taken across the DC motor terminals. Angular position was determined using an optical encoder and a discrete observer that used a fixed time step of 0.001 [seconds] was used to estimate the angular velocity in [rad/sec].

Motor characteristics were measured for two different motor operating configurations. The first configuration consisted of the wheel being held at a fixed position while various voltages were supplied across the DC motor terminals. The corresponding current output from the motor driver to the motor was measured. The next operating mode allowed the wheel to spin freely. The corresponding DC motor current and wheel speed were measured along with the voltage measured across the DC motor terminals.

The measurements from each motor configuration were inserted into Equation (112) to calculate the best guess for the voltage and the least squares method was used to estimate values for the armature resistance and back emf constant. The resistance is given in ohms, while the back emf constant is given by $\left[\frac{V-s}{rad}\right]$. The Excel add-in “Solver” was used to find R_a and K_b to minimize the sum of $V_{err}^2 = (V_{meas} - V_{calc})^2$, where V_{meas} is the voltage measured across the motor terminals and V_{calc} is the best guess of the voltage across the terminals using Equation (112). The results are presented in Table 9.

$$\psi(t) = V_s \left(\frac{b_2 t}{b_1} - \frac{b_2}{b_1^2} + \frac{b_2}{b_1^2} e^{b_1 t} \right) \quad (116)$$

where t is the time in seconds, V_s is the magnitude of the voltage step input, e is Euler's number, $b_1 = \frac{K_t K_b}{R_a J}$, and $b_2 = \frac{K_t}{J R_a}$. (117)

Again the least squares method is used to determine the unknown constants. The best estimate of the wheel position to a voltage step input is given by Equations (116) and (117). V_s is a constant while R_a , K_t , and K_b were found earlier. The Excel add-in "Solver" is used to determine the value of J that minimizes the error for the sum of the position errors squared.

Table 10: Inertia and Viscous Friction Constants

Time [sec]	Right Wheel Measured [rad]	Right Wheel Guess Use Equation	Right Wheel Deviation^2	Left Wheel Measured [rad]	Left Wheel Use Equation	Left Wheel Deviation^2	From Motor Data	
0	0	0	0	0	0	0	Kb	0.594198
0.001	0	0.000183429	3.36462E-08	0	0.000176318	3.1088E-08	Kt	0.594198
0.002	0	0.00072922	5.31762E-07	0	0.000701127	4.9158E-07	Ra	12.35609
0.003	0	0.001630721	2.65925E-06	0	0.001568295	2.4595E-06	Deviation^2 Sum	
0.004	0.001	0.002881402	3.53967E-06	0.001	0.002771796	3.1393E-06	0.008267317	
0.005	0.001	0.004474854	1.20746E-05	0.001	0.004305712	1.0928E-05		
0.007	0.004	0.008665017	2.17624E-05	0.004	0.008341629	1.885E-05	J=	0.00157
0.009	0.008	0.014152248	3.78502E-05	0.008	0.013630746	3.1705E-05		
0.01	0.011	0.017367451	4.05444E-05	0.011	0.01673153	3.285E-05		
0.012	0.017	0.024712355	5.94804E-05	0.017	0.023818946	4.6498E-05		
0.014	0.025	0.033239569	6.78905E-05	0.024	0.0320531	6.4852E-05		
0.016	0.033	0.042906086	9.81305E-05	0.033	0.041393985	7.0459E-05		
0.018	0.043	0.053670462	0.000113859	0.043	0.051802989	7.7493E-05		
0.02	0.054	0.065492762	0.000132084	0.053	0.06324285	0.00010492		
0.022	0.067	0.078334504	0.000128471	0.066	0.075677601	9.3656E-05		

For the data in Table 10, a constant voltage of 11.94 [volts] is supplied for V_s . A final value of $J = 0.00157[kg - m^2]$ was calculated.

Remaining Parameters

I_1 , I_2 , I_3 , and h_c are calculated in a Solidworks model. Direct measurements were taken on all of the individual parts of the Segbot and then put into an assembly. The remaining dynamic parameters present in the equations of motion could be directly measured with a scale or micrometer. A summary of the complete system parameters is outlined in Table 11.

Table 11: Segbot Implementation Parameters

Constant	Description	Value	Units
J	Combined Moment of Inertia for Wheel, Gearbox, and Motor	0.00157	[kg-m ²]
I_1	Moment of Inertia of Pendulum and Wheels About Local x'-Axis	0.00153	[kg-m ²]
I_2	Moment of Inertia of Pendulum About Local y'-Axis	0.00111	[kg-m ²]
I_3	Moment of Inertia of Pendulum and Wheels About Local z'-Axis	0.00089	[kg-m ²]
W	Distance Between Wheel Centers	0.1639	[m]
m_w	Wheel Mass	0.0332	[kg]
R	Wheel Radius	0.0508	[m]
m_p	Mass of the Pendulum	1.2398	[kg]
h_c	Distance From Axle Center to Pendulum C.O.M.	0.0746	[m]
g	Gravitational Constant	9.81	[m/s ²]
K_m	Back EMF constant	0.594	[V-s/rad]
K_t	Torque Constant	0.594	[N-m/amp]
R_a	Armature Resistance	12.36	[ohm]

Because of the configuration of the gearbox and the DC motor, the magnitude of the torque applied to the Segbot body is equal in magnitude, but in the opposite direction of the torque applied to its corresponding wheel, even though the gear ratio is 30. The opposite direction of the applied torque on the Segbot body is indicated by a negative sign in the equations of motion.

It should also be noted that all angle measurements are made in radians. All angular measurements that refer to the motor or wheel are referred to the wheel angle. (i.e. K_b would be multiplied by the angular velocity of the wheel, not the angular velocity of the motor)

State Determination

Encoders were mounted on each DC motor. The resolution was $420 \left[\frac{\text{counts}}{\text{motor rev}} \right]$, and the gear ratio was 30. A linear discrete observer calculated the angular velocity of each wheel. The observer is included in Appendix C.

In order to estimate the lean angle of the Segbot, a MPU-6050 3-axis gyroscope and 3-axis accelerometer was used. $\dot{\theta}$ was measured directly by the gyroscope. In order to estimate θ a complementary filter was implemented. The estimate of θ was given by

$$\theta_{\text{actual}} = 0.998\theta_{\text{gyro}} + 0.002\theta_{\text{accel}} \quad (118)$$

where θ_{actual} is the best estimate of θ , θ_{gyro} is the sum of the previous θ_{actual} and the gyroscope reading for $\dot{\theta}$ multiplied by the time step, and θ_{accel} is the estimate of the pendulum angle given by the accelerometer readings. Complementary filters are widely used and the reader is encouraged to visit the webpage given by *Gyroscopes and Accelerometers on a Chip* (2013) for more information.

All state estimates, besides angular velocities, were calculated in a loop that runs on available processor time. The angular velocities, desired trajectory generation, and control law values were calculated in a timed loop that runs at 100 [Hz]. This time step is small enough to estimate the control law as continuous, while still providing enough time to perform the necessary calculations. The timed loop first calculates a desired torque for each motor. This is performed through the use of the sliding mode control law. The desired torque values are converted to desired currents through the use of Equation (110). The currents can be converted to desired voltage outputs, which can in turn be converted to duty cycles through the application of Equations (112) and (113). The duty cycle of the PWM sent to the motor driver for each motor is calculated by

$$D = \frac{R_a}{K_t V_s} (\tau + \dot{\psi} K_b) \quad (119)$$

where D is the duty cycle of the motor, R_a is the armature resistance, K_t is the torque constant, V_s is the supply voltage, τ is the torque calculate by the control law, $\dot{\psi}$ is the angular velocity of the wheel, and K_b is the back emf constant. The Labview VIs are included in Appendix C.

Linear Control Gains

The first goal was to find a linear set of gains that could stabilize the system. The dynamic model given by Equation (109) was linearized about $(\theta = 0, \dot{\theta} = 0, \dot{\psi}_r = 0, \dot{\psi}_l = 0)$. The physical parameters listed in Table 11 were inserted into the linearized equations and the dynamic system was put into state-space form to yield

$$\dot{x} = \begin{bmatrix} \dot{\psi}_r \\ \dot{\psi}_l \\ \dot{\theta} \\ \ddot{\psi}_r \\ \ddot{\psi}_l \\ \ddot{\theta} \end{bmatrix} = Ax + Bu$$

$$= \begin{bmatrix} 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 \\ 0 & 0 & -141.748 & 0 & 0 & 0 \\ 0 & 0 & -141.748 & 0 & 0 & 0 \\ 0 & 0 & 196.426 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 696.232 & 148.883 \\ 148.883 & 696.232 \\ -372.719 & -372.719 \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}.$$

Bryson's rule [Franklin et al. (2010)] was used to determine an initial set of linear gains to stabilize the dynamic system. The weighting matrices were defined as

$$Q = \begin{bmatrix} \left(\frac{1}{\psi_{max}}\right)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \left(\frac{1}{\psi_{max}}\right)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \left(\frac{1}{\theta_{max}}\right)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \left(\frac{1}{\psi_{max}}\right)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \left(\frac{1}{\psi_{max}}\right)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \left(\frac{1}{\theta_{max}}\right)^2 \end{bmatrix} \text{ and } R = \begin{bmatrix} \left(\frac{1}{K_I V_{max}}\right)^2 & 0 \\ 0 & \left(\frac{1}{K_I V_{max}}\right)^2 \end{bmatrix} \quad (120)$$

where $\psi_{max} = 10$, $\theta_{max} = 0.05$, $\dot{\psi}_{max} = 10$, $\dot{\theta}_{max} = 5$, and $V_{max} = 10$. These weighting matrices were used along with the Matlab command $K = -lqr(A, B, Q, R)$ to determine the feedback gains

$$K = \begin{bmatrix} 0 & 0.047 & 8.848 & 0.035 & 0.070 & 0.290 \\ 0.047 & 0 & 8.848 & 0.070 & 0.035 & 0.290 \end{bmatrix}. \quad (121)$$

In implementation, the state feedback gains from Equation (121) could stabilize the Segbot. However, the values were altered slightly to yield a more desirable response. The final linear gains used to stabilize the Segbot were

$$K = \begin{bmatrix} 0 & 0.12 & 10.27 & 0.026 & 0.14 & 0.19 \\ 0.12 & 0 & 10.27 & 0.14 & 0.026 & 0.19 \end{bmatrix}. \quad (122)$$

Sliding Mode Control Initialization

The dynamic equations of motion used to derive the sliding mode control is given by $M\ddot{q} + C\dot{q} + G = -C_d\dot{q} + \tau_{\text{accel}} = \tau$. Because of the way that the back *emf* constant and torque constant are used in this Chapter, the control law actually calculates the total torque to be applied by each motor of the Segbot. The total torque includes both the torque to be applied to compensate for viscous friction and the torque required to accelerate the system. This removes the viscous friction coefficient contribution from the dynamic equations used to derive the control law. The matrices M , C , and G are inserted into the sliding mode control $\tau = M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + G(q) - K_d s$. The system's dynamic matrices given by Equation (109) and the dynamic properties given by Table 11 were then inserted into the sliding mode control. Both the left and right motor apply a torque to the corresponding wheel. There is a reaction torque of equal magnitude but in the opposite direction that acts on the pendulum body. The sum of these torques must then cancel each other out. Therefore the first, second, and third rows of this control law were summed together. The sum was then solved for $\ddot{\theta}_d$ to ensure the sliding mode control is underactuated. The process provides

$$\begin{aligned} \ddot{\theta}_d(t) = & \frac{1}{2 W^2 (m_p h_c^2 + R m_p h_c \cos(\theta) + I2)} \left(-2 \left(\frac{R m_p h_c \cos(\theta)}{2} + \left(m_w + \frac{m_p}{2} \right) R^2 + J \right) W^2 \ddot{\psi}_{rd}(t) - 2 \left(\frac{R m_p h_c \cos(\theta)}{2} + \left(m_w + \frac{m_p}{2} \right) R^2 + J \right) W^2 \right. \\ & \ddot{\psi}_{rd}(t) + \left(-2 \left((2 m_p h_c^2 - 2 I3 + 2 I1) \cos(\theta) + R m_p h_c \right) R^2 \sin(\theta) \dot{\psi}_{rd}(t) + 2 \left((-I3 + m_p h_c^2 + I1) \cos(\theta) + R m_p h_c \right) R^2 \sin(\theta) \dot{\psi}_{rd}(t) \right. \\ & + 2 R^2 \cos(\theta) \sin(\theta) (-I3 + m_p h_c^2 + I1) \dot{\theta}_d(t) - 2 \left(R \lambda 3 (-I3 + m_p h_c^2 + I1) (\theta - \theta_d) \sin(\theta) - \frac{m_p h_c W^2 \lambda 2}{2} \right) R \cos(\theta) + 2 \left(\frac{\lambda 2 (2 m_w + m_p) R^2}{2} \right. \\ & + J \lambda 2 + Km22 + Km32 + Km12 \left. \right) W^2 \dot{\psi}_r(t) + \left(2 \left((-I3 + m_p h_c^2 + I1) \cos(\theta) + R m_p h_c \right) R^2 \sin(\theta) \dot{\psi}_{rd}(t) - 2 R^3 \sin(\theta) h_c m_p \dot{\psi}_{rd}(t) \right. \\ & - 2 R^2 \cos(\theta) \sin(\theta) (-I3 + m_p h_c^2 + I1) \dot{\theta}_d(t) + 2 R \left(R \lambda 3 (-I3 + m_p h_c^2 + I1) (\theta - \theta_d) \sin(\theta) + \frac{m_p h_c W^2 \lambda 1}{2} \right) \cos(\theta) + 2 \left(\frac{\lambda 1 (2 m_w + m_p) R^2}{2} \right. \\ & + Km21 + Km31 + Km11 + J \lambda 1 \left. \right) W^2 \dot{\psi}_r(t) + \left(2 R^2 \cos(\theta) \sin(\theta) (-I3 + m_p h_c^2 + I1) \dot{\theta}(t) - 2 \left(\frac{R m_p h_c \cos(\theta) \lambda 1}{2} + \frac{\lambda 1 (2 m_w + m_p) R^2}{2} + Km21 \right. \right. \\ & + Km31 + Km11 + J \lambda 1 \left. \right) W^2 \dot{\psi}_{rd}(t) + \left(-2 R^2 \cos(\theta) \sin(\theta) (-I3 + m_p h_c^2 + I1) \dot{\theta}(t) - 2 \left(\frac{R m_p h_c \cos(\theta) \lambda 2}{2} + \frac{\lambda 2 (2 m_w + m_p) R^2}{2} + J \lambda 2 + Km22 \right. \right. \\ & + Km32 + Km12 \left. \right) W^2 \dot{\psi}_{rd}(t) + 2 \left((\dot{\theta}_d(t) h_c m_p R \sin(\theta) + \lambda 3 R \cos(\theta) h_c m_p - \lambda 3 R h_c m_p (\theta - \theta_d) \sin(\theta) + Km33 + \lambda 3 h_c^2 m_p + Km23 + Km13 \right. \\ & + \lambda 3 I2) \dot{\theta}(t) + (-\lambda 3 R \cos(\theta) h_c m_p - Km13 - \lambda 3 h_c^2 m_p - Km33 - \lambda 3 I2 - Km23) \dot{\theta}_d(t) + h_c \sin(\theta) m_p g + \lambda 3 (\theta - \theta_d) (Km33 + Km23 + Km13) \left. \right) \\ & W^2 \end{aligned} \quad (123)$$

This value for $\ddot{\theta}_d$ was used in the sliding mode control. The first line of the control law gives the desired torque for the right wheel, while the second line of the control law gives the desired torque for the left wheel. The remaining task is to choose appropriate gains for Λ and K_d .

Non-Holonomic Coordinate Tracking

The sliding mode control will first be used in conjunction with the navigational control to track the global X and Y coordinates given by

$$\begin{aligned}
 X_d &= r \frac{\cos(\omega t)}{(1 + \sin(\omega t)^2)} \\
 \text{and} \\
 Y_d &= r \frac{\cos(\omega t) \sin(\omega t)}{(1 + \sin(\omega t)^2)}
 \end{aligned} \tag{124}$$

where ω is the angular frequency of the figure-8 and $2r$ is the width across the figure-8. Equation (124) is very similar to Equation (87). Equation (124) can be differentiated to determine $\dot{X}_d, \dot{Y}_d, \ddot{X}_d$, and \ddot{Y}_d . Because the path is written with trig functions instead of square roots and logic tests for the sign of the desired path values, the desired coordinates can be processed faster and cleaner. This is especially helpful when calculating desired time derivatives.

It should be noted that the actual (X, Y) contact position was calculated by integrating \dot{X} and \dot{Y} from Equation (106). The values for \dot{X} and \dot{Y} are calculated through the use of an observer on the encoder readings of each wheel to calculate the angular velocity of each wheel. The actual encoder readings are to determine the heading direction. The width between the wheels, W , may vary slightly during implementation due to the motor shafts slightly sliding in or out. There could be other slight violations on the roll without slip constraint. This means that integrating \dot{X} and \dot{Y} will slowly drift away from the actual contact position over time. However, drift was very minimal. Because the drift is very slow, integrating the values obtained for \dot{X} and \dot{Y} to calculate X and Y will yield suitable position coordinates to test the dynamic response to control parameters. Using cameras to compliment the encoders to calculate the axel center coordinate would yield better position calculations over a period of time.

Sliding Mode Control Terms for Figure-8 Trajectory Tracking

The linear feedback gains given by Equation (122) use the generalized coordinates $(\psi_r, \psi_l, \theta, \dot{\psi}_r, \dot{\psi}_l, \dot{\theta})$. Tracking a specific trajectory for ψ_r and ψ_l is not important to track the figure-8 path. The coordinates X and Y are important. The values for $(\theta, \dot{\psi}_r, \dot{\psi}_l, \dot{\theta})$ will have an impact on the tracking performance of the X and Y coordinates. With this in mind, the sliding

mode control law from the end of the previous *Sliding Mode Control Initialization* section is used with $\ddot{\theta}_d$ from Equation (123) to ensure the system is underactuated. The control law is then linearized about $(\theta = 0, \dot{\theta} = 0, \dot{\psi}_r = \dot{\psi}_l)$.

The gains in the linearized version of the sliding mode control are then solved for the elements of K_d and Λ so that the constant coefficients for $(\theta, \dot{\psi}_r, \dot{\psi}_l, \dot{\theta})$ match those given by Equation (122). The remaining elements of K_d and Λ are chosen so that the matrices are positive definite. The “*solve*” function of *Maple 16* was used to calculate the elements of K_d and Λ . The procedure is straight forward and is outlined in Appendix C. The control values were solved to be

$$K_d = \begin{bmatrix} 1 & 0.895 & 1 \\ 0.895 & 1 & 1 \\ 3.696 & 3.696 & 4.437 \end{bmatrix} \text{ and } \Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 53.170 \end{bmatrix}. \quad (125)$$

ψ_{rd} and ψ_{ld} are set equal to ψ_r and ψ_l respectively. The values used for $\dot{\psi}_{rd}$, $\ddot{\psi}_{rd}$, $\dot{\psi}_{ld}$ and $\ddot{\psi}_{ld}$ in the sliding mode control come from the navigational control.

Effects of the Navigational Control Constants

The gains used by the navigational control are used to adjust the tracking performance of the Segbot. The sliding mode parameters presented in Equation (125) are left the same. The initial values used in the navigational control were

$$(k_1 = 0.3, k_2 = 5, k_3 = \sqrt{k_2}, k_4 = 0, k_5 = 0). \quad (126)$$

The Segbot is initially at rest and stabilizing at $(X, Y, \phi) = (r, 0, \frac{\pi}{2})$ before it begins to track the path described by Equation (124). Ideal initial conditions would have the Segbot in motion. r was set to 1.8 [m] and ω was set to 0.2 [rad/sec] in the desired path given by Equation (124). This demonstrates some disturbance rejection. The position of the Segbot is recorded for five figure-8 revolutions in Figure 31. The navigational gains used are given by Equation (126) and the sliding mode control values are given by Equation (125).

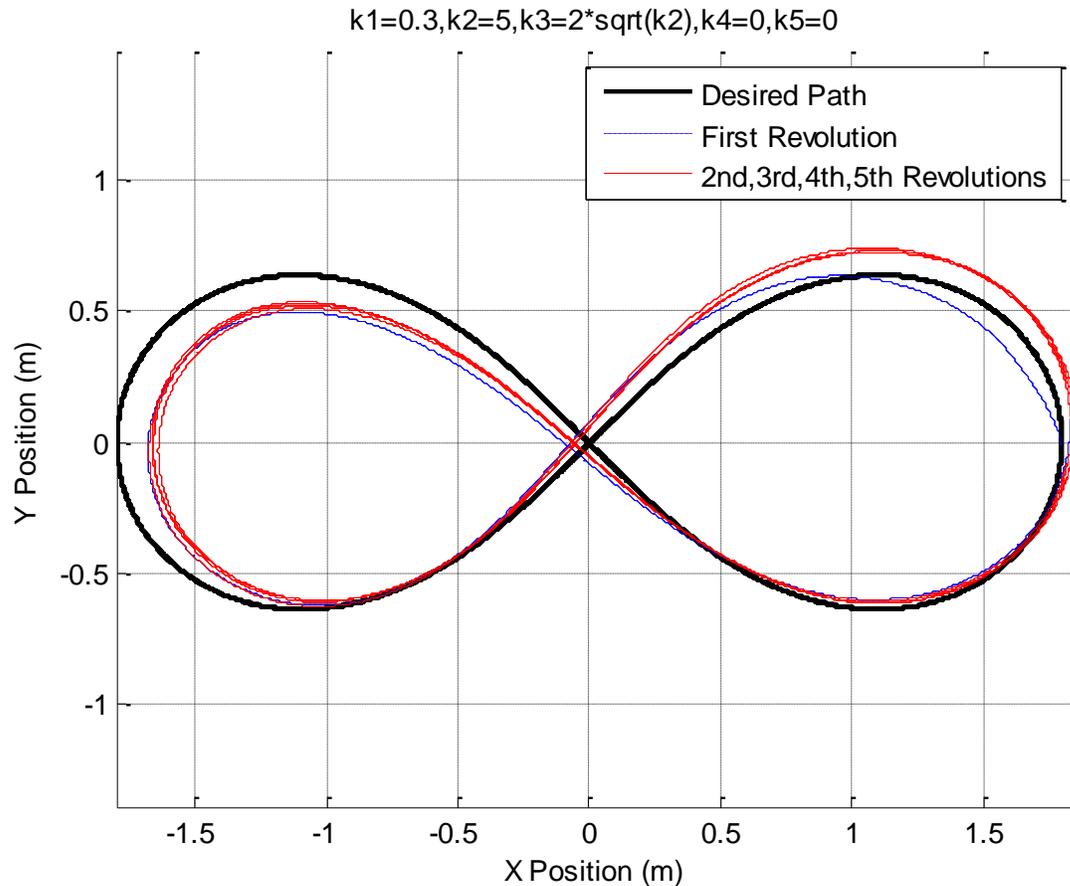


Figure 31: Segbot XY Coordinate Tracking

Figure 31 shows that the first loop around the figure-8 path differs from the remaining loops. This is because the Segbot is initially at rest and must overcome the initial conditions. The remaining loops are relatively similar to each other. This observation continued for other gains as well. The results in Figure 31 show that the Segbot is able to track the figure-8 path with success. However, the figure leaves out some crucial information because there is no indication of time vs position. To address this issue, a graph of time vs distance of the position error is provided in Figure 32.

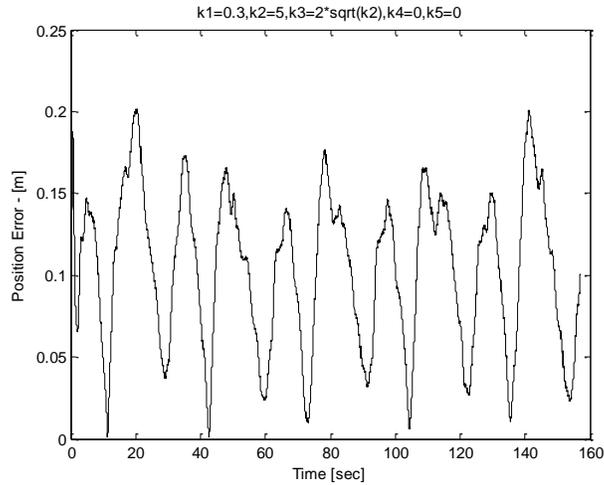


Figure 32: $k_1=0.1$ Position Absolute Value Error

Effects of k_1 in the Navigational Control

In order for the Segbot to begin tracking the desired trajectory it must first overcome adverse initial conditions. If navigational control gains are too large with adverse initial conditions, the voltage supply to the motor can become saturated and the Segbot may go unstable. However, the Segbot could begin tracking the figure-8 path with smaller navigational control gains and the gains could be made larger after the adverse initial conditions are overcome.

In order to compare the steady-state tracking of various navigational control values, the Segbot was allowed to track the figure-8 path at least two full revolutions with a given set of navigational control gains in the following figures. This allows the Segbot to reach its steady-state path for a given set of gains. The position of the Segbot was recorded for a single revolution after the Segbot had reached its steady state response for a given set of navigational control gains. Plots of the Segbot's steady state response are given in Figure 33 and Figure 34 to demonstrate the effects of modifying k_1 while the other navigational gains were left the same.

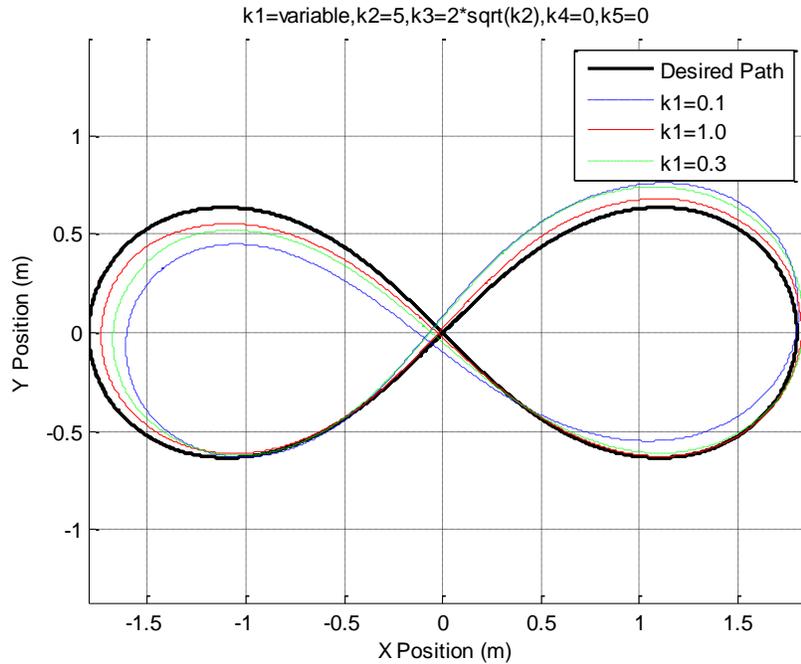


Figure 33: Steady State Position Variations for k_1

Figure 33 shows that the larger k_1 gains more precisely track the figure-8 path. However, it does not show the position error with respect to time. Larger gains for k_1 place an emphasis on the position error of the body centered drive direction, or the e_d coordinate from Figure 15. If k_1 is chosen too large then the Segbot will shoot past the desired trajectory. It also began to develop an undesirable audible chatter in the motor that cannot be observed in the graphs for large k_1 gains. This chatter was not developed for k_1 gains up to a value of about 1. A graph of time vs position error is provided in Figure 34 for the various k_1 gains.

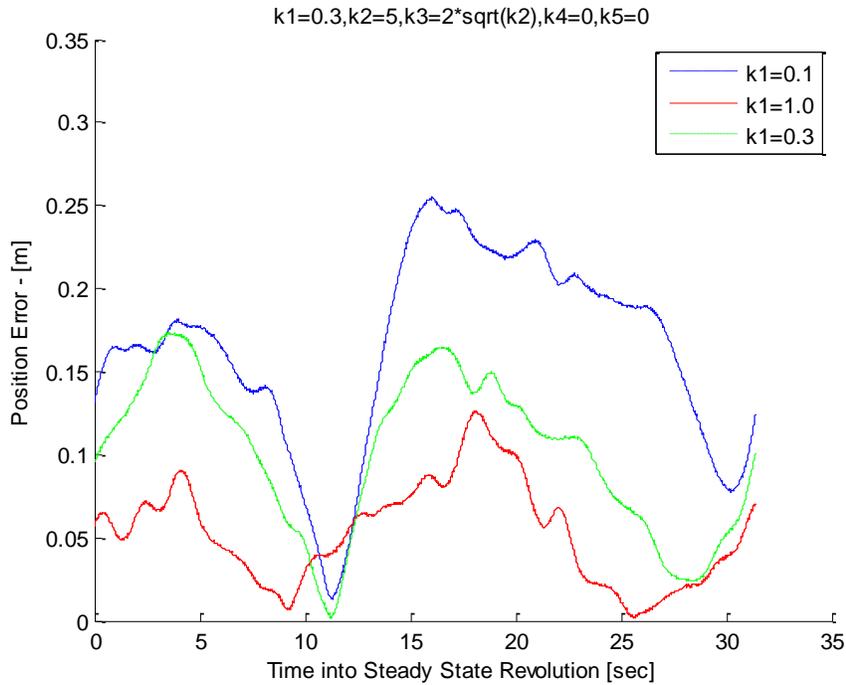


Figure 34: Steady State Time vs Absolute Value Distance Error for k_1

Figure 33 and Figure 34 show that larger gains of k_I track better with the other navigational gains chosen in Equation (126). $k_1 = 0.1$ tracks the least precisely, while $k_1 = 1$ tracks the most precisely. It should be noted that a gain of $k_1 = 1$ could not be used to overcome the initial conditions. Instead a smaller gain of k_1 had to be used at first and then k_1 could be changed to 1. This suggests that the navigational control may work well with gain scheduling, where smaller gains could be used to overcome initial conditions and higher gains could be used to increase precision after initial conditions are overcome. The results for varying k_2 (and effectively k_3), while leaving the other navigational control law values the same as those in Equation (126) follow.

Effects of k_2 in the Navigational Control

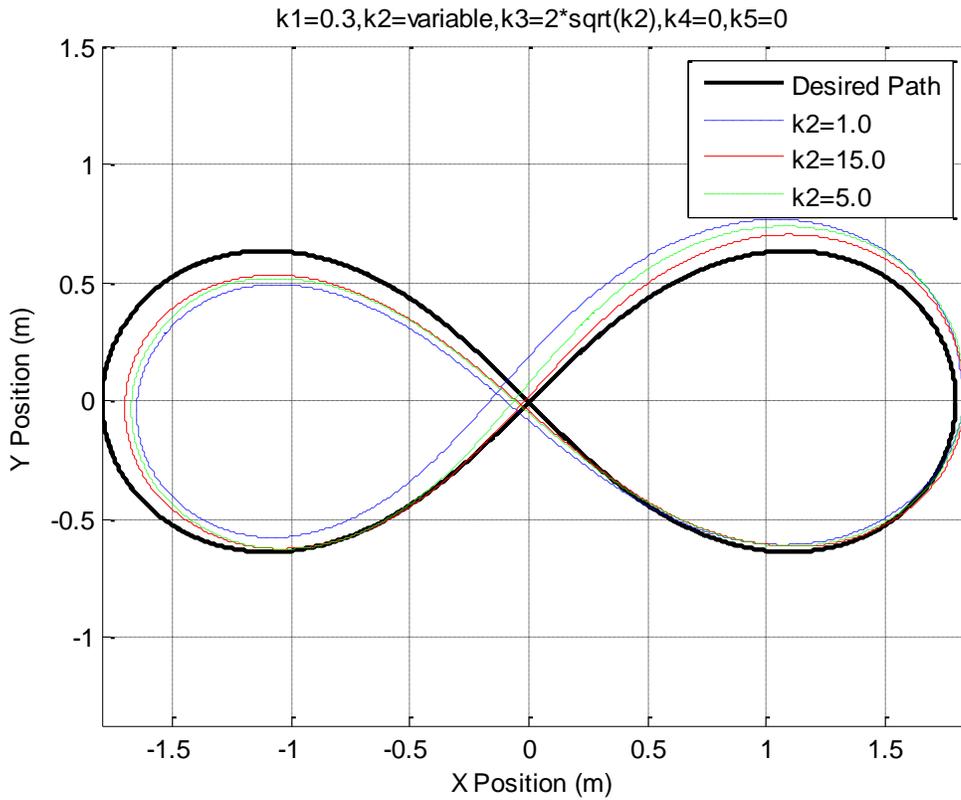


Figure 35: Steady State Position Variations for k_2

It can be inferred from Figure 35 that $k_2 = 1.0$ tracks the least precisely, while $k_2 = 15$ tracks the most precisely. A graph of time vs position error in Figure 36 will provide some new insight to this observation though.

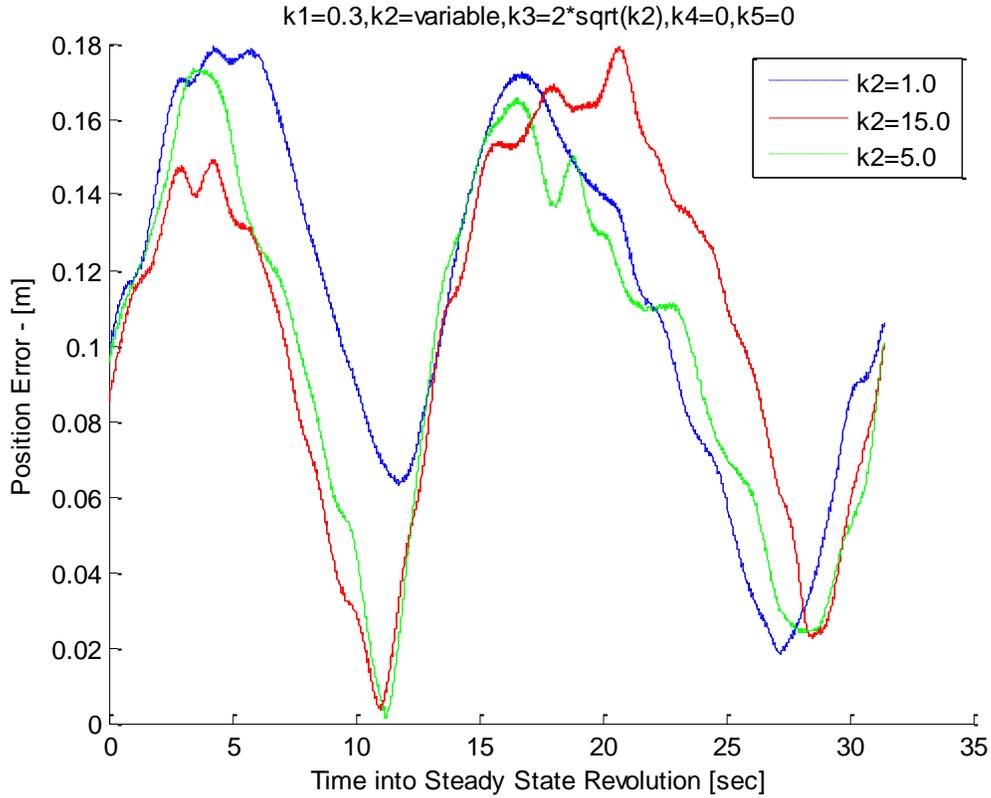


Figure 36: Steady State Time vs Absolute Value Distance Error for k_2

Figure 36 indicates that $k_2 = 15.0$ may not track much better than the smaller values for k_2 . However, changing the gain for k_2 has a profound impact on the response. As will be observed later, the best tracking results can be obtained by increasing both k_1 and k_2 .

Effects of k_4 and k_5 in the Navigational Control

k_4 and k_5 did not provide as profound of an impact during implementation when using the sliding mode control gains. It can be observed in Equation (72) that k_1 , k_2 , and k_3 are linearly related to the position error. k_4 and k_5 are cubically related to the position error. k_4 and k_5 must be chosen on the order of magnitude of 10-100 times larger than k_1 , k_2 , and k_3 to produce much of an impact when using the sliding mode control. The results of varying k_4 and k_5 are demonstrated in Figure 37 and Figure 38.

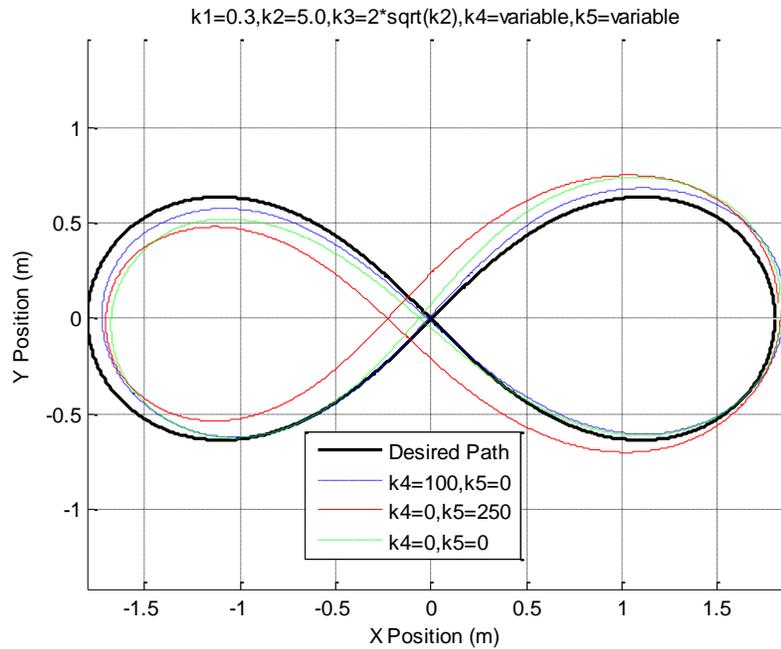


Figure 37: Steady State Position Variations for k_4 and k_5

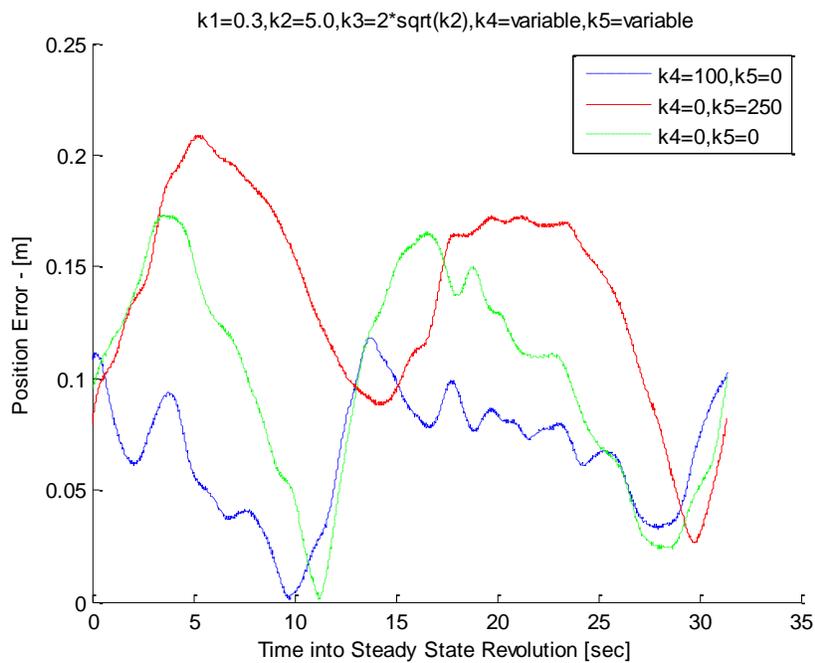


Figure 38: Steady State Time vs Distance Error for k_4 and k_5

Figure 37 and Figure 38 suggest that tracking performance can be slightly increased due to the addition of k_4 . The figures show that the addition of k_5 changes the dynamic response. However it is debatable if the addition of k_5 increases the tracking precision.

Comparison to Linear Feedback Control

The next test was to compare the implementation of the sliding mode control law derived in this thesis to linear feedback gains. The linear feedback gains used for the inner control loop are provided by Equation (122). These are the same gains that will be obtained if the sliding mode control is linearized about $(\theta = 0, \dot{\theta} = 0, \psi_r = \dot{\psi}_l)$. A variety of navigational control gains were used. Again the Segbot was allowed to make at least two revolutions and approach its steady state path for a set of navigational control gains. A final steady state loop was then recorded and the results are presented in Figure 40-Figure 46. The legend for these figures is described in Figure 39.

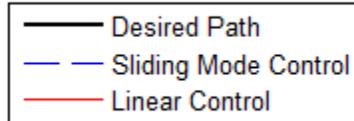


Figure 39: Legend for Figure 40-Figure 46

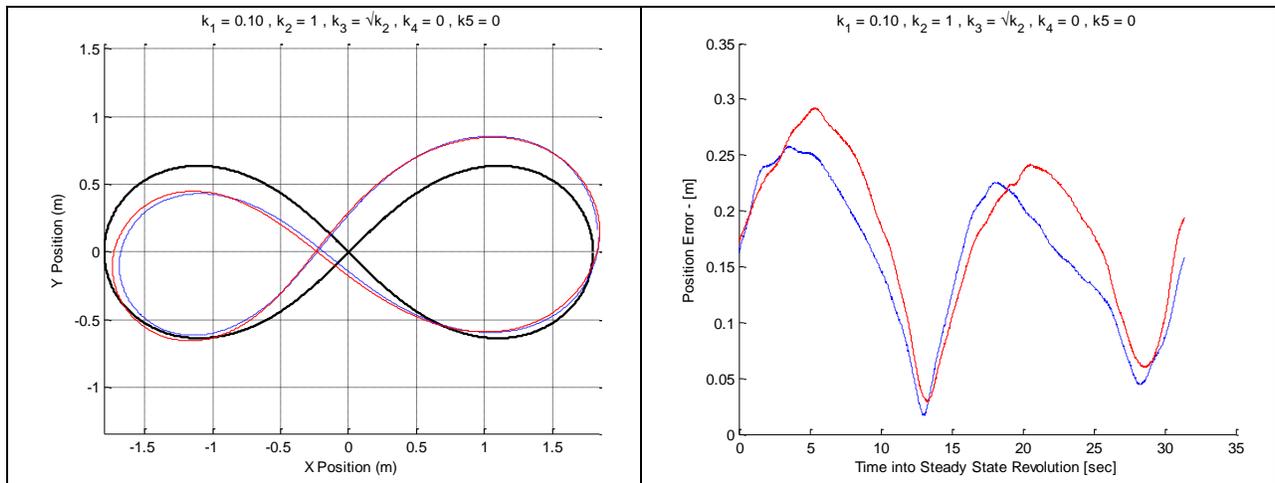


Figure 40: Steady State $k_1=0.1, k_2=1.0, k_4=0, k_5=0$

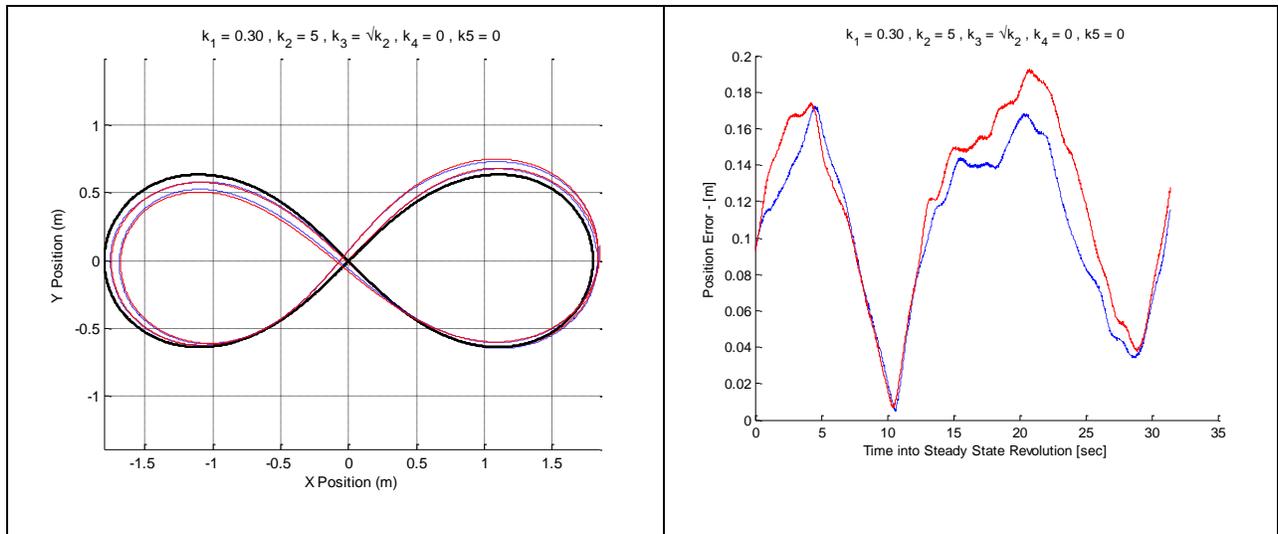


Figure 41: Steady State $k_1=0.30$, $k_2=5.0$, $k_4=0$, $k_5=0$

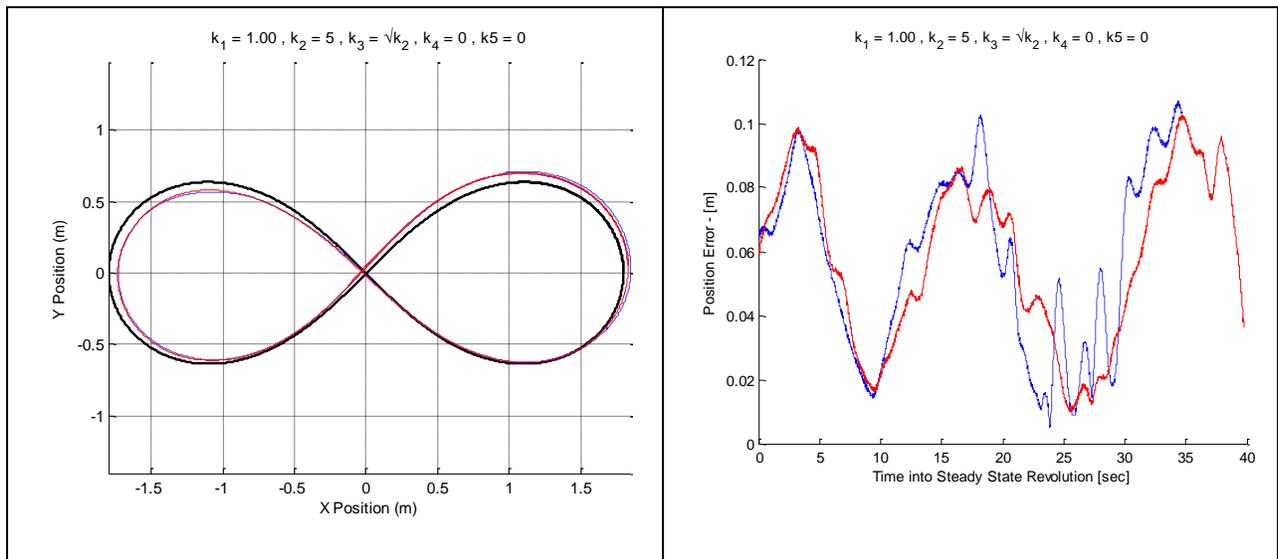


Figure 42: Steady State $k_1=1.0$, $k_2=5.0$, $k_4=0$, $k_5=0$

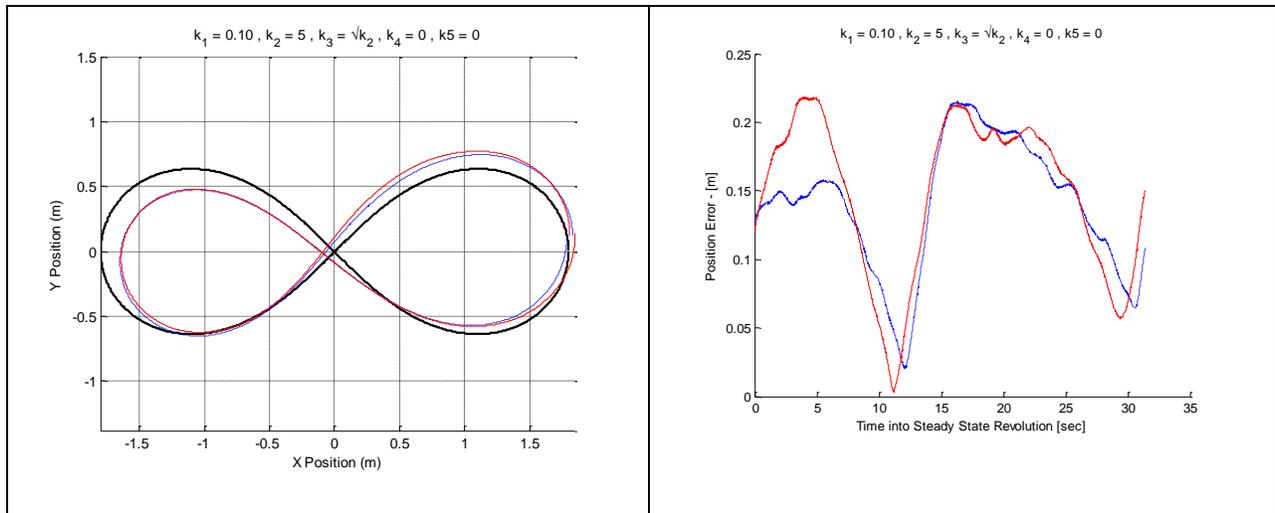


Figure 43: Steady State $k_1=0.10$, $k_2=5.0$, $k_4=0$, $k_5=0$

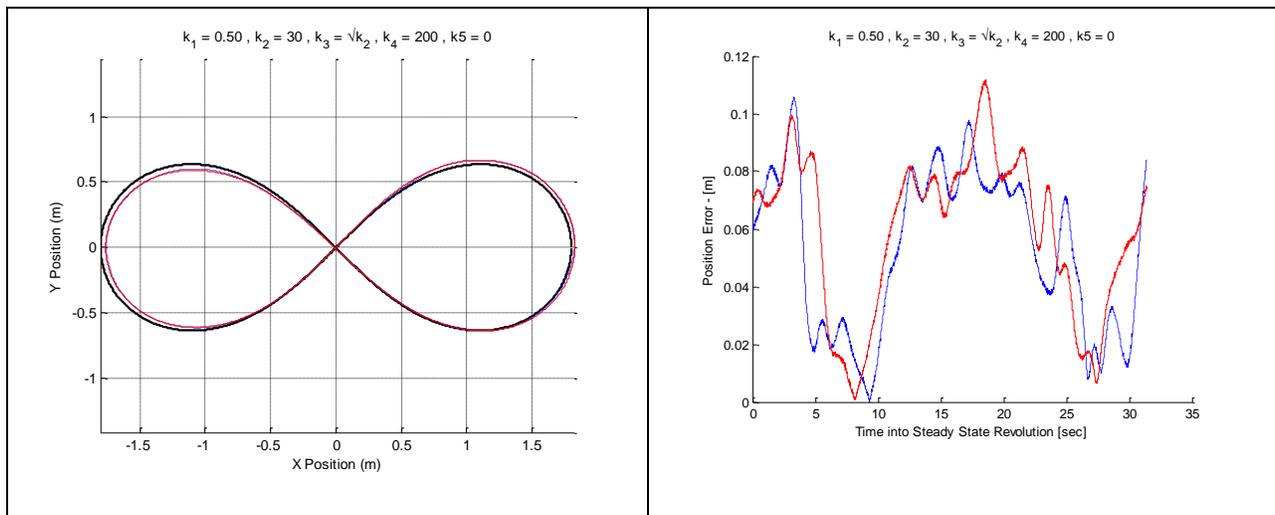


Figure 44: Steady State $k_1=0.5$, $k_2=30.0$, $k_4=200$, $k_5=0$

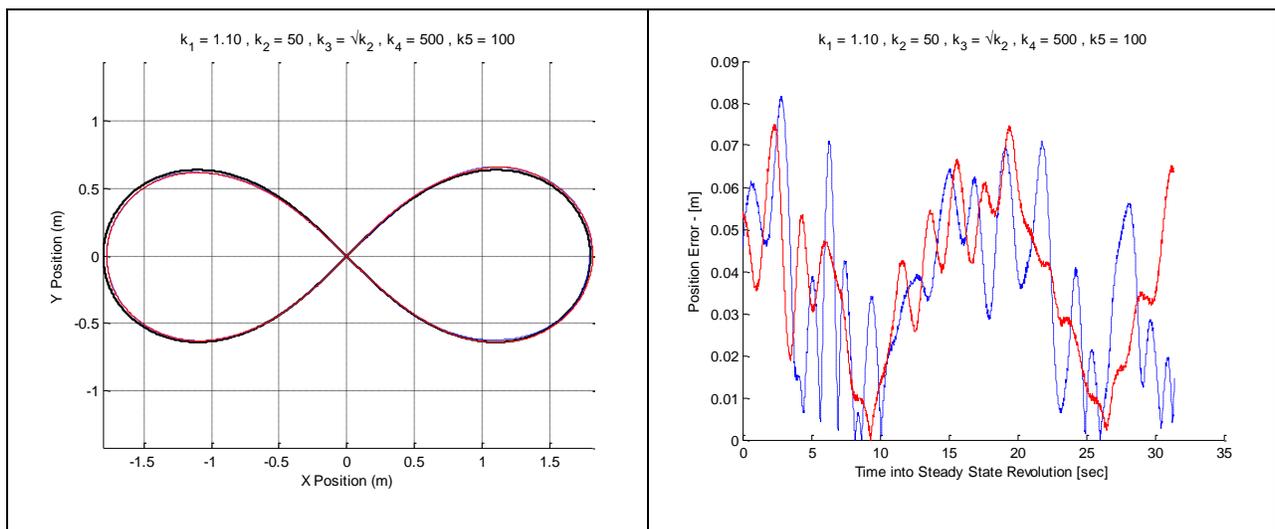


Figure 45: Steady State $k_1=1.1$, $k_2=50$, $k_4=500$, $k_5=100$

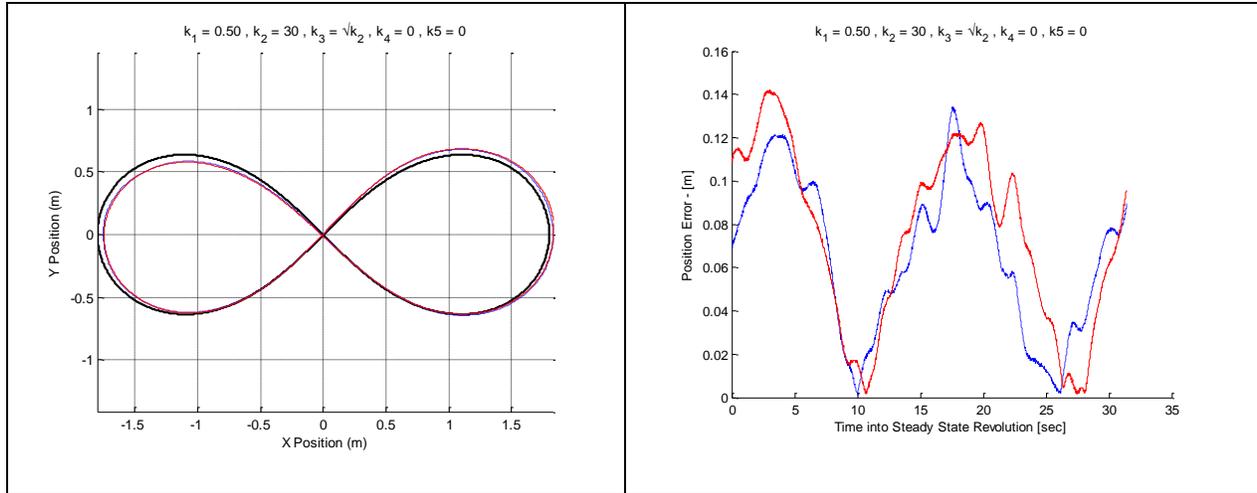


Figure 46: Steady State $k_1=0.5, k_2=30, k_4=0, k_5=0$

These figures show a few general trends. Larger k_1 and k_2 values provide more precise trajectory tracking. When chosen correctly, k_4 and k_5 provide a slight improvement on the XY tracking response.

As shown in the Segbot figure-8 tracking simulations in Figure 28 and Figure 29 from Chapter 4, using the navigational control in conjunction with the sliding mode control will not decay to a constant zero error. The goal is to minimize the tracking error, though keeping the tracking error at zero may not be feasible due to the system being underactuated. This is accomplished by adequate gain selections. Imperfections in the dynamic and electric models could also contribute to the tracking error. The surface on which the Segbot made the figure-8 path was not perfectly flat.

The sliding mode control provided a slightly smaller average trajectory tracking error than its linear counterpart for every control value presented. The gains in Figure 45 were found to provide the smallest average tracking error. In order to quantify how good a particular control works with the Segbot, the average position error for the navigational gains are provided in Table 12.

Table 12: Implementation Average Position Error

k1	k2	k3	k4	k5	Inner Loop Control	Average Error [m]
0.1	1	2	0	0	sliding mode	0.160
0.1	1	2	0	0	linear	0.181
0.5	30	10.95445	0	0	sliding mode	0.066
0.5	30	10.95445	0	0	linear	0.075
0.3	5	4.472136	0	0	sliding mode	0.106
0.3	5	4.472136	0	0	linear	0.118
1	5	4.472136	0	0	sliding mode	0.058
1	5	4.472136	0	0	linear	0.059
0.1	5	4.472136	0	0	sliding mode	0.140
0.1	5	4.472136	0	0	linear	0.148
0.5	30	10.95445	200	0	sliding mode	0.053
0.5	30	10.95445	200	0	linear	0.059
1.1	50	14.14214	500	100	sliding mode	0.037
1.1	50	14.14214	500	100	linear	0.038

Table 12 confirms that the average error for the sliding mode control provides a slightly smaller tracking error than its linear counterpart for all the navigational control gains. It is also observed that k_4 and k_5 must be much larger than k_1 , k_2 , and k_3 to have much of an impact. It can be observed that the navigational control gains have more of an impact on XY tracking performance than the choice of linear control gains or sliding mode control gains for the inner control loop. During implementation θ is always small. When θ is small, the non-linear and linear inner control loops will provide similar calculations. In order to excite the non-linear dynamics of the system, the next section will show tracking results for a desired path for θ .

Desired θ Coordinate Tracking

The previous section attempts to track the coordinates X and Y . In order to track the coordinates X and Y in the previous section, the navigational control must be used in conjunction with an inner loop control. The angle θ was relatively small. This section will attempt to excite the non-linear dynamics of the system and test the response of the sliding mode controller versus traditional linear feedback gains. The desired path to track was defined as

$$\theta_d = a \sin\left(\frac{2\pi}{T}t\right) \quad (127)$$

where θ_d is the desired tilt of the Segbot, a is the amplitude, T is the period, and t is the time. ψ_r , ψ_l , $\dot{\psi}_r$, and $\dot{\psi}_l$ were all initially set to zero. $\dot{\theta}_d$ was found by taking the time derivative of Equation (127).

Desired Angle Tracking: Control Law Calculation

There were three desired paths that were used for θ_d . The path characteristics are outlined in Table 13.

Table 13: θ_d Trajectories and Linear Control Gains

	Period [sec]	Amplitude [rad]	Linear Gains
Path1	0.8	0.2	$K = \begin{bmatrix} 0 & 0.12 & 15.0 & 0.026 & 0.14 & 0.20 \\ 0.12 & 0 & 15.0 & 0.14 & 0.026 & 0.20 \end{bmatrix}$
Path2	0.6	0.25	$K = \begin{bmatrix} 0 & 0.12 & 15.0 & 0.026 & 0.14 & 0.40 \\ 0.12 & 0 & 15.0 & 0.14 & 0.026 & 0.40 \end{bmatrix}$
Path3	0.625	0.325	$K = \begin{bmatrix} 0 & 0.12 & 15.0 & 0.026 & 0.14 & 0.60 \\ 0.12 & 0 & 15.0 & 0.14 & 0.026 & 0.60 \end{bmatrix}$

The first step was to adjust the linear feedback gains from Equation (122) to find the linear gains that work the best for each desired path. The feedback gains that were used for each path are also given in Table 13.

The gains from the linearized version of the sliding mode control are then solved for the elements of K_d and Λ so that the constant coefficients for $(\theta, \dot{\psi}_r, \dot{\psi}_l, \dot{\theta})$ match those given in Table 13. The remaining elements of K_d and Λ are chosen so that the matrices are positive definite. The “*solve*” function in *Maple 16* was used to calculate the elements of K_d and Λ . The procedure is straight forward and is outlined in Appendix C. The sliding mode control values are given in Figure 47.

	Kd	Lambda
Path1	$\begin{bmatrix} 1 & 0.888 & 1 \\ 0.888 & 1 & 1 \\ 3.670 & 3.670 & 4.491 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 74.161 \end{bmatrix}$
Path2	$\begin{bmatrix} 1 & 0.888 & 1 \\ 0.888 & 1 & 1 \\ 3.670 & 3.670 & 5.573 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 37.081 \end{bmatrix}$
Path3	$\begin{bmatrix} 1 & 0.888 & 1 \\ 0.888 & 1 & 1 \\ 3.670 & 3.670 & 6.655 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 24.720 \end{bmatrix}$

Figure 47: θ_d Path Tracking Sliding Mode Control Gains

Desired Angle Tracking: Implementation Comparison

The Segbot was allowed to initially stabilize at $\theta = 0$ before beginning to follow the desired path. The response for the first 10 periods of each path are presented in Figure 49-Figure 51. Table 14 summarizes the average phase shift, average error, and average absolute angle error for the linear and non-linear response of each specified path. Figure 48 shows the legend used for Figure 49-Figure 51.

	Desired Path
	Sliding Mode Control
	Linear Control

Figure 48: Legend for Figure 49-Figure 51

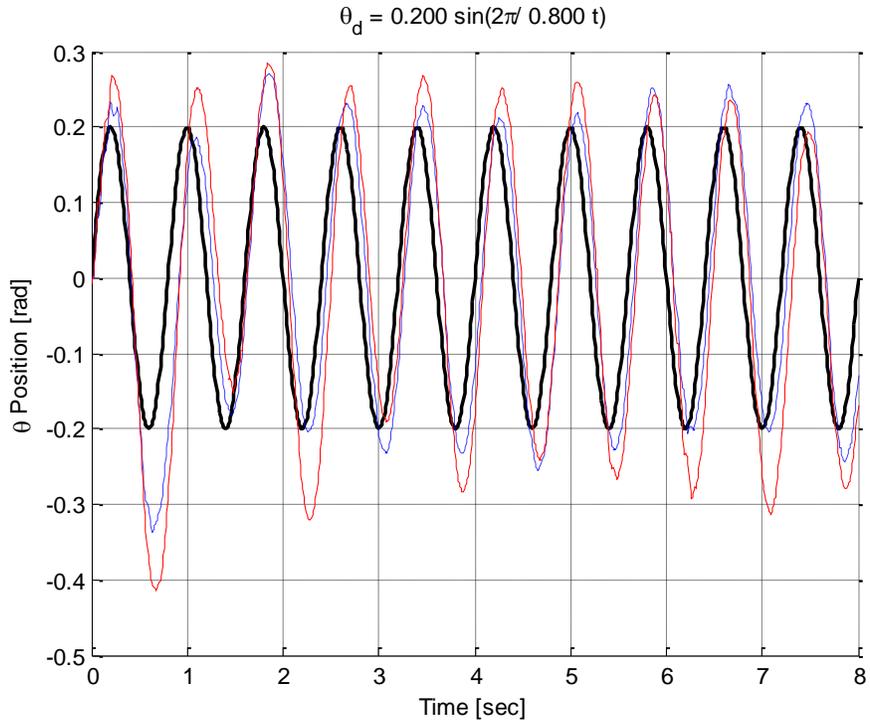


Figure 49: Desired Angle Response for Path 1

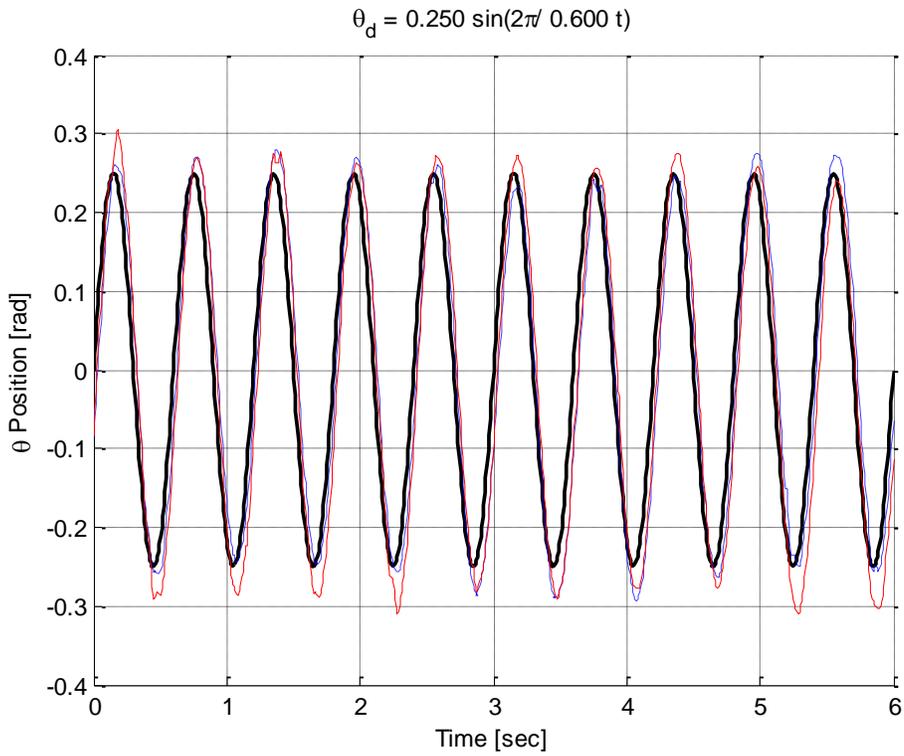


Figure 50: Desired Angle Response for Path 2

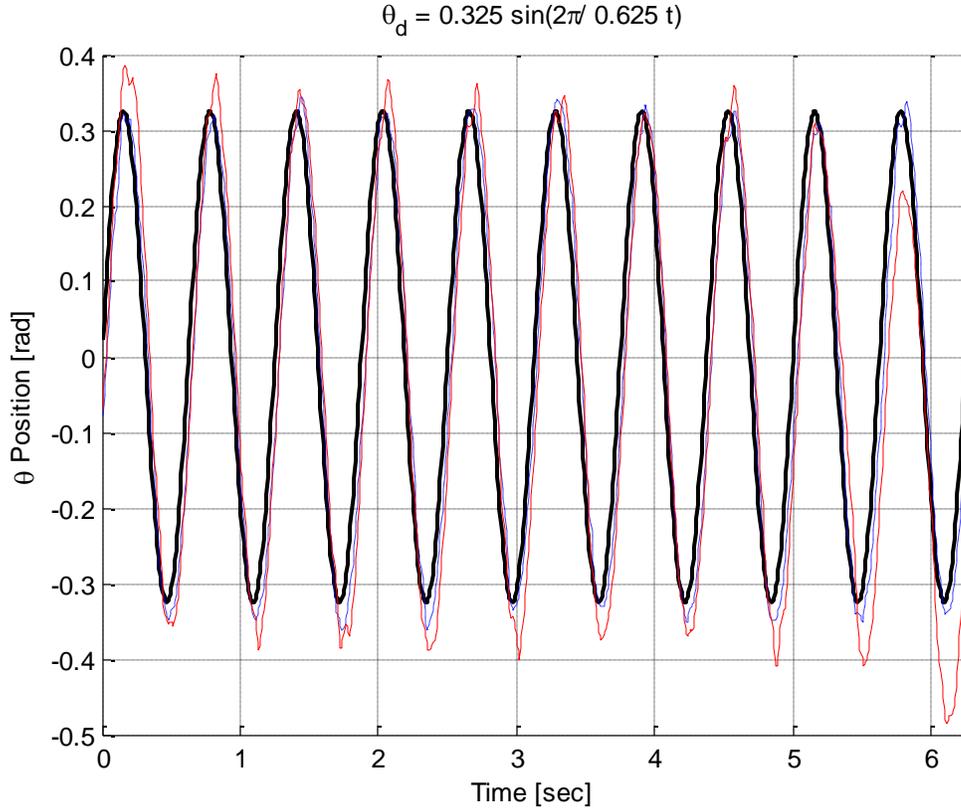


Figure 51: Desired Angle Response for Path 3

The average phase lag and average overshoot for 50 periods of θ_d for each path are provided in Table 14. The average absolute angle error over 50 periods is also provided for each response.

Table 14: Response Characteristics for θ_d Tracking

	Inner Control Loop	Average Phase Lag [sec]	Average Overshoot [rad]	Average Error [rad]
Path1	sliding mode	0.062	0.031	0.075
	linear	0.068	0.054	0.095
Path2	sliding mode	0.024	0.013	0.047
	linear	0.028	0.028	0.056
Path3	sliding mode	0.018	0.010	0.056
	linear	0.030	0.057	0.085

Table 14 suggests that the sliding mode control response is substantially better than its linear counterpart for each desired path. The sliding mode control has a smaller phase lag, a smaller average overshoot, and smaller average error for each desired path.

Implementation Conclusions

The results from the implementation confirm the validity of the sliding mode control and the navigational control presented in this thesis. The results indicate that the gains used in the

sliding mode control can be calculated by linearizing the non-linear control and matching it to a linear control. The linear control can be found using *lqr* or other techniques.

For tracking the trajectory of a given *XY* path the sliding mode control provided slightly more precise tracking than its linear counterpart. Given the dynamics of a system and a working linear control, the sliding mode control can be calculated with relatively little effort. However, the tracking improvements that the sliding mode control provides could be critical in some situations.

Choosing the control gains in the navigational control has a large effect on the *XY* tracking performance. The effect was much larger than the choice of using the sliding mode control or its linear counterpart for the inner control loop. While tracking the desired *XY* path the angle of the pendulum body was always very small. This will make the calculations provided by the sliding mode control and its linear counterpart very similar. To observe if the sliding mode control provided significant advantages over its linear counterpart when dynamic non-linearities were excited it was decided to track a path for θ .

The sliding mode control provided much more desirable tracking of θ than its linear counterpart. The phase lag, overshoot, and average error were smaller while using the sliding mode control for every path of θ_d .

Suggestions for future implementations include:

- Include an op-amp for current control instead of back calculating the voltage applied to the motors
- Implement DC motors that use belts instead of a gearbox. This will help with backlash
- Include a compass or magnetometer to help provide a more accurate estimate of the heading direction over an extended period of time
- Include GPS or a camera set-up for a mock GPS positioning system

Chapter 6: Conclusion and Recommendations

Conclusion

A new underactuated sliding mode control was presented. The key idea is that m coordinate histories have their position, velocity, and acceleration trajectories totally specified. This allows the control to work on a variety of underactuated dynamic systems. In order to ensure the control is underactuated, the remaining $n-m$ coordinate accelerations are found so that the control is underactuated. To ensure that these coordinates stay stable, the remaining $n-m$ desired position and velocities, along with control values, are appropriately chosen. The control was shown through simulations to be robust. After stable control values are determined, the control can easily be assimilated with other control techniques. It was shown that an adaptive parameter estimator term can be easily implemented into the technique.

Through simulations it was shown that the control can emphasize more precise tracking on a specific generalized coordinate. A tracking emphasis can be placed on any coordinate. The proposed underactuated sliding mode control was shown to increase the sphere of attraction compared to linear methods.

A navigational control was presented to use in conjunction with the underactuated sliding mode control, so that dynamic systems containing a non-holonomic roll without slip constraint could be tracked. The navigational control does not physically use actuators in implementation, but acts as a trajectory planning algorithm for the desired coordinates used by the inner control loop. The navigational control can be completed in a microprocessor in implementation. After stable gains are found for the underactuated sliding mode control, the dynamic response can be adjusted simply by tuning the values in the navigational control. It was found that precision tracking could be performed on the desired XY path.

The sliding mode control is not limited to the proposed navigational control. Non-holonomic control that uses kinematics (as opposed to applied forces or torques) is a research topic of its own. The implementation of other navigational control laws with the proposed underactuated sliding mode control is an activity that could be performed in future research.

The control derived in this thesis was then implemented on the Segbot. The dynamic system is underactuated, has multiple inputs, multiple outputs, and is non-holonomic. These characteristics make it a very good model to test the control laws on. The sliding mode control law values for K_d and Λ in the implementation were found by matching a linearized version of the non-linear control law to linear control values.

When the sliding mode control was implemented as the inner control loop on a desired XY path, slight increases in performance were obtained when compared to using the corresponding linear control gains in the inner control loop. During implementation it was found that tracking performance could be adjusted simply by changing control law values in the navigational control while leaving the sliding mode control at a set value. In general, higher gains in the navigational control yield more precise trajectory tracking, but require more ideal initial conditions.

It was found that the Segbot could overcome adverse initial conditions with smaller navigational control gains. Once the Segbot began to track the path, the navigational control law values could be made larger to increase the tracking performance. This shows that a simple gain scheduling in time scheme on the navigational control can provide a robust and precise trajectory tracking implementation.

The next desired trajectory was for the desired angle at which the Segbot body leans. This path helped to excite the non-linearities of the dynamic system. The sliding mode control showed a dramatic increase in performance compared to its linear counterpart for this test.

Given a working linear control law and the system dynamics, the sliding mode control law can be determined with relatively little effort. A small performance boost was provided by the sliding mode control over linear feedback gains when non-linearities in the Segbot were not excited. The sliding mode control demonstrated slightly better tracking performance than its linear counterpart for tracking desired XY coordinates. Non-linearities in the dynamics could be excited by following a desired angle for the pendulum body to lean at. The performance boost provided by the sliding mode control was dramatic for this case.

Recommendations

The research done in this work was specific to a few underactuated systems. Implementation on other underactuated dynamic systems could improve the effectiveness of the technique.

An area of investigation is determining the remaining $n-m$ positions and velocities. Because the unspecified acceleration is determined to ensure that the control is underactuated it makes sense that this acceleration should be integrated to determine the desired position and velocity. It would be monumental if a method was found to obtain control gains so that the desired positions and velocities were ensured to stay in a stable range after calculating the desired acceleration.

The simulations investigated have no noise. In actual systems, the feedback from sensors will have noise. This could have an effect on the controller performance. Handling noise for complicated systems is an area of interest.

The main focus of this thesis was the underactuated sliding mode control law. In order to work with non-holonomic systems, the navigational control was introduced to produce desired trajectories for the sliding mode control. The control developed by Kanayama et al. (1990) was shown to work well for the navigational control purposes. The navigational control simply uses kinematic constraints to calculate desired trajectories. This is a broad research area and investigating different navigational control schemes is an area for future research.

The implementation of the control law had several areas that could be improved. Implementation recommendations include implementing an op-amp for current control, implementing a belt transmission to address backlash, including a compass or magnetometer to provide a more accurate estimate of the heading direction, and including either GPS or a camera set-up for a mock GPS.

References

- Auckly, D, L. Kapitanski, and W. White (2000). Control of Nonlinear Underactuated Systems. *Comm. Pure Appl. Math*, 53(3), 354-369.
- Ashrafiuon, H., & S. R. Erwin (2004). Sliding control approach to underactuated multibody systems. *Proceeding of the American Control Conference*, (2), 1283-1288.
- Bloch, Anthony M., Naomi Ehrich Leonard, and Jerrold E. Marsden (2000). Controlled Lagrangians and the Stabilization of Mechanical Systems. I: The First Matching Theorem. *IEEE Transactions on Automatic Control*, 45(12), 2253-2270.
- Cheng, L., Cao, L., Wu, H., Zhu, Q., Xu, W., & Liu, L. (2006) Trajectory Tracking Control of Nonholonomic Mobile Robots by Backstepping . *Proceedings of 2011 International Conference on Modelling, Identification and Control*, 134-139.
- Chernousko, F. L. (2008). *Control of Nonlinear Dynamical Systems*. Berlin: Springer-Verlag.
- Difiore, D. C. (2009). *Sliding Mode Control Applied to an Underactuated Fuel Cell System*. Thesis. Rochester, Institute of Technology, Department of Mechanical Engineering
- Fossen, T. I. (1994). *Guidance and Control of Ocean Vehicles*. Chichester: Wiley.
- Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2010). *Feedback Control of Dynamic Systems* (6th ed.). Reading, Mass.: Addison-Wesley.
- Freeman, R. A., & Kokotović, P. V. (1996). *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Boston: Birkhäuser.
- Greenwood, Donald T. (2003). *Advanced Dynamics*. Cambridge UK: Cambridge Univ. Press.
- Gyroscopes and Accelerometers on a Chip. (2013, March 31). . Retrieved , from <http://www.geekmomprojects.com/gyroscopes-and-accelerometers-on-a-chip/>
- Hedrick, J. K., & Girard, A. (2010). Control of Nonlinear Dynamic Systems Spring 2010. Retrieved from <http://www.me.berkeley.edu/ME237/notes2.html>
- Henson, M., & Seborg, D. (1997). Non Linear Process Control. *Upper Saddle River: Prentice Hall*.
- Hongrui, W., Yantao, T., Siyan, F., & Zhen, S. Nonlinear Control for Output Regulation of Ball and Plate System. *Proceedings of the 27th Chinese Control Conference*, 382-387.
- Isobe, D., Yamanaka, K., & Kitamura, Y. (2008). Inverse Dynamics Calculation of Underactuated Link Systems Using Parallel Solution Scheme. *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE* , 1632 - 1637 .

- Kanayama, Yutaka, and et al (1990). A Stable Tracking Control Method for an Autonomous Mobile Robot. *IEEE International Conference on Robotics and Automation*: 384-389.
- Kansas State University- Mechanical and Nuclear Engineering (2012). http://www.mne.ksu.edu/static/nlc/tiki-index.php?page=Stabilization_Holonomic
- Khalil, Hassan K. (2002). *Nonlinear Systems*. Upper Saddle River, NJ: Prentice-Hall.
- Kim, W., Shin, J., & Lee, J. (2002). Sliding Mode Control for a Robot Manipulator with Passive Joints. *ICASE: The Institute of Control, Automation and Systems Engineer, KOREA*, 4(1), 78-83.
- Nightingale, J. (2012). *GEOMETRIC ANALYSIS AND CONTROL OF UNDERACTUATED MECHANICAL SYSTEMS*. Dissertation. Notre Dame, Aerospace and Mechanical Engineering and Mathematics
- NLCLab.mne.ksu.edu (2007). <<http://www.mne.ksu.edu/research/laboratories/non-linearcontrols-lab/non-linear-home>>.
- Olfati-Saber, R. (2001). *Nonlinear Control of Underactuated Systems with Application to Robotics and Aerospace Vehicles*. Doctoral disertation, Massachusetts Institute of Technology, Department of Electrical and Computer Science.
- Olfati-Saber, R. (2002). Normal forms for underactuated mechanical systems with symmetry. *IEEE Transactions on Automatic Control*, 47(2), 305-308.
- Ortega, R., Spong, M. W., Gómez-Estern, F., & Blankenstein, G. (2002). Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment. *IEEE Transactions on Automatic Control*, 47(8), 1218-1233.
- Patenaude, J. (2008). *A DIRECT LYAPUNOV APPROACH TO STABILIZATION AND TRACKING OF UNDERACTUATED MECHANICAL SYSTEMS*. Thesis. Kansas State University, Department of Mechanical Engineering
- Schkoda, R. F. (2007). *Dynamic Inversion of Underactuated Systems Via Squaring Transformation Matrix*. Rochester, Institute of Technology, Department of Mechanical Engineering
- Schoonwinkel, A. (1987). *Design and test of a computer stabilized unicycle*. Dissertation. Stanford University, Department of Aeronautics and Astronautics
- Slotine, J. J., & Li, W. (1991). *Applied Nonlinear Control*. Englewood Cliffs, N.J: Prentice Hall.
- Slotine, J. J., & Li, W. (1988). Adaptive Manipulator Control: A Case Study. *IEEE Transactions on Automatic Control*, 33(11), 995 - 1003.
- Spong, M. (2004). The Control of Underactuated Mechanical Systems. . Retrieved , from http://www.clemson.edu/ces/crb/ece496/spring2004/GroupA/index_files/C59Spong.pdf

- Spong, M. W., & Vidyasagar, M. (1989). *Robot Dynamics and Control*. New York: Wiley.
- Sunaina, K. (2013, August 1). myGimbal. . Retrieved , from <https://decibel.ni.com/content/docs/DOC-30443>
- Tedrake, Russell. 6.832 Underactuated Robotics, Spring 2009. (MIT OpenCourseWare: Massachusetts Institute of Technology), <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-832-underactuated-robotics-spring-2009> (Accessed 17 Feb, 2014). License: Creative Commons BY-NC-SA
- Qiang, Z., Zengbo, L., & Yao, C. (2008) A Back-stepping Based Trajectory Tracking Controller for a Non-chained Nonholonomic Spherical Robot. *Chinese Journal of Aeronautics*, 472-480.
- White, W. N., M. Foss, & X. Guo (2006). A Direct Lyapunov Approach for a Class of Underactuated Mechanical Systems. *Proceedings of the American Control Conference*, Minneapolis, Minnesota (pp. 103-110).
- White, W. N., M. Foss, & X. Guo (2007). A Direct Lyapunov Approach for Stabilization of Underactuated Mechanical Systems. *Proceedings of the American Control Conference*, New York, NY (pp. 4817-4822).
- White, W. N., M. Foss, J. Patenaude, X. Guo, & D. Garcia (2008). Improvements in Direct Lyapunov Stabilization of Underactuated, Mechanical Systems. *Proceedings of the American Control Conference*, Seattle, WA (pp. 2927-2932).
- Xu, J., & Lee, T. H. (2014). Design and Implementation of Integral Sliding-Mode Control on an Underactuated Two-Wheeled Mobile Robot. *IEEE Transactions on Industrial Electronics*, 61(7), 3671-3681.
- X., Yangsheng & S. Kwok-Wai Au (2004). Stabilization and Path Following of a Single Wheel Robot. *IEEE Transactions on Mechatronics*, 9, 407-419.
- Yang, F., & Wang, C. (2012). Adaptive tracking control for uncertain dynamic nonholonomic mobile robots based on visual servoing. *Journal of Control Theory and Applications*, 10, 56-63.
- Yang, J., & Kim, J. (1999). Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 15(3), 578-587.

Appendix A - Rolling Wheel

Rolling Wheel Dynamics

This section shows the Maple code used to find the dynamics of the rolling wheel.

Find the Body Centered Velocities

Wheel Translational Data: Position and Velocity

$$> r[w_x] := x + R[w] * \sin(\theta) * \sin(\phi)$$

$$r_{w_x} := x + R_w \sin(\theta) \sin(\phi)$$

$$> v[w_x] := \text{diff}(r[w_x], x) \cdot Dx + \text{diff}(r[w_x], \theta) * Dtheta \\ + \text{diff}(r[w_x], \phi) \cdot Dphi$$

$$v_{w_x} := Dx + R_w \cos(\theta) \sin(\phi) Dtheta + R_w \sin(\theta) \cos(\phi) Dphi$$

$$> r[w_y] := y - R[w] * \sin(\theta) * \cos(\phi)$$

$$r_{w_y} := y - R_w \sin(\theta) \cos(\phi)$$

$$> v[w_y] := (\text{diff}(r[w_y], y) \cdot Dy + \text{diff}(r[w_y], \theta) * Dtheta \\ + \text{diff}(r[w_y], \phi) \cdot Dphi)$$

$$v_{w_y} := Dy - R_w \cos(\theta) \cos(\phi) Dtheta + R_w \sin(\theta) \sin(\phi) Dphi$$

$$> r[w_z] := R[w] * \cos(\theta)$$

$$r_{w_z} := R_w \cos(\theta)$$

$$> v[w_z] := \text{diff}(r[w_z], \theta) * Dtheta$$

$$v_{w_z} := -R_w \sin(\theta) Dtheta$$

Transformation Matrix about x' axis

$$> Rtheta := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} :$$

Angular Velocity

$$> \omega[w] := \text{Multiply} \left(Dtheta, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right) + \text{Multiply} \left(Dpsi, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right) + Dphi \cdot \text{Multiply} \left(Rtheta, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$$

$$\omega_w := \begin{bmatrix} Dtheta \\ Dpsi + Dphi \sin(\theta) \\ Dphi \cos(\theta) \end{bmatrix}$$

$$> \omega_{wx} := \omega[w](1)$$

$$\omega_{wx} := Dtheta$$

$$> \omega_{wy} := \omega[w](2)$$

$$\omega_{wy} := Dpsi + Dphi \sin(\theta)$$

$$> \omega_{wz} := \omega[w](3)$$

$$\omega_{wz} := Dphi \cos(\theta)$$

Kinetic and Potential Energies

KE wheel

$$> \#\# \text{RotationalKws} := \frac{1}{2} \left(J_{w1} \cdot \omega_{wx}^2 + J_{w2} \cdot \omega_{wy}^2 + J_{w3} \cdot \omega_{wz}^2 \right)$$

$$Kws := \frac{1}{2} J_{w1} Dtheta^2 + \frac{1}{2} J_{w2} (Dpsi + Dphi \sin(\theta))^2 + \frac{1}{2} J_{w3} Dphi^2 \cos^2(\theta)$$

$$> \#\# \text{TranslationalKw} := m_w \cdot \frac{1}{2} \left(v[w_x]^2 + v[w_y]^2 + v[w_z]^2 \right)$$

$$\begin{aligned}
K_w := & \frac{1}{2} m_w \left((Dx + R_w \cos(\theta) \sin(\phi) D\theta \right. \\
& + R_w \sin(\theta) \cos(\phi) D\phi)^2 + (Dy \\
& - R_w \cos(\theta) \cos(\phi) D\theta + R_w \sin(\theta) \sin(\phi) D\phi)^2 + \\
& \left. R_w^2 \sin(\theta)^2 D\theta^2 \right)
\end{aligned}$$

> $KE_w := K_w + K_{ws}$

$$\begin{aligned}
KE_w := & \frac{1}{2} m_w \left((Dx + R_w \cos(\theta) \sin(\phi) D\theta \right. \\
& + R_w \sin(\theta) \cos(\phi) D\phi)^2 + (Dy \\
& - R_w \cos(\theta) \cos(\phi) D\theta + R_w \sin(\theta) \sin(\phi) D\phi)^2 + \\
& \left. R_w^2 \sin(\theta)^2 D\theta^2 \right) + \frac{1}{2} J_{w1} D\theta^2 + \frac{1}{2} J_{w2} (D\psi \\
& + D\phi \sin(\theta))^2 + \frac{1}{2} J_{w3} D\phi^2 \cos(\theta)^2
\end{aligned}$$

Potential Energy wheel

> $PE_w := R_w \cdot m_w \cdot \cos(\theta) g$

$$PE_w := R_w m_w \cos(\theta) g$$

Final Kinetic and Potential Energies

capital M is reserved for the mass matrix so I substitute lower case m here. The subscripts do not convert nice to Matlab for simulations.

some of the other variable names are also slightly changed

The wheel inertia's are modeled as a thin solid disk.

Build the Lagrangian.

>

$KE_w - PE_w :$

$subs(R_w = R, \%):subs(m_w = m, \%):$

$subs\left(J_{w1} = \frac{mR^2}{4}, \%\right):subs\left(J_{w3} = \frac{mR^2}{4}, \%\right):subs\left(J_{w2} = \frac{mR^2}{2}, \%\right):$

$L := \%$

$$L := \frac{1}{2} m \left((Dx + R \cos(\theta) \sin(\phi) Dtheta + R \sin(\theta) \cos(\phi) Dphi)^2 + (Dy - R \cos(\theta) \cos(\phi) Dtheta + R \sin(\theta) \sin(\phi) Dphi)^2 + R^2 \sin^2(\theta) Dtheta^2 + \frac{1}{8} m R^2 Dtheta^2 + \frac{1}{4} m R^2 (Dpsi + Dphi \sin(\theta))^2 + \frac{1}{8} m R^2 Dphi^2 \cos^2(\theta) - R m \cos(\theta) g \right)$$

>

Non holonomic constraint equations

$C1 := Dx - R * Dpsi * \cos(phi)$

$C1 := Dx - R Dpsi \cos(\phi)$

>

$C2 := Dy - R * Dpsi * \sin(phi)$

$C2 := Dy - R Dpsi \sin(\phi)$

Apply Lagrange's Equation to Yield 7 Equations:

Equation found w.r.t. the x coordinate. Notice that the 1st Lagrange multiplier is included.

>

$(diff(L, Dx, Dx)) * D Dx + (diff(L, Dx, Dy)) * D Dy + (diff(L, Dx, Dpsi)) * D Dpsi + (diff(L, Dx, Dtheta)) * D Dtheta + (diff(L, Dx, Dphi)) * D Dphi + (diff(L, Dx, x)) * Dx + (diff(L, Dx, y)) * Dy + (diff(L, Dx, psi)) * D Dpsi + (diff(L, Dx, theta)) * Dtheta + (diff(L, Dx, phi)) * Dphi - diff(L, x) + lambda[1]:$
 $eq1 := \%$

$$\begin{aligned}
eq1 := & m D D x + m R \cos(\theta) \sin(\phi) D D \theta \\
& + m R \sin(\theta) \cos(\phi) D D \phi + \frac{1}{2} m (-2 R \sin(\theta) \sin(\phi) D \theta \\
& + 2 R \cos(\theta) \cos(\phi) D \phi) D \theta \\
& + \frac{1}{2} m (2 R \cos(\theta) \cos(\phi) D \theta \\
& - 2 R \sin(\theta) \sin(\phi) D \phi) D \phi + \lambda_1
\end{aligned}$$

Equation found w.r.t. the y coordinate. Notice that the 2nd Lagrange multiplier is included.

>

$$\begin{aligned}
& (diff(L, Dy, Dx)) * D D x + (diff(L, Dy, Dy)) * D D y + (diff(L, Dy, \\
& Dpsi)) * D D psi + (diff(L, Dy, Dtheta)) * D D theta + (diff(L, Dy, \\
& Dphi)) * D D phi + (diff(L, Dy, Dzeta)) * D D zeta + (diff(L, Dy, \\
& Dnu)) * D D nu + (diff(L, Dy, x)) * D x + (diff(L, Dy, y)) * D y \\
& + (diff(L, Dy, psi)) * D D psi + (diff(L, Dy, theta)) * D theta \\
& + (diff(L, Dy, phi)) * D phi + (diff(L, Dy, zeta)) * D zeta \\
& + (diff(L, Dy, nu)) * D nu \\
& - diff(L, y) + lambda[2] : \\
eq2 := \%
\end{aligned}$$

$$\begin{aligned}
eq2 := & m D D y - m R \cos(\theta) \cos(\phi) D D \theta \\
& + m R \sin(\theta) \sin(\phi) D D \phi + \frac{1}{2} m (2 R \sin(\theta) \cos(\phi) D \theta \\
& + 2 R \cos(\theta) \sin(\phi) D \phi) D \theta \\
& + \frac{1}{2} m (2 R \cos(\theta) \sin(\phi) D \theta \\
& + 2 R \sin(\theta) \cos(\phi) D \phi) D \phi + \lambda_2
\end{aligned}$$

Equation found w.r.t. the ψ coordinate. Notice that both Lagrange multipliers are included as well as the applied torque.

>

$$\begin{aligned}
& (diff(L, Dpsi, Dx)) * D D x + (diff(L, Dpsi, Dy)) * D D y + (diff(L, Dpsi, \\
& Dpsi)) * D D psi + (diff(L, Dpsi, Dtheta)) * D D theta + (diff(L, \\
& Dpsi, Dphi)) * D D phi + (diff(L, Dpsi, x)) * D x + (diff(L, Dpsi, \\
& y)) * D y + (diff(L, Dpsi, psi)) * D D psi + (diff(L, Dpsi, theta)) \\
& * D theta + (diff(L, Dpsi, phi)) * D phi \\
& - diff(L, psi) - R \cdot \cos(phi) \cdot lambda[1] - R \cdot \sin(phi) \cdot lambda[2] \\
& - tau1 : \\
eq3 := \%
\end{aligned}$$

$$\begin{aligned}
eq3 := & \frac{1}{2} m R^2 DDpsi + \frac{1}{2} R^2 m \sin(\theta) DDphi \\
& + \frac{1}{2} R^2 m Dtheta \cos(\theta) Dphi - R \cos(\phi) \lambda_1 - R \sin(\phi) \lambda_2 - \tau l
\end{aligned}$$

Equation found w.r.t. the θ coordinate. Notice that the second torque is included as well.

$$\begin{aligned}
& (diff(L, Dtheta, Dx)) * DDx + (diff(L, Dtheta, Dy)) * DDy + (diff(L, \\
& Dtheta, Dpsi)) * DDpsi + (diff(L, Dtheta, Dtheta)) * DDtheta \\
& + (diff(L, Dtheta, Dphi)) * DDphi + (diff(L, Dtheta, x)) * Dx \\
& + (diff(L, Dtheta, y)) * Dy + (diff(L, Dtheta, psi)) * DDpsi \\
& + (diff(L, Dtheta, theta)) * Dtheta + (diff(L, Dtheta, phi)) * Dphi \\
& + -diff(L, theta) : \\
> eq4 := \% - tau2
\end{aligned}$$

$$\begin{aligned}
eq4 := & m R \cos(\theta) \sin(\phi) DDx - m R \cos(\theta) \cos(\phi) DDy \\
& + \left(\frac{1}{2} m (2 R^2 \cos(\theta)^2 \sin(\phi)^2 + 2 R^2 \cos(\theta)^2 \cos(\phi)^2 \right. \\
& \left. + 2 R^2 \sin(\theta)^2) + \frac{1}{4} m R^2 \right) DDtheta + \frac{1}{2} m (2 (\\
& -R \sin(\theta) \sin(\phi) Dtheta + R \cos(\theta) \cos(\phi) Dphi) \\
& R \cos(\theta) \sin(\phi) - 2 (Dx + R \cos(\theta) \sin(\phi) Dtheta \\
& + R \sin(\theta) \cos(\phi) Dphi) R \sin(\theta) \sin(\phi) \\
& - 2 (R \sin(\theta) \cos(\phi) Dtheta \\
& + R \cos(\theta) \sin(\phi) Dphi) R \cos(\theta) \cos(\phi) + 2 (Dy \\
& - R \cos(\theta) \cos(\phi) Dtheta + R \sin(\theta) \sin(\phi) Dphi) \\
& R \sin(\theta) \cos(\phi) + 4 R^2 \sin(\theta) Dtheta \cos(\theta)) Dtheta \\
& + \frac{1}{2} m (2 (R \cos(\theta) \cos(\phi) Dtheta \\
& - R \sin(\theta) \sin(\phi) Dphi) R \cos(\theta) \sin(\phi) + 2 (Dx \\
& + R \cos(\theta) \sin(\phi) Dtheta + R \sin(\theta) \cos(\phi) Dphi) \\
& R \cos(\theta) \cos(\phi) - 2 (R \cos(\theta) \sin(\phi) Dtheta \\
& + R \sin(\theta) \cos(\phi) Dphi) R \cos(\theta) \cos(\phi) + 2 (Dy \\
& - R \cos(\theta) \cos(\phi) Dtheta + R \sin(\theta) \sin(\phi) Dphi) \\
& R \cos(\theta) \sin(\phi)) Dphi - \frac{1}{2} m (2 (Dx + R \cos(\theta) \sin(\phi) Dtheta \\
& + R \sin(\theta) \cos(\phi) Dphi) (-R \sin(\theta) \sin(\phi) Dtheta \\
& + R \cos(\theta) \cos(\phi) Dphi) + 2 (Dy - R \cos(\theta) \cos(\phi) Dtheta \\
& + R \sin(\theta) \sin(\phi) Dphi) (R \sin(\theta) \cos(\phi) Dtheta \\
& + R \cos(\theta) \sin(\phi) Dphi) + 2 R^2 \sin(\theta) Dtheta^2 \cos(\theta)) \\
& - \frac{1}{2} m R^2 (Dpsi + Dphi \sin(\theta)) Dphi \cos(\theta) \\
& + \frac{1}{4} m R^2 Dphi^2 \cos(\theta) \sin(\theta) - R m \sin(\theta) g - \tau 2
\end{aligned}$$

Equation found w.r.t. the ϕ coordinate.

$$\begin{aligned}
& (\text{diff}(L, Dphi, Dx)) * DDx + (\text{diff}(L, Dphi, Dy)) * DDy + (\text{diff}(L, \\
& \quad Dphi, Dpsi)) * DDpsi + (\text{diff}(L, Dphi, Dtheta)) * DDtheta \\
& \quad + (\text{diff}(L, Dphi, Dphi)) * DDphi + (\text{diff}(L, Dphi, x)) * Dx \\
& \quad + (\text{diff}(L, Dphi, y)) * Dy + (\text{diff}(L, Dphi, psi)) * Dpsi + (\text{diff}(L, \\
& \quad Dphi, theta)) * Dtheta + (\text{diff}(L, Dphi, phi)) * Dphi + -\text{diff}(L, \\
& \quad phi) : \\
eq5 := \%
\end{aligned}$$

$$\begin{aligned}
eq5 := & m R \sin(\theta) \cos(\phi) DDx + m R \sin(\theta) \sin(\phi) DDy \\
& + \frac{1}{2} m R^2 \sin(\theta) DDpsi + \left(\frac{1}{2} m (2 R^2 \sin(\theta)^2 \cos(\phi)^2 \right. \\
& \left. + 2 R^2 \sin(\theta)^2 \sin(\phi)^2) + \frac{1}{2} m R^2 \sin(\theta)^2 + \frac{1}{4} m R^2 \cos(\theta)^2 \right) \\
& DDphi + \left(\frac{1}{2} m (2 (-R \sin(\theta) \sin(\phi) Dtheta \right. \\
& \left. + R \cos(\theta) \cos(\phi) Dphi) R \sin(\theta) \cos(\phi) + 2 (Dx \right. \\
& \left. + R \cos(\theta) \sin(\phi) Dtheta + R \sin(\theta) \cos(\phi) Dphi) \right. \\
& \left. R \cos(\theta) \cos(\phi) + 2 (R \sin(\theta) \cos(\phi) Dtheta \right. \\
& \left. + R \cos(\theta) \sin(\phi) Dphi) R \sin(\theta) \sin(\phi) + 2 (Dy \right. \\
& \left. - R \cos(\theta) \cos(\phi) Dtheta + R \sin(\theta) \sin(\phi) Dphi) \right. \\
& \left. R \cos(\theta) \sin(\phi) \right) + \frac{1}{2} m R^2 (Dpsi + Dphi \sin(\theta)) \cos(\theta) \left. \right) \\
& Dtheta + \frac{1}{2} m (2 (R \cos(\theta) \cos(\phi) Dtheta \\
& - R \sin(\theta) \sin(\phi) Dphi) R \sin(\theta) \cos(\phi) - 2 (Dx \\
& + R \cos(\theta) \sin(\phi) Dtheta + R \sin(\theta) \cos(\phi) Dphi) \\
& R \sin(\theta) \sin(\phi) + 2 (R \cos(\theta) \sin(\phi) Dtheta \\
& + R \sin(\theta) \cos(\phi) Dphi) R \sin(\theta) \sin(\phi) + 2 (Dy \\
& - R \cos(\theta) \cos(\phi) Dtheta + R \sin(\theta) \sin(\phi) Dphi) \\
& R \sin(\theta) \cos(\phi)) Dphi - \frac{1}{2} m (2 (Dx + R \cos(\theta) \sin(\phi) Dtheta \\
& + R \sin(\theta) \cos(\phi) Dphi) (R \cos(\theta) \cos(\phi) Dtheta \\
& - R \sin(\theta) \sin(\phi) Dphi) + 2 (Dy - R \cos(\theta) \cos(\phi) Dtheta \\
& + R \sin(\theta) \sin(\phi) Dphi) (R \cos(\theta) \sin(\phi) Dtheta \\
& + R \sin(\theta) \cos(\phi) Dphi))
\end{aligned}$$

Differentiate the first non-holonomic constraint

$$\text{> } eq6 := \text{diff}(C1, Dx) \cdot DDx + \text{diff}(C1, Dpsi) \cdot DDpsi + \text{diff}(C1, phi) \cdot Dphi$$

$$eq6 := DDx - R \cos(\phi) DDpsi + R Dpsi \sin(\phi) Dphi$$

Differentiate the second non-holonomic constraint

$$\text{> } eq7 := \text{diff}(C2, Dy) \cdot DDy + \text{diff}(C2, Dpsi) \cdot DDpsi + \text{diff}(C2, phi) \cdot Dphi$$

$$eq7 := DDy - R \sin(\phi) DDpsi - R Dpsi \cos(\phi) Dphi$$

Simplify to 3 Equations:

Gets rid of the non-holonomic constraints and also gets rid of x and y generalized coordinates

$$\text{> } eq12 := eq1 = 0:$$

$$\text{> } eq22 := eq2 = 0:$$

$$\text{> } \# \text{ solves for lambda1 and lambda2 } Lsol := \text{solve}(\{eq12, eq22\}, \{\lambda_1, \lambda_2\}):$$

$$\text{> } \text{eval}(eq3, Lsol) : \text{expand}(\%) : \text{simplify}(\%) : eq1n := \%$$

$$\begin{aligned} eq1n := & \frac{1}{2} m R^2 DDpsi + \frac{3}{2} R^2 m \sin(\theta) DDphi \\ & + \frac{5}{2} R^2 m Dtheta \cos(\theta) Dphi + R \cos(\phi) m DDx \\ & + R \sin(\phi) m DDy - \tau l \end{aligned}$$

$$\text{> } \text{eval}(eq4, Lsol) : \text{expand}(\%) : \text{simplify}(\%) : eq2n := \%$$

$$\begin{aligned} eq2n := & \frac{5}{4} DDtheta m R^2 + m R \cos(\theta) \sin(\phi) DDx \\ & - m R \cos(\theta) \cos(\phi) DDy - \frac{1}{2} m R^2 \cos(\theta) Dpsi Dphi \\ & - \frac{5}{4} m R^2 \sin(\theta) Dphi^2 \cos(\theta) - R m \sin(\theta) g - \tau 2 \end{aligned}$$

$$\text{> } \text{eval}(eq5, Lsol) : \text{expand}(\%) : \text{simplify}(\%) : eq3n := \%$$

$$\begin{aligned}
 eq3n := & -\frac{1}{4} m R \left(-4 \sin(\theta) \cos(\phi) DDx - 4 \sin(\theta) \sin(\phi) DDy \right. \\
 & - 2 R \sin(\theta) DDpsi - 6 DDphi R + 5 DDphi R \cos(\theta)^2 \\
 & \left. - 10 Dtheta R \sin(\theta) \cos(\theta) Dphi - 2 Dtheta R \cos(\theta) Dpsi \right)
 \end{aligned}$$

> eval(eq6, Lsol) : expand(%) : simplify(%) : eq4n := %

$$eq4n := DDx - R \cos(\phi) DDpsi + R Dpsi \sin(\phi) Dphi$$

> eval(eq7, Lsol) : expand(%) : simplify(%) : eq5n := %

$$eq5n := DDy - R \sin(\phi) DDpsi - R Dpsi \cos(\phi) Dphi$$

Eliminate time derivatives of x and y

> Csolve := solve({eq4n=0, eq5n=0, C1, C2}, {DDx, DDy, Dx, Dy}

$$\begin{aligned}
 Csolve := & \{DDx = R \cos(\phi) DDpsi - R Dpsi \sin(\phi) Dphi, DDy \\
 & = R \sin(\phi) DDpsi + R Dpsi \cos(\phi) Dphi, Dx = R Dpsi \cos(\phi), Dy \\
 & = R Dpsi \sin(\phi)\}
 \end{aligned}$$

Final equations:

> eval(eq1n, Csolve) : expand(%) : simplify(%, trig) : expand(%) : eq1f := %

$$\begin{aligned}
 eq1f := & \frac{3}{2} m R^2 DDpsi + \frac{3}{2} R^2 m \sin(\theta) DDphi \\
 & + \frac{5}{2} R^2 m Dtheta \cos(\theta) Dphi - \tau l
 \end{aligned}$$

> eval(eq2n, Csolve) : expand(%) : simplify(%, trig) : expand(%) : eq2f := %

$$\begin{aligned}
 eq2f := & \frac{5}{4} DDtheta m R^2 - \frac{3}{2} m R^2 \cos(\theta) Dpsi Dphi \\
 & - \frac{5}{4} m R^2 \sin(\theta) Dphi^2 \cos(\theta) - R m \sin(\theta) g - \tau 2
 \end{aligned}$$

> eval(eq3n, Csolve) : expand(%) : simplify(%, trig) : expand(%) : eq3f := %

$$\begin{aligned}
 eq3f := & \frac{3}{2} m R^2 \sin(\theta) DDpsi + \frac{3}{2} R^2 m DDphi \\
 & - \frac{5}{4} m R^2 DDphi \cos(\theta)^2 + \frac{5}{2} m R^2 Dtheta \sin(\theta) \cos(\theta) Dphi \\
 & + \frac{1}{2} Dtheta m R^2 \cos(\theta) Dpsi
 \end{aligned}$$

Put in Matrix Form

M

> map $\left(\begin{array}{l} eval, \\ [[coeff(eq1f, DDpsi), coeff(eq1f, DDtheta), coeff(eq1f, \\ DDphi)], \\ [coeff(eq2f, DDpsi), coeff(eq2f, DDtheta), coeff(eq2f, DDphi)], \\ [coeff(eq3f, DDpsi), coeff(eq3f, DDtheta), coeff(eq3f, DDphi)]] \end{array} \right)$

$$\begin{bmatrix} \frac{3}{2} m R^2 & 0 & \frac{3}{2} R^2 m \sin(\theta) \\ 0 & \frac{5}{4} m R^2 & 0 \\ \frac{3}{2} R^2 m \sin(\theta) & 0 & \frac{3}{2} m R^2 - \frac{5}{4} m R^2 \cos(\theta)^2 \end{bmatrix}$$

G

> $\begin{bmatrix} eq1f \\ eq2f \\ eq3f \end{bmatrix} : coeff\sim(\%, g) \cdot g$

$$\begin{bmatrix} 0 \\ -R m \sin(\theta) g \\ 0 \end{bmatrix}$$

C vector

$$> \begin{bmatrix} eq1f \\ eq2f \\ eq3f \end{bmatrix} - \begin{bmatrix} \frac{3}{2} m R^2 & 0 & \frac{3}{2} R^2 m \sin(\theta) \\ 0 & \frac{5}{4} m R^2 & 0 \\ \frac{3}{2} R^2 m \sin(\theta) & 0 & \frac{3}{2} m R^2 - \frac{5}{4} m R^2 \cos(\theta)^2 \end{bmatrix}$$

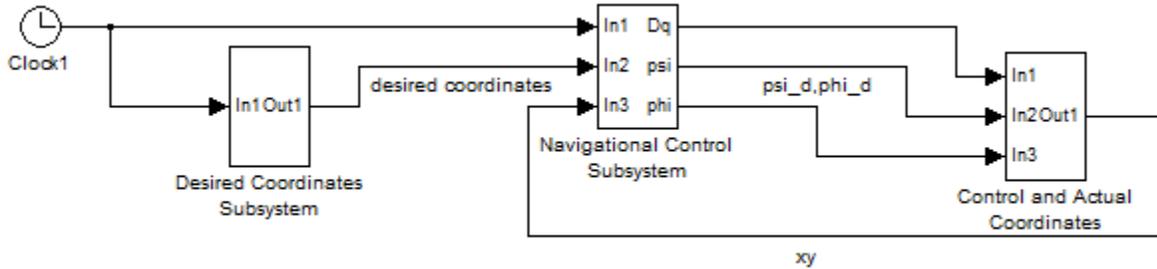
$$\cdot \begin{bmatrix} DDpsi \\ DDtheta \\ DDphi \end{bmatrix} - \begin{bmatrix} 0 \\ -R m \sin(\theta) g \\ 0 \end{bmatrix} :$$

simplify~(%)

$$\begin{bmatrix} \frac{5}{2} R^2 m Dtheta \cos(\theta) Dphi - \tau l \\ -\frac{3}{2} m R^2 \cos(\theta) Dpsi Dphi - \frac{5}{4} m R^2 \sin(\theta) Dphi^2 \cos(\theta) - \tau 2 \\ \frac{1}{2} m R^2 Dtheta \cos(\theta) (5 Dphi \sin(\theta) + Dpsi) \end{bmatrix}$$

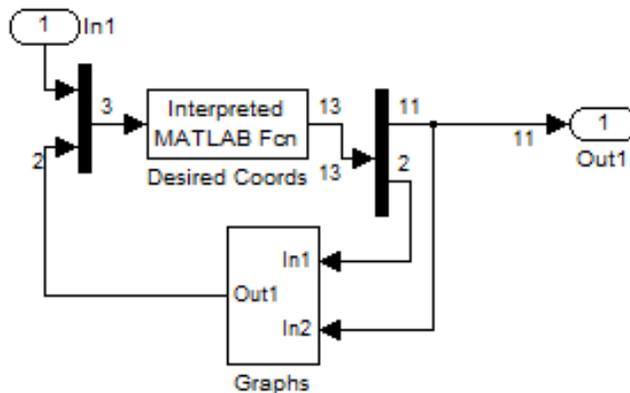
Rolling Wheel: Simulink and Matlab Files

Rolling Wheel Outer Control Loop



This is the outer Simulink diagram used for the Rolling Wheel. There are three subsystems; they are *Desired Coordinates*, *Navigational control*, and the *Control and Actual Coordinates*. The *Desired Coordinates* subsystem makes the trajectories to follow and sends it to the *Navigational Control*. The *Navigational Control* inputs the desired trajectories and the actual positions, and sends the desired translational and rotational velocities to the *Control and Actual Coordinates* subsystem. The *Control and Actual Coordinates* also inputs the actual states of the system. From these inputs the subsystem calculates the torques to apply to the dynamic system. The subsystem also applies the torque the dynamic system and calculates the state response. The *Control and Actual Coordinates* subsystem is discussed last.

Desired Coordinates Subsystem



This shows the contents of the *Desired Coordinates Subsystem*. It inputs time and the time is passed to the Matlab function *DesiredCoords2*. *DesiredCoords2* outputs the desired coordinates.

It also contains a subsystem that contains graphs to confirm the correct trajectories are obtained. However, this is just for confirmation and no necessary information is passed from it. An inside look at this subsystem is omitted. The Matlab function *DesiredCoords2* follows.

```
function Y=DesiredCoords2(u)

DDtheta0 = 0;
Dtheta0 = 0;

R=.5;    %Wheel Radius [m]
g=9.81; %Gravitational Constant [m/s^2]

r = 10; % Figure-8 Width
omega = 0.5; % Figure-8 Angular Velocity

t      = u(1);
theta  = u(2);

    x=r*sin(omega*t);    %x-position
    dx=omega*r*cos(omega*t); % time derivatives of x
    ddx=-omega^2*r*sin(omega*t);
    dddx=-omega^3*r*cos(omega*t);
    ddddx=omega^4*r*sin(omega*t);
    if x==0
        y=0;
    end

% y position logic
if x>0 & ddx<0 & dx>0
    y = (1/2)*sqrt(-4*x^2-2*r^2+2*sqrt(8*r^2*x^2+r^4));
end
if x>0 & ddx<0 & dx<0
    y = -(1/2)*sqrt(-4*x^2-2*r^2+2*sqrt(8*r^2*x^2+r^4));
end
if x<0 & ddx>0 & dx<0
    y = (1/2)*sqrt(-4*x^2-2*r^2+2*sqrt(8*r^2*x^2+r^4));
end
```

```

if x<0 & ddx>0 & dx>0
    y = -(1/2)*sqrt(-4*x^2-2*r^2+2*sqrt(8*r^2*x^2+r^4));
end

% y time derivatives, use chain rule.
dy = (-4*x^3*dx-4*y^2*x*dx+2*r^2*x*dx)/(4*x^2*y+4*y^3+2*r^2*y);
ddy = (-2*(2*x*dx+2*y*dy)^2-4*x^2*dx^2-4*x^3*ddx-4*x^2*dy^2-4*y^2*dx^2-4*y^2*x*ddx-4*y^2*dy^2+2*r^2*dx^2+2*r^2*x*ddx-2*r^2*dy^2)/(4*x^2*y+4*y^3+2*r^2*y);
ddy = (-6*(2*x*dx+2*y*dy)*(2*dx^2+2*x*ddx+2*dy^2+2*y*ddy)-12*x^2*dx*ddx-4*x^3*dddx-12*x^2*dy*ddy-12*y^2*dx*ddx-4*y^2*x*dddx-12*y^2*dy*ddy+6*r^2*dx*ddx+2*r^2*x*dddx-6*r^2*dy*ddy)/(4*x^2*y+4*y^3+2*r^2*y);
ddddy = (-6*(2*dx^2+2*x*ddx+2*dy^2+2*y*ddy)^2-(8*(2*x*dx+2*y*dy)*(6*dx*ddx+2*x*dddx+6*dy*ddy+2*y*dddy)-12*x^2*ddx^2-16*x^2*dx*dddx-4*x^3*ddddd-12*x^2*ddy^2-16*x^2*dy*dddy-12*y^2*ddx^2-16*y^2*dx*dddx-4*y^2*x*ddddd-12*y^2*ddy^2-16*y^2*dy*dddy+6*r^2*ddx^2+8*r^2*dx*dddx+2*r^2*x*ddddd-6*r^2*ddy^2-8*r^2*dy*dddy)/(4*x^2*y+4*y^3+2*r^2*y);

% Ensure everything works at first
if t<0.01
    dy = 5;
    ddy=0;
    dddy=-16.25;
    ddddy=0;
end

% Ensure figure-8 loop iterations are ok.
for n = 1:12
    if abs(t-n*pi)<.01
        ddddy = 0;
    end
end

if t==0
    phi=45*pi/180;
    dphi=0;
    ddphi=-1.4995;
    dddphi=10*t;
    % dpsi=14.1421;
    ddpsi=0;
    dddpsi=-24.7465;

```

```

else

% Heading Direction
    if dx<0 & dy>0
        phi=atan2(dy,dx)-2*pi;

    else
        phi=atan2(dy,dx);
    end

% Heading Direction Time Derivatives, use chain rule
    dphi=-(-ddy*dx+dy*ddx)/(dx^2+dy^2);
    ddphi=(ddy*dx-dy*dddx)/(dx^2+dy^2)-2*(-
dy*ddx+ddy*dx)*(dx*ddx+dy*ddy)/(dx^2+dy^2)^2;

for n = 1:12
    if abs(t-(2*n-1)*pi)<.01
        ddphi = 0;
    end
end

    dddphi=- (dy*ddddx*dx^4-3*ddy*dddx*dy^4+3*ddy*dddx*dx^4-
3*ddy*ddx*dy^4+3*ddy*ddx*dx^4-ddddy*dx*dy^4-
2*ddddy*dx^3*dy^2+2*dy^3*ddddx*dx^2+6*dy^3*ddx*ddy^2+6*dy*ddx^3*dx^2-
6*ddy^3*dx*dy^2-6*ddy*dx^3*ddx^2+18*ddy*dx*ddx^2*dy^2-18*dy*ddx*ddy^2*dx^2-
6*dy^3*dddx*dx*ddx-6*dy*dddx*dx^3*ddx+6*ddy*dx*dy^3*ddy+6*ddy*dx^3*dy*ddy-
2*dy^3*ddx^3+2*ddy^3*dx^3-ddddy*dx^5+dy^5*ddddx)/(dx^2+dy^2)^3;

for n = 1:12
    if abs(t-(n)*pi)<.02
        dddphi = 0;
    end
end

% Translational Components to Angular Wheel Components
dpsi=(dx*cos(phi)+dy*sin(phi))/R;
ddpsi = (ddx*cos(phi)-dx*sin(phi)*dphi+ddy*sin(phi)+dy*cos(phi)*dphi)/R;

```

```

    dddpsi = (dddxd*cos(phi)-2*ddx*sin(phi)*dphi-
dx*cos(phi)*dphi^2+dddy*sin(phi)+2*ddy*cos(phi)*dphi-dy*sin(phi)*dphi^2)/R;
for n = 1:12
    if abs(t-(2*n-1)*pi)<.005
        dddpsi = .9625;
    end
end

end

end

% Check that uses inverses dynamics to ensure that theta stays stable while
%% following the path. Not very important.

dtheta = Dtheta0+(-0.6e1 * sin(theta) * ddpsi - (6 * ddphi) + 0.5e1 * ddphi *
cos(theta) ^ 2) / cos(theta) / (dpsi + 0.5e1 * sin(theta) * dphi) / 0.2e1;

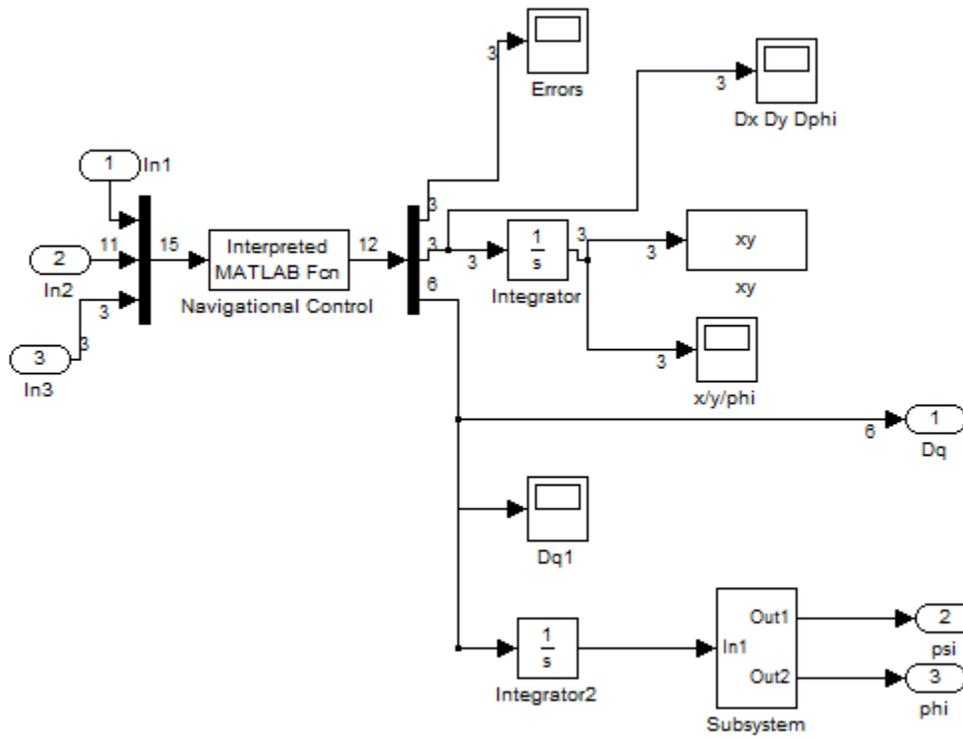
ddtheta = DDtheta0 + dtheta * ((-0.6e1 * cos(theta) * ddpsi - 0.10e2 *
ddphi * cos(theta) * sin(theta)) / cos(theta) / (dpsi + 0.5e1 * sin(theta) *
dphi) / 0.2e1 + (-0.6e1 * sin(theta) * ddpsi - 0.6e1 * ddphi + 0.5e1 * ddphi
* cos(theta) ^ 2) / cos(theta) ^ 2 / (dpsi + 0.5e1 * sin(theta) * dphi) *
sin(theta) / 0.2e1 - 0.5e1 / 0.2e1 * (-0.6e1 * sin(theta) * ddpsi - 0.6e1 *
ddphi + 0.5e1 * ddphi * cos(theta) ^ 2) / (dpsi + 0.5e1 * sin(theta) * dphi)
^ 2 * dphi) - 0.3e1 * dddpsi * sin(theta) / cos(theta) / (dpsi + 0.5e1 *
sin(theta) * dphi) + dddphi * (-0.6e1 + 0.5e1 * cos(theta) ^ 2) / cos(theta)
/ (dpsi + 0.5e1 * sin(theta) * dphi) / 0.2e1 - ddpsi * (-0.6e1 * sin(theta) *
ddpsi - 0.6e1 * ddphi + 0.5e1 * ddphi * cos(theta) ^ 2) / cos(theta) / (dpsi
+ 0.5e1 * sin(theta) * dphi) ^ 2 / 0.2e1 - 0.5e1 / 0.2e1 * ddphi * (-0.6e1 *
sin(theta) * ddpsi - 0.6e1 * ddphi + 0.5e1 * ddphi * cos(theta) ^ 2) /
cos(theta) / (dpsi + 0.5e1 * sin(theta) * dphi) ^ 2 * sin(theta);

% Output Array
Y = [x y dx dy phi dphi ddphi dddphi dpsi ddpsi dddpsi dtheta ddtheta]';

```

The *Navigational Control Subsystem* is discussed next.

Navigational Control Subsystem



This shows the *Navigational Control* subsystem. The first input is the simulation time. It is not used. The second input contains an array that has the desired positions and comes from the *Desired Coordinates* subsystem. The third input contains the actual states. These come from the *Control and Actual Coordinates* subsystem.

The first output, labeled Dq , contains the desired translational and angular velocity. This is the only output that is used by the inner control loop.

The 2nd and 3rd outputs contain an estimate of the wheel angles and heading direction. However, actual wheel angles and the heading direction used in the control come from the *Control and Actual Coordinates* subsystem. The 2nd and 3rd outputs are not used by other subsystems and could be omitted entirely.

The main part of the subsystem is the Matlab function *Navigational_Control.m*. This takes the inputs and calculates a desired translational and angular velocity to be used by the sliding mode control, or inner control loop. The Matlab function follows:

```

function gout = Navigational_Control(u)

Rw      = 0.5;    % Wheel Radius
xd      = u(2);   %% Desired Positions and Time Derivatives: the jerk terms
                %% are not necessary
yd      = u(3);

phid    = u(6);
Dphid   = u(7);
DDphid  = u(8);
DDDphid = u(9);
Dpsid   = u(10);
DDpsid  = u(11);
DDDpsid = u(12);

%% Actual Positions: From Control and Actual Position Subsystem
x       = u(13);
y       = u(14);
phi     = u(15);

%%Constants
vd = Dpsid*Rw;    %% Desired Linear
wd = Dphid;      %% Desired Rotational

%% Global Differences
e = [xd-x yd-y phid-phi]';

% Rotation to body
R = [cos(phi) sin(phi) 0; -sin(phi) cos(phi) 0; 0 0 1];

%% Local Error: [tangent lateral phi]'
pe = R*e;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
e1 = pe(1);
e2 = pe(2);
e3 = pe(3);

```

```

%%% These are the least aggressive gains
% k1 = 2 ; %2 %first 3 First set k2: the bigger k1 is the
wider the loop while coming back in: play with k1 = 2 k2 = 0.02or0.01
% k2 = .01 ; %0.01 %% first 4 %% The smaller the gain, the larger the
curve
% k3 = .5*2*sqrt(k2) ;
% k4 = 0;k5 =0;

%% Best so far: k1 = 1, k2 = 0.1 ; k3 = 1; k4 = 0.5 ; k5 = 5

k1 = 1 ;
k2 = 0.1;
k3 = 2*sqrt(k2);
k4 = 0;k5 =0;

%% Velocity and Angular Velocity Control:
v = (vd*cos(e3)+k1*e1+ k4*e2^2);
% if abs(v)>15 %% Saturation
% v = 15;
% end
w = wd + vd*(k2*e2 +k3*sin(e3)) + k2*k5*sin(e3)*e2^2;
Dphi = w;
% if abs(Dphi)>6 %% Saturation
% Dphi = sign(Dphi)*6;
% end

% X and Y Velocity Components
Dx = v*cos(phi);
Dy = v*sin(phi);
% %%%%%%%%%%% Derivatives of error: Not Necessary
De = [Dphi * e2 - v + vd * cos(e3) -Dphi * e1 + vd * sin(e3) Dphid - Dphi];
De1 = De(1);
De2 = De(2);
De3 = De(3);
% %%%%%%%%%%% Find Linear accelaration and DDphi: Not necessary
Dvd = Rw*DDpsid; %% Desired Linear Acceleration
Dq = [Dvd*cos(e3) + De1 * k1 - De3 * vd * sin(e3) Dvd * (k2 * e2 + k3 *
sin(e3)) + DDphid + De2 * vd * k2 + De3 * vd * k3 * cos(e3)];
Dv = Dq(1); %% Linear Acceleration
% if abs(Dv)>25
% Dv = 10*sign(Dv);
% end

```

```

DDphi = Dq(2);    %% DDphi
% if abs(DDphi) > 1.5
%     DDphi = sign(DDphi)*1.5;
% end

%%% Find Second Derivatives of error: Not necessary
DDe = [DDphi * e2 - Dv + Dvd * cos(e3) + De2 * Dphi - De3 * vd * sin(e3) -
DDphi * e1 + Dvd * sin(e3) - De1 * Dphi + De3 * vd * cos(e3) -DDphi+DDphid];
DDe1 = DDe(1);    %% DDe1
DDe2 = DDe(2);
DDe3 = DDe(3);

%% Find Jerks: Not necessary
DDvd = Rw*DDDpsid;    %% Desired Linear Jerk
DDq = [-Dvd * De3 * sin(e3) + De3 * (-Dvd * sin(e3) - De3 * vd * cos(e3)) +
DDvd * cos(e3) + DDe1 * k1 - DDe3 * vd * sin(e3) Dvd * (De2 * k2 + De3 * k3 *
cos(e3)) + DDvd * (k2 * e2 + k3 * sin(e3)) + De2 * Dvd * k2 + De3 * (Dvd * k3
* cos(e3) - De3 * vd * k3 * sin(e3)) + DDe2 * vd * k2 + DDe3 * vd * k3 *
cos(e3) + DDDphid];
DDv = DDq(1);    %% Linear Jerk
% if abs(DDv) > 30
%     DDv = sign(DDv)*30;
% end

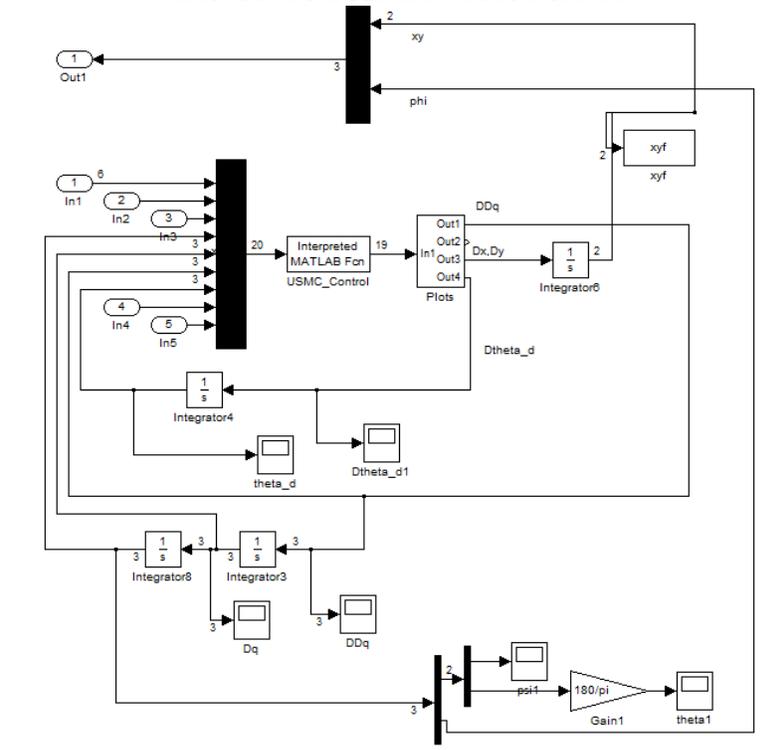
DDDphi = DDq(2);
% if abs(DDDphi)> 4
%     DDDphi=4*sign(DDDphi);
% end

Dpsi = v/Rw;
DDpsi = Dv/Rw;
DDDpsi = DDv/Rw;

% Output
qout = [e'   Dx Dy Dphi   Dpsi DDpsi DDDpsi Dphi DDphi DDDphi]';

```

Control and Actual Coordinates



This shows the sliding mode control and the actual contact position and actual heading direction. The main part of the subsystem is the Matlab function *USMC_Control_Dynamics.m*. This inputs the desired angular velocities and the actual velocities. It can be observed that the actual XY position is output back to the Navigational Control at the top of the subsystem. The actual state positions and velocities are returned to the control law on the bottom of the subsystem diagram. The main Matlab function of this subsystem calculates the applied torque, then applies the torque to the dynamic system to calculate the generalized coordinate acceleration response. The function follows.

```
function output = USMC_Control_Dynamics(u)

%% System Constants: Given in standard SI units
Rw = 0.5;      %% Wheel Radius
m=5;          % mass
R=0.5;        % wheel radius
g=9.81;       % gravitational constant

%% Desired Path: As can be observed some of the u-array elements
%% are unnecessary.
Dpsi_d       = u(1);
DDpsi_d      = u(2); %DDpsi_d=0;
DDphi_d      = u(5); %DDphi_d=0;
```

```

psi_d      = u(7);
phi_d      = u(8);

%% Actual Positions
psi        = u(9);
theta      = u(10);
phi        = u(11);

Dpsi       = u(12);
Dtheta     = u(13);
Dphi       = u(14);

%% Desired theta_d: found by integrating the calculated Dtheta_d needed
%% to ensure the control law is underactuated.
theta_d    = u(18);

%% System: Dynamic Matrices
M = [0.3e1 / 0.2e1 * m * R ^ 2 0 0.3e1 / 0.2e1 * m * R ^ 2 * sin(theta); 0
0.5e1 / 0.4e1 * m * R ^ 2 0; 0.3e1 / 0.2e1 * m * R ^ 2 * sin(theta) 0 0.3e1 /
0.2e1 * m * R ^ 2 * sin(theta) ^ 2 + m * R ^ 2 * cos(theta) ^ 2 / 0.4e1];
C = [0 0.3e1 / 0.4e1 * Dphi * cos(theta) * m * R ^ 2 0.7e1 / 0.4e1 * Dtheta *
m * R ^ 2 * cos(theta); -0.3e1 / 0.4e1 * Dphi * cos(theta) * m * R ^ 2 0 -
0.3e1 / 0.4e1 * m * R ^ 2 * cos(theta) * Dpsi - 0.5e1 / 0.4e1 * Dphi *
cos(theta) * m * R ^ 2 * sin(theta); -Dtheta * m * R ^ 2 * cos(theta) / 0.4e1
0.3e1 / 0.4e1 * m * R ^ 2 * cos(theta) * Dpsi + 0.5e1 / 0.4e1 * Dphi *
cos(theta) * m * R ^ 2 * sin(theta) 0.5e1 / 0.4e1 * Dtheta * m * R ^ 2 *
cos(theta) * sin(theta)];
G = [0 -m * g * R * sin(theta) 0]';

%%%%%%%%%First Matching%%%%%%%%% FINAL GUESS
%%%%%%%%%Control Law Constants %%%%%%%%%%
%%%%%%%%%
%% Initial Guesses %%%%%%%%%%
%%%%%%%%%
lambda1 = 1;lambda2 = 10; lambda3 = 1;
Kd = [10 -1 -1;-1 10 -1;-1 -1 10];
kd11 = Kd(1,1); kd21 = Kd(2,1); kd31=Kd(3,1);kd32 = Kd(3,2); kd33 = Kd(3,3);
%%%%%%%%%

%% Ensures underactuation:
Dtheta_d = (-0.3e1 / 0.2e1 * m * R ^ 2 * sin(theta) * (DDpsi_d - lambda1 *
(Dpsi - Dpsi_d)) - (0.3e1 / 0.2e1 * m * R ^ 2 * sin(theta) ^ 2 + m * R ^ 2 *
cos(theta) ^ 2 / 0.4e1) * (DDphi_d - lambda3 * (Dphi - Dphi_d)) + Dtheta * m
* R ^ 2 * cos(theta) * (Dpsi_d - lambda1 * (psi - psi_d)) / 0.4e1 - 0.5e1 /
0.4e1 * Dtheta * m * R ^ 2 * cos(theta) * sin(theta) * (Dphi_d - lambda3 *
(phi - phi_d)) + kd31 * (Dpsi - Dpsi_d + lambda1 * (psi - psi_d)) + kd33 *
(Dphi - Dphi_d + lambda3 * (phi - phi_d)) + 0.3e1 / 0.4e1 * m * R ^ 2 *
cos(theta) * Dpsi * lambda2 * theta - 0.3e1 / 0.4e1 * m * R ^ 2 * cos(theta)
* Dpsi * lambda2 * theta_d + 0.5e1 / 0.4e1 * Dphi * cos(theta) * m * R ^ 2 *
sin(theta) * lambda2 * theta - 0.5e1 / 0.4e1 * Dphi * cos(theta) * m * R ^ 2
* sin(theta) * lambda2 * theta_d + kd32 * Dtheta + kd32 * lambda2 * theta -
kd32 * lambda2 * theta_d) / (0.3e1 / 0.4e1 * m * R ^ 2 * cos(theta) * Dpsi +
0.5e1 / 0.4e1 * Dphi * cos(theta) * m * R ^ 2 * sin(theta) + kd32);

%% Simply step DDtheta_d to zero

```

```

DDtheta_d = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Slide Vector%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
q = [psi theta phi]';
Dq = [Dpsi Dtheta Dphi]';

%% Slide Vector:
q_d = [psi_d theta_d phi_d]';
Dq_d = [Dpsi_d Dtheta_d Dphi_d]';
DDq_d = [DDpsi_d DDtheta_d DDphi_d]';
Lam = [lambda1 0 0;0 lambda2 0;0 0 lambda3];
Dq_r = Dq_d - Lam*(q-q_d);
DDq_r = DDq_d - Lam*(Dq-Dq_d);
s = Dq - Dq_r;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculate Torques
tau = M*DDq_r + C*Dq_r + G - Kd*s;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Apply torque to wheel:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DDq = M\(-C*Dq - G + tau);
sdot = DDq - DDq_r;

%% First Matching Condition: A check
Mdot = [0 0 0.3e1 / 0.2e1 * Dtheta * m * R ^ 2 * cos(theta); 0 0 0; 0.3e1 /
0.2e1 * Dtheta * m * R ^ 2 * cos(theta) 0 0.5e1 / 0.2e1 * Dtheta * m * R ^ 2
* cos(theta) * sin(theta)];
con1=s'*(Mdot/2-C)*s;

%% Find Actual Dx and Dy
Dx = Dpsi*cos(phi)*Rw;
Dy = Dpsi*sin(phi)*Rw;

% Shows Lyapunov Function
V = .5*s'*M*s;
% Lyapunov Time Derivative
Vdot = s'*M*sdot + s'*(Mdot/2)*s;

% More Checks
check = [Vdot-s'*(-C*Dq-G+tau-M*DDq_d+M*Lam*(Dq-Dq_d))-s'*Mdot/2*s 0 0]';
check2 = [s'*(-C*Dq_r-G+tau-M*DDq_r)+s'*(Mdot/2-C)*s 0 s'*(Mdot/2-C)*s]';

% Ouput
output= [DDq' DDtheta_d Dx Dy con1 V tau' Dtheta_d check' check2' Vdot]';

```

Appendix B - Non-Holonomic Segbot

Non-Holonomic Segbot Equations of Motion

Derive Lagrangian

Kinetic Energy of the Wheels:

$$> KE_{wr} := \frac{1}{2} \cdot \left(\frac{3}{2} \cdot mw \cdot R^2 \right) \cdot Dpsi^2 + \frac{1}{2} \left(\frac{mw \cdot R^2}{4} \cdot Dphi^2 \right);$$

$$KE_{wr} := \frac{3}{4} mw R^2 Dpsi^2 + \frac{1}{8} mw R^2 Dphi^2$$

$$> KE_{wl} := \frac{1}{2} \cdot \left(\frac{3}{2} \cdot mw \cdot R^2 \right) \cdot Dpsil^2 + \frac{1}{2} \left(\frac{mw \cdot R^2}{4} \cdot Dphi^2 \right);$$

$$KE_{wl} := \frac{3}{4} mw R^2 Dpsil^2 + \frac{1}{8} mw R^2 Dphi^2$$

> # position of base center of mass w.r.t. axel center

$$rp := \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} + \begin{bmatrix} h_c \cdot \sin(\theta) \cdot \cos(\phi) \\ h_c \cdot \sin(\theta) \cdot \sin(\phi) \\ R + h_c \cdot \cos(\theta) \end{bmatrix}$$

$$rp := \begin{bmatrix} x + h_c \sin(\theta) \cos(\phi) \\ y + h_c \sin(\theta) \sin(\phi) \\ R + h_c \cos(\theta) \end{bmatrix}$$

> ## pendulum center of mass velocities

$$rp : \\ diff~(\%, x) \cdot Dx + diff~(\%, y) \cdot Dy + diff~(\%, \theta) \cdot Dtheta + diff~(\%, \phi) \cdot Dphi :$$

$$Drp := \%$$

$$Drp := \begin{bmatrix} Dx + Dtheta h_c \cos(\theta) \cos(\phi) - Dphi h_c \sin(\theta) \sin(\phi) \\ Dy + Dtheta h_c \cos(\theta) \sin(\phi) + Dphi h_c \sin(\theta) \cos(\phi) \\ -Dtheta h_c \sin(\theta) \end{bmatrix}$$

>

Body Velocity of Pendulum

$$\begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ Dphi \end{bmatrix} + \begin{bmatrix} 0 \\ Dtheta \\ 0 \end{bmatrix} :$$

$\Omega_{pend} := \%$

$$\Omega_{pend} := \begin{bmatrix} -\sin(\theta) Dphi \\ Dtheta \\ \cos(\theta) Dphi \end{bmatrix}$$

>

Base translational KE + Rotational KE

$$\frac{1}{2} \cdot (mp \cdot (Drp(1)^2 + Drp(2)^2 + Drp(3)^2)) + \frac{1}{2} \cdot (I1 \cdot \Omega_{pend}(1)^2 + I2 \cdot \Omega_{pend}(2)^2 + I3 \cdot \Omega_{pend}(3)^2) :$$

KEpen := %

$$\begin{aligned} KEpen := & \frac{1}{2} mp \left((Dx + Dtheta h_c \cos(\theta) \cos(\phi) - Dphi h_c \sin(\theta) \sin(\phi))^2 + (Dy \right. \\ & \left. + Dtheta h_c \cos(\theta) \sin(\phi) + Dphi h_c \sin(\theta) \cos(\phi))^2 + Dtheta^2 h_c^2 \sin^2(\theta) \right) \\ & + \frac{1}{2} I1 \sin^2(\theta) Dphi^2 + \frac{1}{2} I2 Dtheta^2 + \frac{1}{2} I3 \cos^2(\theta) Dphi^2 \end{aligned}$$

>

variable PEPE := hc*cos(theta)*mp*g

$$PE := hc \cos(\theta) mp g$$

Velocities of the left and right wheel centers.

>

$$v + \frac{w}{2} Dphi = Dpsir \cdot R$$

$$v + \frac{1}{2} w Dphi = Dpsir R$$

>

$$v - \frac{w}{2} \cdot Dphi = Dpsil \cdot R$$

$$v - \frac{1}{2} w Dphi = Dpsil R$$

>

Holonomic Constraint

$$v + \frac{w \cdot Dphi}{2} - \left(v - \frac{w \cdot Dphi}{2} \right) = (Dpsir - Dpsil) \cdot R :$$

$isolate(\%, Dphi)$

$$Dphi = \frac{(Dpsir - Dpsil) R}{w}$$

>

```

KEwl + KEwr + KEpen - PE :
subs(KEm=0, %) :
eval(%, {hc=h_c, mp=m_p}) :
### NON HOLONOMIC CONSTRAINTS: CAN NOT BE SUBSTITUTED YET

### Holonomic Constraints: Can be substituted
subs(Dphi =  $\frac{R \cdot (Dpsir - Dpsil)}{W}$ , %) :
subs(phi =  $\frac{R \cdot (psir - psil)}{W}$ , %) :

#eval(%, {Dpsil = Dpsir, psil = psir, Dphi = 0}) :
##this makes both both wheel angles, velocities, accels, the same. It checks with
SegBot_same_tau
L := % : simplify(%, size)

```

$$\begin{aligned}
& \frac{3}{4} mw R^2 Dpsil^2 + \frac{1}{4} \frac{mw R^4 (Dpsir - Dpsil)^2}{W^2} + \frac{3}{4} mw R^2 Dpsir^2 + \frac{1}{2} m_p \left(\left(Dx \right. \right. \\
& \quad \left. \left. + Dtheta h_c \cos(\theta) \cos\left(\frac{R (psir - psil)}{W}\right) \right. \right. \\
& \quad \left. \left. - \frac{R (Dpsir - Dpsil) h_c \sin(\theta) \sin\left(\frac{R (psir - psil)}{W}\right)}{W} \right)^2 + \left(Dy \right. \right. \\
& \quad \left. \left. + Dtheta h_c \cos(\theta) \sin\left(\frac{R (psir - psil)}{W}\right) \right. \right. \\
& \quad \left. \left. + \frac{R (Dpsir - Dpsil) h_c \sin(\theta) \cos\left(\frac{R (psir - psil)}{W}\right)}{W} \right)^2 + Dtheta^2 h_c^2 \sin^2(\theta) \right) \\
& + \frac{1}{2} \frac{I1 \sin^2(\theta) R^2 (Dpsir - Dpsil)^2}{W^2} + \frac{1}{2} I2 Dtheta^2 \\
& + \frac{1}{2} \frac{I3 \cos^2(\theta) R^2 (Dpsir - Dpsil)^2}{W^2} - h_c \cos(\theta) m_p g
\end{aligned}$$

Apply Lagrange's Equation

Equations with respect to generalized coordinates

>

$$\begin{aligned}
 & (\text{diff}(L, Dx, Dx)) * DDx + (\text{diff}(L, Dx, Dy)) * DDy + (\text{diff}(L, Dx, Dpsir)) * DDpsir \\
 & + (\text{diff}(L, Dx, Dpsil)) * DDpsil + (\text{diff}(L, Dx, Dtheta)) * DDtheta + (\text{diff}(L, Dx, \\
 & Dphi)) * DDphi + (\text{diff}(L, Dx, x)) * Dx + (\text{diff}(L, Dx, y)) * Dy + (\text{diff}(L, Dx, \\
 & psir)) * Dpsir + (\text{diff}(L, Dx, psil)) * Dpsil + (\text{diff}(L, Dx, theta)) * Dtheta \\
 & + (\text{diff}(L, Dx, phi)) * Dphi - \text{diff}(L, x) - \text{lambda}[1] : \\
 \text{eq1} := \% : \text{simplify}(\%, \text{size})
 \end{aligned}$$

$$\begin{aligned}
 & \frac{1}{W^2} \left(-m_p h_c \left((Dtheta^2 W^2 + R^2 (Dpsil - Dpsir)^2) \sin(\theta) \right. \right. \\
 & \left. \left. - \cos(\theta) DDtheta W^2 \right) \cos\left(\frac{R (psir - psil)}{W}\right) - \left(m_p R h_c \left((DDpsir \right. \right. \right. \\
 & \left. \left. - DDpsil) \sin(\theta) - 2 Dtheta \cos(\theta) (Dpsil - Dpsir) \right) \sin\left(\frac{R (psir - psil)}{W}\right) \right. \\
 & \left. \left. - (m_p DDx - \lambda_1) W \right) W \right)
 \end{aligned}$$

>

$$\begin{aligned}
 & (\text{diff}(L, Dy, Dx)) * DDx + (\text{diff}(L, Dy, Dy)) * DDy + (\text{diff}(L, Dy, Dpsir)) * DDpsir \\
 & + (\text{diff}(L, Dy, Dpsil)) * DDpsil + (\text{diff}(L, Dy, Dtheta)) * DDtheta + (\text{diff}(L, Dy, \\
 & Dphi)) * DDphi + (\text{diff}(L, Dy, x)) * Dx + (\text{diff}(L, Dy, y)) * Dy + (\text{diff}(L, Dy, \\
 & psir)) * Dpsir + (\text{diff}(L, Dy, psil)) * Dpsil + (\text{diff}(L, Dy, theta)) * Dtheta \\
 & + (\text{diff}(L, Dy, phi)) * Dphi - \text{diff}(L, y) - \text{lambda}[2] : \\
 \text{eq2} := \% : \text{simplify}(\%, \text{size})
 \end{aligned}$$

$$\begin{aligned}
 & \frac{1}{W^2} \left(-m_p h_c \left((Dtheta^2 W^2 + R^2 (Dpsil - Dpsir)^2) \sin(\theta) \right. \right. \\
 & \left. \left. - \cos(\theta) DDtheta W^2 \right) \sin\left(\frac{R (psir - psil)}{W}\right) + \left(m_p R h_c \left((DDpsir \right. \right. \right. \\
 & \left. \left. - DDpsil) \sin(\theta) - 2 Dtheta \cos(\theta) (Dpsil - Dpsir) \right) \cos\left(\frac{R (psir - psil)}{W}\right) \right. \\
 & \left. \left. + (m_p DDy - \lambda_2) W \right) W \right)
 \end{aligned}$$

>

$$\begin{aligned}
 & (\text{diff}(L, Dpsir, Dx)) * DDx + (\text{diff}(L, Dpsir, Dy)) * DDy + (\text{diff}(L, Dpsir, Dpsir)) \\
 & * DDpsir + (\text{diff}(L, Dpsir, Dpsil)) * DDpsil + (\text{diff}(L, Dpsir, Dtheta)) * DDtheta \\
 & + (\text{diff}(L, Dpsir, Dphi)) * DDphi + (\text{diff}(L, Dpsir, x)) * Dx + (\text{diff}(L, Dpsir, y)) \\
 & * Dy + (\text{diff}(L, Dpsir, psir)) * Dpsir + (\text{diff}(L, Dpsir, psil)) * Dpsil + (\text{diff}(L, \\
 & Dpsir, theta)) * Dtheta + (\text{diff}(L, Dpsir, phi)) * Dphi - \text{diff}(L, psir) + \frac{R}{2} \\
 & \cdot (\text{lambda}[1] \cdot \cos(\text{phi}) + \text{lambda}[2] \cdot \sin(\text{phi})) - \text{taur} : \\
 \text{eq3} := \% : \text{simplify}(\%, \text{size})
 \end{aligned}$$

$$\begin{aligned}
& \frac{1}{2} \frac{1}{W^2} \left(-4 R^2 \sin(\theta) h_c^2 \left(\left(-\frac{1}{2} DDpsir + \frac{1}{2} DDpsil \right) \sin(\theta) \right. \right. \\
& \quad \left. \left. + Dtheta \cos(\theta) (Dpsil - Dpsir) \right) m_p \cos\left(\frac{R(psir - psil)}{W}\right)^2 \right. \\
& \quad \left. + 2 m_p R h_c \sin(\theta) \cos\left(\frac{R(psir - psil)}{W}\right) DDy W - 4 R^2 \sin(\theta) h_c^2 \left(\left(-\frac{1}{2} DDpsir \right. \right. \right. \\
& \quad \left. \left. + \frac{1}{2} DDpsil \right) \sin(\theta) + Dtheta \cos(\theta) (Dpsil - Dpsir) \right) \\
& \quad m_p \sin\left(\frac{R(psir - psil)}{W}\right)^2 - 2 m_p R h_c \sin(\theta) \sin\left(\frac{R(psir - psil)}{W}\right) DDx W \\
& \quad - 2 R^2 II (-DDpsir + DDpsil) \sin(\theta)^2 - 4 R^2 Dtheta \cos(\theta) (-I3 + II) (Dpsil \\
& \quad - Dpsir) \sin(\theta) - 2 R^2 I3 (-DDpsir + DDpsil) \cos(\theta)^2 + R W^2 \lambda_1 \cos(\phi) \\
& \quad + R W^2 \lambda_2 \sin(\phi) - mw (-DDpsir + DDpsil) R^4 + 3 R^2 DDpsir mw W^2 \\
& \quad \left. - 2 taur W^2 \right)
\end{aligned}$$

>

$$\begin{aligned}
& (diff(L, Dpsil, Dx)) * DDx + (diff(L, Dpsil, Dy)) * DDy + (diff(L, Dpsil, Dpsir)) \\
& \quad * DDpsir + (diff(L, Dpsil, Dpsil)) * DDpsil + (diff(L, Dpsil, Dtheta)) * DDtheta \\
& \quad + (diff(L, Dpsil, Dphi)) * DDphi + (diff(L, Dpsil, x)) * Dx + (diff(L, Dpsil, y)) \\
& \quad * Dy + (diff(L, Dpsil, psir)) * Dpsir + (diff(L, Dpsil, psil)) * Dpsil + (diff(L, \\
& \quad Dpsil, theta)) * Dtheta + (diff(L, Dpsil, phi)) * Dphi - diff(L, psil) + \frac{R}{2} \\
& \quad \cdot (\text{lambda}[1] \cdot \cos(\text{phi}) + \text{lambda}[2] \cdot \sin(\text{phi})) - \text{taul} : \\
& \text{eq4} := \% : \text{simplify}(\%, \text{size})
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2} \frac{1}{W^2} \left(4 R^2 \sin(\theta) h_c^2 \left(\left(-\frac{1}{2} DDpsir + \frac{1}{2} DDpsil \right) \sin(\theta) + Dtheta \cos(\theta) (Dpsil \right. \right. \\
& \quad \left. \left. - Dpsir) \right) m_p \cos\left(\frac{R(psir - psil)}{W}\right)^2 \right. \\
& \quad \left. - 2 m_p R h_c \sin(\theta) \cos\left(\frac{R(psir - psil)}{W}\right) DDy W + 4 R^2 \sin(\theta) h_c^2 \left(\left(-\frac{1}{2} DDpsir \right. \right. \right. \\
& \quad \left. \left. + \frac{1}{2} DDpsil \right) \sin(\theta) + Dtheta \cos(\theta) (Dpsil - Dpsir) \right) \\
& \quad m_p \sin\left(\frac{R(psir - psil)}{W}\right)^2 + 2 m_p R h_c \sin(\theta) \sin\left(\frac{R(psir - psil)}{W}\right) DDx W \\
& \quad + 2 R^2 II (-DDpsir + DDpsil) \sin(\theta)^2 + 4 R^2 Dtheta \cos(\theta) (-I3 + II) (Dpsil \\
& \quad - Dpsir) \sin(\theta) + 2 R^2 I3 (-DDpsir + DDpsil) \cos(\theta)^2 + R W^2 \lambda_1 \cos(\phi) \\
& \quad + R W^2 \lambda_2 \sin(\phi) + mw (-DDpsir + DDpsil) R^4 + 3 R^2 DDpsil mw W^2 \\
& \quad \left. - 2 taul W^2 \right)
\end{aligned}$$

>

$$\begin{aligned} & (\text{diff}(L, Dtheta, Dx)) * DDx + (\text{diff}(L, Dtheta, Dy)) * DDy + (\text{diff}(L, Dtheta, Dpsir)) \\ & * DDpsir + (\text{diff}(L, Dtheta, Dpsil)) * DDpsil + (\text{diff}(L, Dtheta, Dtheta)) \\ & * DDtheta + (\text{diff}(L, Dtheta, Dphi)) * DDphi + (\text{diff}(L, Dtheta, x)) * Dx \\ & + (\text{diff}(L, Dtheta, y)) * Dy + (\text{diff}(L, Dtheta, psir)) * Dpsir + (\text{diff}(L, Dtheta, \\ & psil)) * Dpsil + (\text{diff}(L, Dtheta, theta)) * Dtheta + (\text{diff}(L, Dtheta, phi)) * Dphi \\ & - \text{diff}(L, theta) + \frac{(taul + taur)}{N} : \end{aligned}$$

eq5 := % : simplify(% , size)

$$\begin{aligned} & \frac{1}{W^2 N} \left(-m_p N \left((Dtheta^2 W^2 + R^2 (Dpsil - Dpsir)^2) \sin(\theta) \right. \right. \\ & \left. \left. - \cos(\theta) DDtheta W^2 \right) \cos(\theta) h_c^2 \cos\left(\frac{R (psir - psil)}{W}\right)^2 \right. \\ & \left. + m_p h_c \cos(\theta) \cos\left(\frac{R (psir - psil)}{W}\right) DDx W^2 N - m_p N \left((Dtheta^2 W^2 \right. \right. \right. \\ & \left. \left. + R^2 (Dpsil - Dpsir)^2) \sin(\theta) - \cos(\theta) DDtheta W^2 \right) \cos(\theta) \right. \\ & \left. h_c^2 \sin\left(\frac{R (psir - psil)}{W}\right)^2 + m_p h_c \cos(\theta) \sin\left(\frac{R (psir - psil)}{W}\right) DDy W^2 N \right. \\ & \left. - N \sin(\theta) \left(-m_p h_c^2 W^2 Dtheta^2 + R^2 (Dpsil - Dpsir)^2 (-I3 + I1) \right) \cos(\theta) \right. \\ & \left. + W^2 (taul + DDtheta N I2 - h_c \sin(\theta) m_p g N + taur + DDtheta N m_p \right. \\ & \left. h_c^2 \sin(\theta)^2) \right) \end{aligned}$$

Non-Holonomic Constraints

>

$$Cc1 := Dx - \frac{R}{2} \cdot (Dpsir + Dpsil) \cdot \cos(phi) :$$

$$Cc2 := Dy - \frac{R}{2} \cdot (Dpsir + Dpsil) \cdot \sin(phi) :$$

Time Derivative of Non-Holonomic Constraints

>

$$\text{diff}(Cc1, Dx) \cdot DDx + \text{diff}(Cc1, Dpsir) \cdot DDpsir + \text{diff}(Cc1, Dpsil) \cdot DDpsil \\ + \text{diff}(Cc1, phi) \cdot Dphi : eq7 := %;$$

$$\text{diff}(Cc2, Dy) \cdot DDy + \text{diff}(Cc2, Dpsir) \cdot DDpsir + \text{diff}(Cc2, Dpsil) \cdot DDpsil \\ + \text{diff}(Cc2, phi) \cdot Dphi : eq8 := %;$$

$$\begin{aligned} eq7 := & DDx - \frac{1}{2} R \cos(\phi) DDpsir - \frac{1}{2} R \cos(\phi) DDpsil + \frac{1}{2} R (Dpsir \\ & + Dpsil) \sin(\phi) Dphi \end{aligned}$$

$$\begin{aligned} eq8 := & DDy - \frac{1}{2} R \sin(\phi) DDpsir - \frac{1}{2} R \sin(\phi) DDpsil - \frac{1}{2} R (Dpsir \\ & + Dpsil) \cos(\phi) Dphi \end{aligned}$$

Solve for the Lagrange multipliers, and time derivatives of X and Y.

Use the constraint equations, time derivative of the constraint equations, and the Lagrange Equations w.r.t. x and y

> $Lsol := solve(\{eq1=0, eq2=0, eq7=0, eq8=0, Cc1=0, Cc2=0\}, \{\lambda_1, \lambda_2, DDx, DDy, Dx, Dy\})$

$$\begin{aligned}
Lsol := & \left[DDx = -\frac{1}{2} R \sin(\phi) Dphi Dpsil + \frac{1}{2} R \cos(\phi) DDpsir + \frac{1}{2} R \cos(\phi) DDpsi \right. \\
& - \frac{1}{2} R \sin(\phi) Dphi Dpsir, DDy = \frac{1}{2} R \cos(\phi) Dphi Dpsil + \frac{1}{2} R \sin(\phi) DDpsir \\
& + \frac{1}{2} R \sin(\phi) DDpsil + \frac{1}{2} R \cos(\phi) Dphi Dpsir, Dx = \frac{1}{2} R \cos(\phi) Dpsir \\
& + \frac{1}{2} R \cos(\phi) Dpsil, Dy = \frac{1}{2} R \sin(\phi) Dpsir + \frac{1}{2} R \sin(\phi) Dpsil, \lambda_1 \\
& = \frac{1}{2} \frac{1}{W^2} \left(m_p \left(4 h_c R Dpsir Dtheta \cos(\theta) \sin\left(\frac{R(-psir+psil)}{W}\right) W \right. \right. \\
& + 2 R h_c \sin(\theta) \sin\left(\frac{R(-psir+psil)}{W}\right) DDpsir W \\
& - 2 R h_c \sin(\theta) \sin\left(\frac{R(-psir+psil)}{W}\right) DDpsil W \\
& + 2 h_c \cos(\theta) \cos\left(\frac{R(-psir+psil)}{W}\right) DDtheta W^2 \\
& - 2 h_c R^2 Dpsil^2 \sin(\theta) \cos\left(\frac{R(-psir+psil)}{W}\right) \\
& + 4 h_c R^2 Dpsir \sin(\theta) \cos\left(\frac{R(-psir+psil)}{W}\right) Dpsil \\
& - 2 h_c R^2 Dpsir^2 \sin(\theta) \cos\left(\frac{R(-psir+psil)}{W}\right) \\
& - 4 h_c R Dpsil Dtheta \cos(\theta) \sin\left(\frac{R(-psir+psil)}{W}\right) W \\
& \left. \left. - 2 h_c Dtheta^2 W^2 \sin(\theta) \cos\left(\frac{R(-psir+psil)}{W}\right) - W^2 R \sin(\phi) Dphi Dpsil \right. \right. \\
& \left. \left. + W^2 R \cos(\phi) DDpsir + W^2 R \cos(\phi) DDpsil - W^2 R \sin(\phi) Dphi Dpsir \right) \right), \lambda_2 \\
& = \frac{1}{2} \frac{1}{W^2} \left(m_p \left(4 h_c R Dpsir Dtheta \cos(\theta) \cos\left(\frac{R(-psir+psil)}{W}\right) W \right. \right. \\
& + 2 R h_c \sin(\theta) \cos\left(\frac{R(-psir+psil)}{W}\right) DDpsir W \\
& - 2 R h_c \sin(\theta) \cos\left(\frac{R(-psir+psil)}{W}\right) DDpsil W \\
& - 2 h_c \cos(\theta) \sin\left(\frac{R(-psir+psil)}{W}\right) DDtheta W^2 \\
& + 2 h_c R^2 Dpsil^2 \sin(\theta) \sin\left(\frac{R(-psir+psil)}{W}\right) \\
& - 4 h_c R^2 Dpsir \sin(\theta) \sin\left(\frac{R(-psir+psil)}{W}\right) Dpsil \\
& + 2 h_c R^2 Dpsir^2 \sin(\theta) \sin\left(\frac{R(-psir+psil)}{W}\right) \\
& - 4 h_c R Dpsil Dtheta \cos(\theta) \cos\left(\frac{R(-psir+psil)}{W}\right) W \\
& \left. \left. + 2 h_c Dtheta^2 W^2 \sin(\theta) \sin\left(\frac{R(-psir+psil)}{W}\right) + W^2 R \cos(\phi) Dphi Dpsil \right. \right. \\
& \left. \left. + W^2 R \sin(\phi) DDpsir + W^2 R \sin(\phi) DDpsil + W^2 R \cos(\phi) Dphi Dpsir \right) \right]
\end{aligned}$$

Eliminate the Lagrange multipliers, velocity of x and y , acceleration of x and y .

```

> [ eq3
   eq4
   eq5 ] :
map( eval, %, Lsol ) :
subs( Dphi =  $\frac{R \cdot (Dpsir - Dpsil)}{W}$ , % ) :
subs( phi =  $\frac{R \cdot (psir - psil)}{W}$ , % ) :
simplify( %, trig ) :
eq1f := %(1);
eq2f := %%%(2);
eq3f := %%%%(3);

```

$$\begin{aligned}
eq1f := & \frac{1}{4} \frac{1}{W^2} \left(2 R^4 DDpsir mw - 2 R^4 DDpsil mw \right. \\
& - 8 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsil + 8 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsir \\
& + 8 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsil - 8 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsir \\
& + 6 R^2 DDpsir mw W^2 + 4 R^2 DDpsir I3 \cos(\theta)^2 - 4 R^2 DDpsil I3 \cos(\theta)^2 \\
& - 4 taur W^2 - 4 m_p R^3 h_c \sin(\theta) Dpsil^2 - 8 R^2 \cos(\theta) Dtheta m_p h_c^2 \sin(\theta) Dpsil \\
& + 2 R m_p h_c \cos(\theta) DDtheta W^2 - 2 R m_p h_c Dtheta^2 W^2 \sin(\theta) \\
& + 4 R^3 m_p h_c Dpsir \sin(\theta) Dpsil - 4 R^2 DDpsil m_p h_c^2 - 4 R^2 DDpsir I1 \cos(\theta)^2 \\
& + 4 R^2 DDpsil I1 \cos(\theta)^2 - 4 R^2 DDpsir m_p h_c^2 \cos(\theta)^2 + 4 R^2 DDpsil m_p h_c^2 \cos(\theta)^2 \\
& + R^2 m_p W^2 DDpsir + R^2 m_p W^2 DDpsil + 4 R^2 DDpsir m_p h_c^2 \\
& \left. + 8 R^2 \cos(\theta) Dtheta m_p h_c^2 \sin(\theta) Dpsir - 4 R^2 DDpsil I1 + 4 R^2 DDpsir I1 \right)
\end{aligned}$$

$$\begin{aligned}
eq2f := & \frac{1}{4} \frac{1}{W^2} \left(-4 taur W^2 + 6 R^2 DDpsil mw W^2 - 2 R^4 DDpsir mw + 2 R^4 DDpsil mw \right. \\
& + 8 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsil - 8 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsir \\
& - 8 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsil + 8 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsir \\
& - 4 R^2 DDpsir I3 \cos(\theta)^2 + 4 R^2 DDpsil I3 \cos(\theta)^2 + 8 R^2 \cos(\theta) Dtheta m_p \\
& h_c^2 \sin(\theta) Dpsil + 2 R m_p h_c \cos(\theta) DDtheta W^2 - 2 R m_p h_c Dtheta^2 W^2 \sin(\theta) \\
& + 4 R^3 m_p h_c Dpsir \sin(\theta) Dpsil + 4 R^2 DDpsil m_p h_c^2 + 4 R^2 DDpsir I1 \cos(\theta)^2 \\
& - 4 R^2 DDpsil I1 \cos(\theta)^2 + 4 R^2 DDpsir m_p h_c^2 \cos(\theta)^2 - 4 R^2 DDpsil m_p h_c^2 \cos(\theta)^2 \\
& - 4 m_p R^3 h_c \sin(\theta) Dpsir^2 + R^2 m_p W^2 DDpsir + R^2 m_p W^2 DDpsil \\
& - 4 R^2 DDpsir m_p h_c^2 - 8 R^2 \cos(\theta) Dtheta m_p h_c^2 \sin(\theta) Dpsir + 4 R^2 DDpsil I1 \\
& \left. - 4 R^2 DDpsir I1 \right)
\end{aligned}$$

$$\begin{aligned}
eq3f := & \frac{1}{2} \frac{1}{W^2 N} (2 \text{ taur } W^2 + 4 m_p h_c^2 N R^2 \sin(\theta) \text{ Dpsil } \cos(\theta) \text{ Dpsir} \\
& + 2 \text{ DDtheta } W^2 N I2 + 2 \text{ taur } W^2 - 2 h_c \sin(\theta) m_p g W^2 N \\
& - 2 II \sin(\theta) R^2 \cos(\theta) N \text{ Dpsil}^2 - 2 II \sin(\theta) R^2 \cos(\theta) N \text{ Dpsir}^2 \\
& + 2 I3 \cos(\theta) R^2 \sin(\theta) N \text{ Dpsil}^2 + 2 I3 \cos(\theta) R^2 \sin(\theta) N \text{ Dpsir}^2 \\
& + 4 II \sin(\theta) R^2 \cos(\theta) N \text{ Dpsil } \text{ Dpsir} - 4 I3 \cos(\theta) R^2 \sin(\theta) N \text{ Dpsil } \text{ Dpsir} \\
& + 2 \text{ DDtheta } W^2 N m_p h_c^2 + m_p h_c \cos(\theta) R W^2 N \text{ DDpsil} \\
& + m_p h_c \cos(\theta) R W^2 N \text{ DDpsir} - 2 m_p h_c^2 N R^2 \sin(\theta) \text{ Dpsil}^2 \cos(\theta) - 2 m_p \\
& h_c^2 N R^2 \sin(\theta) \text{ Dpsir}^2 \cos(\theta))
\end{aligned}$$

$$> \begin{bmatrix} \text{coeff}(eq1f, \text{DDpsir}) & \text{coeff}(eq1f, \text{DDpsil}) & \text{coeff}(eq1f, \text{DDtheta}) \\ \text{coeff}(eq2f, \text{DDpsir}) & \text{coeff}(eq2f, \text{DDpsil}) & \text{coeff}(eq2f, \text{DDtheta}) \\ \text{coeff}(eq3f, \text{DDpsir}) & \text{coeff}(eq3f, \text{DDpsil}) & \text{coeff}(eq3f, \text{DDtheta}) \end{bmatrix} : M := \% : \\
\text{simplify}(\%, \text{size});$$

$$\begin{aligned}
& \left[\frac{1}{2} \frac{1}{W^2} \left(R^2 \left((-2 II - 2 m_p h_c^2 + 2 I3) \cos(\theta)^2 + \left(\frac{1}{2} W^2 + 2 h_c^2 \right) m_p + 3 mw W^2 \right. \right. \right. \\
& \left. \left. \left. + mw R^2 + 2 II \right) \right), \right. \\
& \left. - \frac{1}{2} \frac{1}{W^2} \left(R^2 \left((-2 II - 2 m_p h_c^2 + 2 I3) \cos(\theta)^2 + \left(-\frac{1}{2} W^2 + 2 h_c^2 \right) m_p \right. \right. \right. \right. \\
& \left. \left. \left. + mw R^2 + 2 II \right) \right), \frac{1}{2} R m_p h_c \cos(\theta) \right], \\
& \left[\right. \\
& \left. - \frac{1}{2} \frac{1}{W^2} \left(R^2 \left((-2 II - 2 m_p h_c^2 + 2 I3) \cos(\theta)^2 + \left(-\frac{1}{2} W^2 + 2 h_c^2 \right) m_p \right. \right. \right. \right. \\
& \left. \left. \left. + mw R^2 + 2 II \right) \right), \frac{1}{2} \frac{1}{W^2} \left(R^2 \left((-2 II - 2 m_p h_c^2 + 2 I3) \cos(\theta)^2 + \left(\frac{1}{2} W^2 \right. \right. \right. \right. \\
& \left. \left. \left. + 2 h_c^2 \right) m_p + 3 mw W^2 + mw R^2 + 2 II \right) \right), \frac{1}{2} R m_p h_c \cos(\theta) \right], \\
& \left[\frac{1}{2} R m_p h_c \cos(\theta), \frac{1}{2} R m_p h_c \cos(\theta), I2 + m_p h_c^2 \right]
\end{aligned}$$

check

$$> M - M^{\%T}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$> \begin{bmatrix} eq1f \\ eq2f \\ eq3f \end{bmatrix} : \text{coeff}(\%, g) \cdot g : G := \%$$

$$G := \begin{bmatrix} 0 \\ 0 \\ -h_c \sin(\theta) m_p g \end{bmatrix}$$

C vector

$$> \begin{bmatrix} eq1f \\ eq2f \\ eq3f \end{bmatrix} - \begin{bmatrix} \text{coeff}(eq1f, DDpsir) & \text{coeff}(eq1f, DDpsil) & \text{coeff}(eq1f, DDtheta) \\ \text{coeff}(eq2f, DDpsir) & \text{coeff}(eq2f, DDpsil) & \text{coeff}(eq2f, DDtheta) \\ \text{coeff}(eq3f, DDpsir) & \text{coeff}(eq3f, DDpsil) & \text{coeff}(eq3f, DDtheta) \end{bmatrix} \cdot \begin{bmatrix} DDpsir \\ DDpsil \\ DDtheta \end{bmatrix} :$$

$\text{eval}(\%, \{g=0, \text{taur}=0, \text{taul}=0\}) :$

$\text{simplify}(\%, \text{size}) : \text{factor}(\%) : C_v := \%$

$$C_v := \left[\left[-\frac{1}{2} \frac{1}{W^2} \left(R \sin(\theta) \left(2 m_p R^2 h_c Dpsil^2 + 4 R \cos(\theta) Dtheta I1 Dpsil - 4 R \cos(\theta) Dtheta I1 Dpsir - 4 R \cos(\theta) Dtheta I3 Dpsil + 4 R \cos(\theta) Dtheta I3 Dpsir - 2 R^2 m_p h_c Dpsir Dpsil + m_p h_c Dtheta^2 W^2 + 4 R \cos(\theta) Dtheta m_p h_c^2 Dpsil - 4 R \cos(\theta) Dtheta m_p h_c^2 Dpsir \right) \right], \right. \\ \left. \left[\frac{1}{2} \frac{1}{W^2} \left(R \sin(\theta) \left(4 R \cos(\theta) Dtheta I1 Dpsil - 4 R \cos(\theta) Dtheta I1 Dpsir - 4 R \cos(\theta) Dtheta I3 Dpsil + 4 R \cos(\theta) Dtheta I3 Dpsir + 2 R^2 m_p h_c Dpsir Dpsil - m_p h_c Dtheta^2 W^2 + 4 R \cos(\theta) Dtheta m_p h_c^2 Dpsil - 4 R \cos(\theta) Dtheta m_p h_c^2 Dpsir - 2 m_p R^2 h_c Dpsir^2 \right) \right], \right. \\ \left. \left[-\frac{R^2 \sin(\theta) \cos(\theta) (Dpsil - Dpsir)^2 (I1 - I3 + m_p h_c^2)}{W^2} \right] \right]$$

Simplify the Equations of Motion

For our control law we want $(\dot{M} - 2*C)$ to be skew-symmetric. This determines the terms in the C matrix.

> $M := \text{diff}(\%, \theta) \cdot D\theta : \dot{M} := \%$

$$\dot{M} := \begin{bmatrix} \frac{1}{4} \frac{1}{W^2} (D\theta (-8 R^2 I_3 \cos(\theta) \sin(\theta) + 8 R^2 I_1 \cos(\theta) \sin(\theta) + 8 R^2 m_p h_c^2 \cos(\theta) \sin(\theta))), \\ \frac{1}{4} \frac{1}{W^2} (D\theta (8 R^2 I_3 \cos(\theta) \sin(\theta) - 8 R^2 I_1 \cos(\theta) \sin(\theta) - 8 R^2 m_p h_c^2 \cos(\theta) \sin(\theta))), -\frac{1}{2} D\theta R m_p h_c \sin(\theta) \Big], \\ \left[\frac{1}{4} \frac{1}{W^2} (D\theta (8 R^2 I_3 \cos(\theta) \sin(\theta) - 8 R^2 I_1 \cos(\theta) \sin(\theta) - 8 R^2 m_p h_c^2 \cos(\theta) \sin(\theta))), \right. \\ \left. \frac{1}{4} \frac{1}{W^2} (D\theta (-8 R^2 I_3 \cos(\theta) \sin(\theta) + 8 R^2 I_1 \cos(\theta) \sin(\theta) + 8 R^2 m_p h_c^2 \cos(\theta) \sin(\theta))), -\frac{1}{2} D\theta R m_p h_c \sin(\theta) \right], \\ \left[-\frac{1}{2} D\theta R m_p h_c \sin(\theta), -\frac{1}{2} D\theta R m_p h_c \sin(\theta), 0 \right] \end{bmatrix}$$

> $DDq := \begin{bmatrix} DD\psi \\ DD\psi \\ DD\theta \end{bmatrix} : Dq := \begin{bmatrix} D\psi \\ D\psi \\ D\theta \end{bmatrix} :$

Make $\dot{M} - 2*C$ Skew-Symmetric

> $C1 := \begin{bmatrix} c11 & c12 & c13 \\ c21 & c22 & c23 \\ c31 & c32 & c33 \end{bmatrix} :$

> $\dot{M} - 2 \cdot C1 : asm := \%$

> $C1 \cdot Dq - Cv : Ccon := \%$

> $\text{solve}(\{asm(1,1)=0, asm(2,2)=0, asm(3,3)=0, asm(1,2)=-asm(2,1), asm(1,3)=-asm(3,1), asm(2,3)=-asm(3,2), Ccon(1)=0, Ccon(2)=0, Ccon(3)=0\}, \{c11, c12, c13, c21, c22, c23, c31, c33\}) :$
 $Csol1 := \%$

> $C1 : \text{eval}(\%, Csol1) : \text{simplify}(\%, \text{size}) : C := \%$

eliminate Dpsir from the denominator

> C : coeff(%, $\frac{1}{Dpsir}$) : eval(%, c32 = $-\frac{R^2 Dpsil (m_p h_c^2 + I1 - I3) \cos(\theta) \sin(\theta)}{W^2}$)

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

> Mdot - 2·C : simplify(%, size) :

Having Dpsir in the denominator will be a real pain: Try to pick c32 so it is not in the denominator

coeff(%(1, 2), $\frac{1}{Dpsir}$) : isolate(%, c32)

$$c32 = -\frac{R^2 Dpsil (I1 - I3 + m_p h_c^2) \cos(\theta) \sin(\theta)}{W^2}$$

>

check skew-symmetric

Mdot - 2·C : eval(%, c32 = $-\frac{R^2 Dpsil (m_p h_c^2 + I1 - I3) \cos(\theta) \sin(\theta)}{W^2}$) :

simplify(%) : Dq^{%T}.%Dq :
simplify(%)

0

> C : eval(%, c32 = $-\frac{R^2 Dpsil (m_p h_c^2 + I1 - I3) \cos(\theta) \sin(\theta)}{W^2}$) : simplify(%) :
C := %;

$$\begin{aligned}
C := & \left[\left[\frac{Dtheta R^2 \sin(\theta) \cos(\theta) (I1 - I3 + m_p h_c^2)}{W^2}, \right. \right. \\
& - \frac{\sin(\theta) R^3 h_c m_p (Dpsil - Dpsir)}{W^2}, - \frac{1}{2} \frac{1}{W^2} (\sin(\theta) R (4 R Dpsil \cos(\theta) m_p \\
& h_c^2 - 2 R \cos(\theta) I1 Dpsir + 2 R \cos(\theta) I3 Dpsir + 4 R Dpsil \cos(\theta) I1 \\
& - 4 R Dpsil \cos(\theta) I3 - 2 R \cos(\theta) m_p h_c^2 Dpsir + Dtheta m_p h_c W^2) \left. \right] , \\
& \left[\frac{1}{W^2} (\sin(\theta) R^2 (-2 Dtheta \cos(\theta) I1 + 2 Dtheta \cos(\theta) I3 \right. \\
& - 2 Dtheta \cos(\theta) m_p h_c^2 + m_p h_c R Dpsil - m_p h_c R Dpsir) \left. \right) , \\
& \frac{Dtheta R^2 \sin(\theta) \cos(\theta) (I1 - I3 + m_p h_c^2)}{W^2}, \frac{1}{2} \frac{1}{W^2} (\sin(\theta) R (\\
& - Dtheta m_p h_c W^2 + 2 R Dpsil \cos(\theta) I1 - 2 R Dpsil \cos(\theta) I3 \\
& + 2 R Dpsil \cos(\theta) m_p h_c^2) \left. \right] , \\
& \left[\frac{R^2 \sin(\theta) \cos(\theta) (I1 - I3 + m_p h_c^2) (2 Dpsil - Dpsir)}{W^2}, \right. \\
& \left. - \frac{R^2 Dpsil (I1 - I3 + m_p h_c^2) \cos(\theta) \sin(\theta)}{W^2}, 0 \right] \left. \right]
\end{aligned}$$

Non-Holonomic Segbot Control

This section shows how to build the linearized equations of motions. Next, the non-linear control gains are determined.

Linear Control

Build the Linearized Dynamic State-Space Matrices

>

$$\begin{aligned}
& -M^{-1} \cdot (Cv + G) : \\
& \text{map}(mtaylor, \% [theta, Dtheta, Dpsir = Dpsil], 2) : \\
Ag := & \text{map} \left(\text{eval}, \begin{bmatrix} 0 & 0 & \text{coeff}(\%(1), \text{theta}) \\ 0 & 0 & \text{coeff}(\%(2), \text{theta}) \\ 0 & 0 & \text{coeff}(\%(3), \text{theta}) \end{bmatrix} \right)
\end{aligned}$$

$$Ag := \begin{bmatrix} 0 & 0 & -\frac{m_p^2 h_c^2 g}{R (3 mw I2 + 3 mw m_p h_c^2 + m_p I2)} \\ 0 & 0 & -\frac{m_p^2 h_c^2 g}{R (3 mw I2 + 3 mw m_p h_c^2 + m_p I2)} \\ 0 & 0 & \frac{(3 mw + m_p) h_c m_p g}{3 mw I2 + 3 mw m_p h_c^2 + m_p I2} \end{bmatrix}$$

build state matrix

$$> A_c := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix};$$

$$> \left\langle \left\langle \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\rangle \left\langle \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\rangle, \left\langle A_g + A_c \right\rangle \left\langle \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\rangle \right\rangle : \text{simplify}(\%, \text{size}) :$$

$$A := \%;$$

$$\text{eval}(\%, \{m_p = mp, h_c = hc\}) :$$

$$\text{eval}\left(\%, \left\{mp = 1, W = \frac{15}{100}, R = \frac{4 \cdot 254}{10000}, N = 10, hp = \frac{15}{100}, g = \frac{9810}{1000}, mw = \frac{1}{4}, hc = \frac{15}{100}, I1 = \frac{49}{10000}, I2 = \frac{79}{10000}, I3 = \frac{23}{10000}\right\}\right) : \text{evalf}(\%);$$

$$A := \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{m_p^2 h_c^2 g}{R (3 mw I2 + 3 mw m_p h_c^2 + m_p I2)} & 0 & 0 & 0 \\ 0 & 0 & -\frac{m_p^2 h_c^2 g}{R (3 mw I2 + 3 mw m_p h_c^2 + m_p I2)} & 0 & 0 & 0 \\ 0 & 0 & \frac{(3 mw + m_p) h_c m_p g}{(3 mw h_c^2 + I2) m_p + 3 mw I2} & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0. & 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 0. & 0. & 1. \\ 0. & 0. & -70.76515171 & 0. & 0. & 0. \\ 0. & 0. & -70.76515171 & 0. & 0. & 0. \\ 0. & 0. & 83.88029316 & 0. & 0. & 0. \end{bmatrix}$$

$$> A : \#map(\text{eval}, \% , \{h_c = hc, m_p = mp\})$$

$$> \#CodeGeneration['Matlab'](\%?)$$

Build Input Matrix

$$> M^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{1}{N} & -\frac{1}{N} \end{bmatrix};$$

map(mtaylor, %, [theta, Dtheta], 2) :

$$\left\langle \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \text{simplify}(\%, \text{size}) \right\rangle :$$

map(eval, %, {h_c = hc, m_p = mp}) :

B := % : map(simplify, %, size) :

eval(%, {m_p = mp, h_c = hc}) :

$$\text{eval}\left(\%, \left\{ mp = 1, W = \frac{15}{100}, R = \frac{4 \cdot 254}{10000}, N = 10, hp = \frac{15}{100}, g = \frac{9810}{1000}, mw = \frac{1}{4}, hc = \frac{15}{100}, I1 = \frac{49}{10000}, I2 = \frac{79}{10000}, I3 = \frac{23}{10000} \right\}\right) : \text{evalf}(\%);$$

$$\begin{bmatrix} 0. & 0. \\ 0. & 0. \\ 0. & 0. \\ 170.5183554 & 30.95680327 \\ 30.95680327 & 170.5183554 \\ -53.79081279 & -53.79081279 \end{bmatrix}$$

Sliding Mode Control

$$> DDq_r := \begin{bmatrix} DDpsir_d - \text{lambda1} \cdot (Dpsir - Dpsir_d) \\ DDpsil_d - \text{lambda2} \cdot (Dpsil - Dpsil_d) \\ DDtheta_d - \text{lambda3} \cdot (Dtheta - Dtheta_d) \end{bmatrix};$$

$$Dq_r := \begin{bmatrix} Dpsir_d - \text{lambda1} \cdot (psir - psir_d) \\ Dpsil_d - \text{lambda2} \cdot (psil - psil_d) \\ Dtheta_d - \text{lambda3} \cdot (\text{theta} - \text{theta_d}) \end{bmatrix};$$

s := Dq - Dq_r :

$$> Km := \begin{bmatrix} Km11 & Km12 & Km13 \\ Km21 & Km22 & Km23 \\ Km31 & Km32 & Km33 \end{bmatrix};$$

> CI := 'CI' :

> ## Underactuated constraint for DDtheta_d

(M.DDq_r + C.Dq_r + G) - Km.s :

%(1) + %(2) + %(3) · N :

eval(%, {psir_d = psir, psil_d = psil}) :

DDthetadsol := isolate(%, DDtheta_d) : simplify(%, size)

$$\begin{aligned}
DDtheta_d = & \frac{1}{2} \frac{1}{(R m_p h_c \cos(\theta) + N (I_2 + m_p h_c^2)) W^2} \left(2 R \left((I_1 - I_3 + m_p \right. \right. \\
& h_c^2) R \left((Dpsil_d - 2 Dpsir_d) Dpsil + Dpsir_d Dpsir \right) N - (\theta - theta_d) (Dpsil \\
& - Dpsir) \lambda_3 + Dpsil Dtheta_d - Dpsir Dtheta_d - Dtheta (Dpsil_d - Dpsir_d) \\
& \sin(\theta) + \frac{1}{2} W^2 m_p h_c \left((\lambda_1 Dpsir - \lambda_1 Dpsir_d + \lambda_2 Dpsil - DDpsir_d \right. \\
& \left. - \lambda_2 Dpsil_d - DDpsil_d) N + 2 \lambda_3 (Dtheta - Dtheta_d) \right) \cos(\theta) \\
& + 2 m_p h_c \left((Dpsil_d - Dpsir_d) (Dpsil - Dpsir) R^3 - (-Dtheta_d + \lambda_3 (\theta \right. \\
& \left. - theta_d)) Dtheta W^2 R + g N W^2 \right) \sin(\theta) + 2 W^2 \left(\frac{1}{2} (3 mw + m_p) (\lambda_1 Dpsir \right. \\
& \left. - \lambda_1 Dpsir_d + \lambda_2 Dpsil - DDpsir_d - \lambda_2 Dpsil_d - DDpsil_d) R^2 + N \right. \\
& h_c^2 \lambda_3 (Dtheta - Dtheta_d) m_p + (I_2 Dtheta - I_2 Dtheta_d + Km33 (\theta \\
& \left. - theta_d)) \lambda_3 + Km31 Dpsir - Km31 Dpsir_d + Km32 Dpsil - Km32 Dpsil_d \\
& + Km33 Dtheta - Km33 Dtheta_d) N + (\theta - theta_d) (Km13 + Km23) \lambda_3 \\
& + (Km12 + Km22) Dpsil + (Km11 + Km21) Dpsir + (-Km11 - Km21) Dpsir_d \\
& \left. + (Km13 + Km23) Dtheta + (-Km22 - Km12) Dpsil_d - Dtheta_d (Km13 \right. \\
& \left. + Km23) \right)
\end{aligned}$$

Linear gains found using LQR in Matlab in LinearGains.m

KK =					
0.0000	1.0000	32.9176	0.1568	1.1640	5.0396
1.0000	0.0000	32.9176	1.1640	0.1568	5.0396

>

(MDDq_r + C.Dq_r + G) - Km.s:
eval(%s, DDthetaadsol) :

#eval(%s, {psir_d=psir, psil_d=psil}) :
#eval(%s, {Dpsir_d=0, Dpsil_d=0, DDpsir_d=0, Dtheta_d=0, theta_d=0, DDpsil_d=0}) :

eval(%s, {m_p=mp, h_c=hc}) :
eval(%s, {mp=1, W=15/100, R=4.254/10000, N=10, hp=15/100, g=9810/1000, mw=1/4, hc=15/100, I1=49/10000, I2=79/10000, I3=23/10000}) :

map(mtaylor, %s, [theta, Dtheta, Dpsil, Dpsir], 2) :
solve({coeff(%(1), theta) = 32 + 91/100, coeff(%(1), Dpsir) = 16/100, coeff(%(1), Dpsil) = 1 + 16/100, coeff(%(1), Dtheta) = 5 + 4/100, coeff(%(2), theta) = 32 + 91/100, coeff(%(2), Dpsir) = 1 + 16/100, coeff(%(2), Dpsil) = 16/100, coeff(%(2), Dtheta) = 5 + 4/100}) :

Fsol := %

$$\begin{aligned}
Fsol := & \left\{ Km11 = -\frac{102520807}{16177057} Km21 - \frac{313599506811}{40442642500} + \frac{297525567}{161770570} Km31, Km12 \right. \\
& = Km12, Km13 = Km13, Km21 = Km21, Km22 = \frac{297525567}{161770570} Km32 \\
& - \frac{313599506811}{40442642500} - \frac{102520807}{16177057} Km12, Km23 = Km13, Km31 = Km31, Km32 \\
& = Km32, Km33 = \frac{1520}{381} Km13 + \frac{335202}{15875}, \lambda1 = -\frac{527343750000}{2054486239} Km31 \\
& + \frac{318777890625000}{260919752353} + \frac{267187500000000}{260919752353} Km21, \lambda2 = -\frac{527343750000}{2054486239} Km32 \\
& \left. + \frac{318777890625000}{260919752353} + \frac{267187500000000}{260919752353} Km12, \lambda3 = \frac{34646867}{5363232} \right\}
\end{aligned}$$

>

(M.DDq_r + C.Dq_r + G) - Km.s :

```
eval(% DDtheta dsol) :
eval(% {mp=mp, hc=hc}) :
eval(% {mp=1, W=15/100, R=4.254/10000, N=10, hp=15/100, g=9810/1000, mw=1/4, hc
=15/100, I1=49/10000, I2=79/10000, I3=23/10000}) :
```

```
eval(% {psir_d=psir, psil_d=psil, theta_d=0, Dtheta_d=0, DDpsir_d=0, DDpsil_d
=0, Dpsir_d=0, Dpsil_d=0}) :
map( mtaylor, % [theta, Dtheta, Dpsil, Dpsir], 2) :
```

Flin := % :

Show that Km is positive definite

>

Km :

```
eval(% Fsol) :
% :
```

```
eval(% {Km13=-1, Km21=-1, Km12=-1, Km31=1, Km32=3}) :
```

```
Determinant(%(1..2, 1..2));
```

```
Determinant(%%);
%%%(1, 1)
```

$$\begin{aligned}
& \frac{1197701270030542855221}{1635607332382806250000} \\
& \frac{284430766079346088420479821}{12982633200788524609375000} \\
& \frac{17083902439}{40442642500}
\end{aligned}$$

> Fsol2 := {Km13=-1, Km21=-1, Km12=-1, Km31=1, Km32=3}

Fsol2 := {Km12=-1, Km13=-1, Km21=-1, Km31=1, Km32=3}

> Flin : eval(% Fsol) : eval(% Fsol2)

$$\begin{bmatrix} \frac{3291}{100} \theta + \frac{126}{25} Dtheta + \frac{29}{25} Dpsil + \frac{4}{25} Dpsir \\ \frac{3291}{100} \theta + \frac{126}{25} Dtheta + \frac{4}{25} Dpsil + \frac{29}{25} Dpsir \\ -\frac{3291}{500} \theta - \frac{126}{125} Dtheta - \frac{33}{250} Dpsil - \frac{33}{250} Dpsir \end{bmatrix}$$

>

Km :

eval(%, *Fsol*) :

$$\text{eval}\left(\%, Km12 = -\frac{318516970872647}{267187500000000} + \frac{381}{1520} Km32\right) :$$

$$\text{eval}\left(\%, Km21 = -\frac{318516970872647}{267187500000000} + \frac{381}{1520} Km31\right) :$$

eval(%, *Km31* = 10) :

$$\text{eval}\left(\%, Km32 = \frac{6500667934273246653191}{415579857893660625000}\right) :$$

eval(%, *Km13* = 0) :

$$Fsol2 := \{Km11 = \%(1, 1), Km12 = \%(1, 2), Km13 = \%(1, 3), Km21 = \%(2, 1), Km22 = \%(2, 2), Km23 = \%(2, 3), Km31 = \%(3, 1), Km32 = \%(3, 2), Km33 = \%(3, 3)\}$$

$$\begin{aligned} Fsol2 := & \left\{ Km11 = \frac{616482613778897}{267187500000000}, Km12 = \frac{1272431544212098103689189691}{466300234250760937500000000}, \right. \\ & Km13 = 0, Km21 = \frac{351209591627353}{267187500000000}, Km22 \\ & = \frac{1735390598695082009708279059}{466300234250760937500000000}, Km23 = 0, Km31 = 10, Km32 \\ & \left. = \frac{6500667934273246653191}{415579857893660625000}, Km33 = \frac{335202}{15875} \right\} \end{aligned}$$

>

Km : *eval*(%, *Fsol2*) : *evalf*(%)

$$\begin{bmatrix} 2.307303350 & 2.728781696 & 0. \\ 1.314468647 & 3.721616399 & 0. \\ 10. & 15.64240377 & 21.11508661 \end{bmatrix}$$

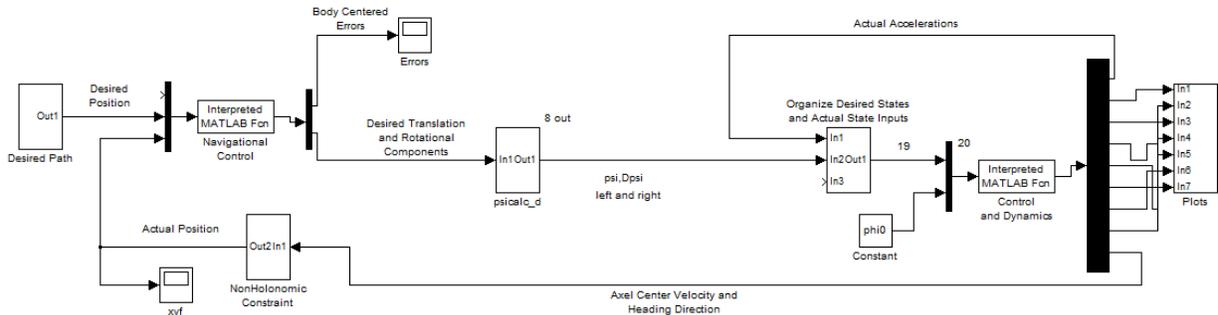
>

lambda3 : *eval*(%, *Fsol*) : *eval*(%, *Fsol2*) : *evalf*(%)

6.460072397

Non-Holonomic Segbot Matlab and Simulink Files

Non-Holonomic Segbot Main Simulink Diagram



This is the outer Simulink diagram for the Non-Holonomic version of the Segbot. The set-up is similar to that used by the rolling wheel, though not totally the same.

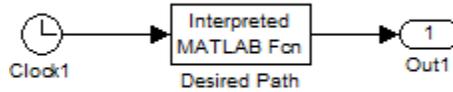
The *Desired Path* subsystem makes the trajectories to follow and sends it to the Navigational Control.

Navigational Control is an m-file that inputs the desired trajectories and the actual positions, and sends the desired translational and rotational velocities to the subsystem *psicalc_d*. This performs a simple coordinate transformation to map the desired translation and desired angular velocities calculated in the *Navigational Control* m-file to $\dot{\psi}_r$ and $\dot{\psi}_l$, which can be used by the sliding mode control.

The *Control and Dynamics* m-file inputs the actual coordinates and the desired trajectories. It then applies the computed torque to the dynamic system and calculates the systems actual accelerations. These accelerations are then fed to the subsystem labeled *Organize Desired States and Actual State Inputs*. The *Control and Dynamics* m-file also computes the Segbot axel center translational and rotational velocities. These are fed to the subsystem *NonHolonomic Constraint*, which calculates the actual orientation that is used by the *Navigational Control*.

The final subsystem is simply labeled *Plots*, and plots data from the *Control and Dynamics* m-file. The details are not presented.

Desired Path Coordinate System



This shows an inside look of the *Desired Path* subsystem. It consists of a clock and a custom Matlab m-file. The m-file follows.

```

function Y=DesiredCoords2(u)

% Wheel Radius
R = 4*2.54/100;

%% Figure 8 Parameters
r = 10;
omega =0.5;

t      = u(1);
x=r*sin(omega*t);
dx=omega*r*cos(omega*t);
ddx=-omega^2*r*sin(omega*t);
ddd=-omega^3*r*cos(omega*t);
if x==0 %%& dx>0
    y=0;
end

% Y-coordinate Logic
if x>0 && ddx<0 && dx>0
    y = (1/2)*sqrt(-4*x^2-2*r^2+2*sqrt(8*r^2*x^2+r^4));
end
if x>0 && ddx<0 && dx<0
    y = -(1/2)*sqrt(-4*x^2-2*r^2+2*sqrt(8*r^2*x^2+r^4));
end
if x<0 && ddx>0 && dx<0
    y = (1/2)*sqrt(-4*x^2-2*r^2+2*sqrt(8*r^2*x^2+r^4));
end
if x<0 && ddx>0 && dx>0
    y = -(1/2)*sqrt(-4*x^2-2*r^2+2*sqrt(8*r^2*x^2+r^4));
end

% Y time Derivatives
dy = (-4*x^3*dx-4*y^2*x*dx+2*r^2*x*dx)/(4*x^2*y+4*y^3+2*r^2*y);
ddy = (-2*(2*x*dx+2*y*dy)^2-4*x^2*dx^2-4*x^3*ddx-4*x^2*dy^2-4*y^2*dx^2-4*y^2*x*ddx-4*y^2*dy^2+2*r^2*dx^2+2*r^2*x*ddx-2*r^2*dy^2)/(4*x^2*y+4*y^3+2*r^2*y);
ddy = (-6*(2*x*dx+2*y*dy)*(2*dx^2+2*x*ddx+2*dy^2+2*y*ddy)-12*x^2*dx*ddx-4*x^3*ddd-12*x^2*dy*ddy-12*y^2*dx*ddx-4*y^2*x*ddd-12*y^2*dy*ddy+6*r^2*dx*ddx+2*r^2*x*ddd-6*r^2*dy*ddy)/(4*x^2*y+4*y^3+2*r^2*y);
  
```

```

% dddy = (-6*(2*dx^2+2*x*ddx+2*dy^2+2*y*ddy)^2-
(8*(2*x*dx+2*y*dy))*(6*dx*ddx+2*x*dddx+6*dy*ddy+2*y*ddd) -12*x^2*ddx^2-
16*x^2*dx*dddx-4*x^3*dddx-12*x^2*ddy^2-16*x^2*dy*ddy-12*y^2*ddx^2-
16*y^2*dx*dddx-4*y^2*x*dddx-12*y^2*ddy^2-
16*y^2*dy*ddy+6*r^2*ddx^2+8*r^2*dx*dddx+2*r^2*x*dddx-6*r^2*ddy^2-
8*r^2*dy*ddy)/(4*x^2*y+4*y^3+2*r^2*y);

% Time Constraints to ensure Bounds
if t<0.01
    dy = 5;
    ddy=0;
    dddy=-16.25;
end
if t==0
    phi=45*pi/180;
    dphi=0;
    ddphi=-1.4995;
    dpsi=70;
    ddpsi=0;
else

% Heading Direction and Time Derivatives
    if dx<0 && dy>0
        phi=atan2(dy,dx)-2*pi;
    else
        phi=atan2(dy,dx);
    end
    dphi=-(-ddy*dx+dy*ddx)/(dx^2+dy^2);
    ddphi=(ddy*dx-dy*dddx)/(dx^2+dy^2)-2*(-
dy*ddx+ddy*dx)*(dx*ddx+dy*ddy)/(dx^2+dy^2)^2;

%% Ensure Bounds
for n = 1:12
    if abs(t-(2*n-1)*pi)<.01
        ddphi = 0;
    end
end

    %% Translational to Rotational
    dpsi=(dx*cos(phi)+dy*sin(phi))/R;
    ddpsi = (ddx*cos(phi)-dx*sin(phi)*dphi+ddy*sin(phi)+dy*cos(phi)*dphi)/R;
end

% Output
Y = [x y dx dy phi dphi ddphi 0 dpsi ddpsi 0]';

```

Navigational Control

The m-file for the navigational control follows.

```
function qout = Navigational_Control_Segbot(u)
Rw = 4*2.54/100;
%%% Good check set v = vd; w = wd;

% % Good gains
% k1 =1;
% k2 =10;
% k3 = .5*2*sqrt(k2) ;
% k4=0;k5=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Best Gains\|/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k1 = 1;k2 =1;
k3 = 2*sqrt(k2);
k4 = 0;k5 = 20;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Imports %%
% t      = u(1);
xd      = u(2);
yd      = u(3);
% Dxd   = u(4);
% Dyd   = u(5);
phid    = u(6);
Dphid   = u(7);
DDphid  = u(8);
% DDDphid = u(9);

Dpsid   = u(10);
DDpsid  = u(11);
% DDDpsid = u(12);

x       = u(13);
y       = u(14);
phi     = u(15);

%%Constants
vd = Dpsid*Rw;    %% Desired Linear
wd = Dphid;      %% Desired Rotational

%% Global Differences
e = [xd-x yd-y phid-phi]';

% Rotation to body
R = [cos(phi) sin(phi) 0;-sin(phi) cos(phi) 0;0 0 1];

%% Local Error: [tangent lateral phi]'
pe = R*e;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
e1 = pe(1);
```

```

e2 = pe(2);
e3 = pe(3);

%% Velocity and Angular Velocity Control:
v = (vd*cos(e3)+k1*e1)      +   e1*k4*e2^2;

w = wd + vd*(k2*e2 +k3*sin(e3))      +   k2*k5*sin(e3)*e2^2;

Dphi = w;

% %%%%%%%%%%%%%%% Derivatives of error
De = [Dphi * e2 - v + vd * cos(e3) -Dphi * e1 + vd * sin(e3) Dphid - Dphi];
De1 = De(1);
De2 = De(2);
De3 = De(3);

%%%%%%%%%%%%%% Find Linear acceleration and DDphi
Dvd = Rw*DDpsid;    %% Desired Linear Acceleration

%% Derivative of [v w]'
Dq = [Dvd*cos(e3) + De1 * k1 - De3 * vd * sin(e3) Dvd * (k2 * e2 + k4 *
sin(e3)) + DDphid + De2 * vd * k2 + De3 * vd * k4 * cos(e3)];
Dv = Dq(1);        %% Linear Acceleration

DDphi = Dq(2);    %% DDphi

qout = [e' v Dv Dphi DDphi]';

```

Control and Dynamics

The *Control_and_Dynamics* m-file follows. It inputs the actual states and desired states. It computes a desired torque through the sliding mode control law and applies the torque to the dynamic system. It then computes the generalized coordinate accelerations, which are fed to the subsystem *Organize Desired States and Actual State Inputs* which calculates the actual positions and velocities by integrating the accelerations. The *Organize Desired States and Actual State Inputs* subsystem also inputs the desired trajectories and organizes the data so that it can be sent to the *Control_and_Dynamics* m-file. The *Control_and_Dynamics* m-file follows.

```
function y = Tracking(u)
```

```

mp = 1;
W = 15/100;
R = 4*2.54/100;
hc = 15/100;hp=hc;
I1 = 0.0049;
I2 = 0.0079;
I3 = 0.0023;
g = 9.81;
N = 10;

```

```

Itheta = 1/12*W*hp^3;
mw = .25;

Km = [0.2307303350e1 0.2728781696e1 0; 0.1314468647e1 0.3721616399e1 0;
0.10e2 0.1564240377e2 0.2111508661e2;];
lambda1=1;
lambda2=1;
lambda3 = 34646867/5363232;

%% From maple file
Km11 = Km(1,1);Km12 = Km(1,2);Km13 = Km(1,3);
Km21 = Km(2,1);Km22 = Km(2,2);Km23 = Km(1,3);
Km31 = Km(3,1);Km32 = Km(3,2);Km33 = Km(3,3);

psir = u(1);
psil = u(2);
theta = u(3);

Dpsir = u(4);
Dpsil = u(5);
Dtheta = u(6);

% psir_d = u(10);
psir_d=psir;
% psil_d = u(11);
psil_d=psil;

Dpsir_d = u(14);
Dpsil_d = u(15);

DDpsir_d = u(16);DDpsir_d=0;
DDpsil_d = u(17);DDpsil_d=0;

theta_d=0;%theta_d=0.2397;
Dtheta_d=0;%%Dtheta_d = -0.7685;

%% Uncomment for stabilization
% psi_d=0;theta_d=0;Dpsi_d=0;Dtheta_d=0;DDpsi_d=0;

%%%%% System Dynamics
M = [R ^ 2 * ((2 * I3 - 2 * mp * hc ^ 2 - 2 * I1) * cos(theta) ^ 2 + (W ^ 2 /
0.2e1 + (2 * hc ^ 2)) * mp + (2 * I1) + mw * R ^ 2 + 0.3e1 * mw * W ^ 2) / W
^ 2 / 0.2e1 -((2 * I3 - 2 * mp * hc ^ 2 - 2 * I1) * cos(theta) ^ 2 + ((2 * hc
^ 2) - W ^ 2 / 0.2e1) * mp + (2 * I1) + mw * R ^ 2) * R ^ 2 / W ^ 2 / 0.2e1 R
* mp * hc * cos(theta) / 0.2e1; -((2 * I3 - 2 * mp * hc ^ 2 - 2 * I1) *
cos(theta) ^ 2 + ((2 * hc ^ 2) - W ^ 2 / 0.2e1) * mp + (2 * I1) + mw * R ^ 2)
* R ^ 2 / W ^ 2 / 0.2e1 R ^ 2 * ((2 * I3 - 2 * mp * hc ^ 2 - 2 * I1) *
cos(theta) ^ 2 + (W ^ 2 / 0.2e1 + (2 * hc ^ 2)) * mp + (2 * I1) + mw * R ^ 2
+ 0.3e1 * mw * W ^ 2) / W ^ 2 / 0.2e1 R * mp * hc * cos(theta) / 0.2e1; R *
mp * hc * cos(theta) / 0.2e1 R * mp * hc * cos(theta) / 0.2e1 I2 + mp * hc ^
2;];
C = [Dtheta * R ^ 2 * sin(theta) * cos(theta) * (mp * hc ^ 2 + I1 - I3) / W ^
2 -0.1e1 / W ^ 2 * sin(theta) * R ^ 3 * hc * mp * (-Dpsir + Dpsil) -0.2e1 *
sin(theta) * (R * (-Dpsir / 0.2e1 + Dpsil) * (mp * hc ^ 2 + I1 - I3) *
cos(theta) + Dtheta * mp * hc * W ^ 2 / 0.4e1) * R / W ^ 2; sin(theta) * R ^

```

```

2 * (-0.2e1 * Dtheta * (mp * hc ^ 2 + I1 - I3) * cos(theta) + R * mp * hc *
(-Dpsir + Dpsil)) / W ^ 2 Dtheta * R ^ 2 * sin(theta) * cos(theta) * (mp * hc
^ 2 + I1 - I3) / W ^ 2 R * sin(theta) * (R * Dpsil * (mp * hc ^ 2 + I1 - I3)
* cos(theta) - Dtheta * mp * hc * W ^ 2 / 0.2e1) / W ^ 2; 0.1e1 / W ^ 2 * R ^
2 * sin(theta) * cos(theta) * (mp * hc ^ 2 + I1 - I3) * (-Dpsir + 0.2e1 *
Dpsil) -R ^ 2 * Dpsil * (mp * hc ^ 2 + I1 - I3) * cos(theta) * sin(theta) / W
^ 2 0;];
G = [0 0 -hc * sin(theta) * mp * g]';

```

```

%% 2nd Condition

```

```

s = [Dpsir-Dpsir_d+lambda1*(psir-psir_d) Dpsil-Dpsil_d+lambda2*(psil-psil_d)
Dtheta-Dtheta_d+lambda3*(theta-theta_d)]';

```

```

DDtheta_d = (0.2e1 * R * (R * (mp * hc ^ 2 + I1 - I3) * (((-2 * Dpsir_d +
Dpsil_d) * Dpsil + Dpsir_d * Dpsir) * N - (theta - theta_d) * (-Dpsir +
Dpsil) * lambda3 + Dpsil * Dtheta_d - Dpsir * Dtheta_d - Dtheta * (Dpsil_d -
Dpsir_d) * sin(theta) + hc * W ^ 2 * ((lambda2 * Dpsil - lambda2 * Dpsil_d -
DDpsir_d - DDpsil_d - lambda1 * Dpsir_d + lambda1 * Dpsir) * N + 2 * lambda3
* (Dtheta - Dtheta_d)) * mp / 0.2e1) * cos(theta) + 0.2e1 * ((Dpsil_d -
Dpsir_d) * (-Dpsir + Dpsil) * R ^ 3 - Dtheta * (lambda3 * (theta - theta_d) -
Dtheta_d) * W ^ 2 * R + g * W ^ 2 * N) * hc * mp * sin(theta) + 0.2e1 * W ^ 2
* (((3 * mw) + mp) * (lambda2 * Dpsil - lambda2 * Dpsil_d - DDpsir_d -
DDpsil_d - lambda1 * Dpsir_d + lambda1 * Dpsir) * R ^ 2 / 0.2e1 + N * hc ^ 2
* lambda3 * (Dtheta - Dtheta_d) * mp + (((I2 * Dtheta - I2 * Dtheta_d + Km33
* (theta - theta_d)) * lambda3 + Km31 * Dpsir - Km31 * Dpsir_d + Km32 * Dpsil
- Km32 * Dpsil_d + Km33 * Dtheta - Km33 * Dtheta_d) * N) + ((theta - theta_d)
* (Km23 + Km13) * lambda3) + ((Km22 + Km12) * Dpsil) + ((Km21 + Km11) *
Dpsir) + ((-Km21 - Km11) * Dpsir_d) + ((Km23 + Km13) * Dtheta) + ((-Km22 -
Km12) * Dpsil_d) - (Dtheta_d * (Km23 + Km13)))) / W ^ 2 / (R * mp * hc *
cos(theta) + N * (I2 + mp * hc ^ 2)) / 0.2e1;

```

```

DDq_r = [DDpsir_d-lambda1*(Dpsir-Dpsir_d) DDpsil_d-lambda2*(Dpsil-Dpsil_d)
DDtheta_d-lambda3*(Dtheta-Dtheta_d)]';
Dq_r = [Dpsir_d-lambda1*(psir-psir_d) Dpsil_d-lambda2*(psil-psil_d) Dtheta_d-
lambda3*(theta-theta_d)]';

```

```

F3 = M*DDq_r+C*Dq_r+G-Km*s;
tau = F3;

```

```

% check: tau(3)+(tau(1)+tau(2))/N

```

```

Dq = [Dpsir Dpsil Dtheta]';
DDq = M\(tau-C*Dq-G);
V = .5*s'*M*s;
% check1 = s'*(Mdot/2-C)*s;
check1=0;
% check2 = s'*(tau-C*Dq_r-G-M*DDq_r);
check2 =0;
Vdot = check1 + check2;
v = (Dpsir+Dpsil)*R/2;
% Dphi = R/W*(Dpsir-Dpsil);
phi0 = u(20);
phi = R/W*(psir-psil)+phi0;

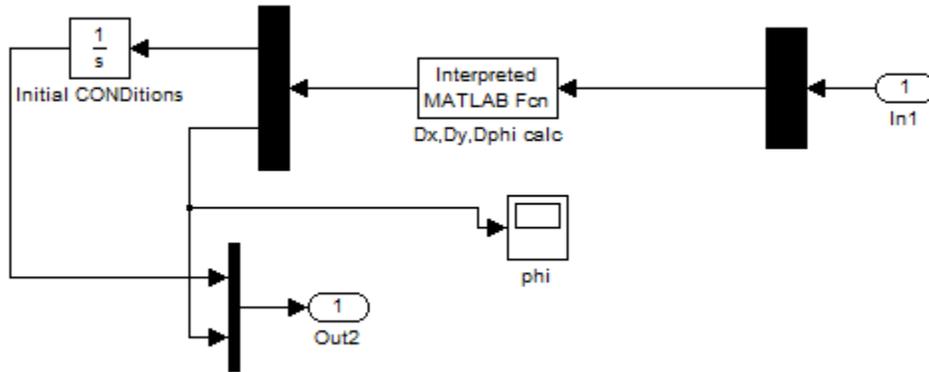
```

```

y = [DDq' V Vdot tau' 0 check1 check2 s' v phi]';

```

NonHolonomic Constraint



The *NonHolonomic Constraint* subsystem inputs the actual translational velocity and heading direction. It then calculates the velocities in the *X* and *Y* directions. The velocities are integrated to determine the actual position that is in turn used by the *Navigational Control*.

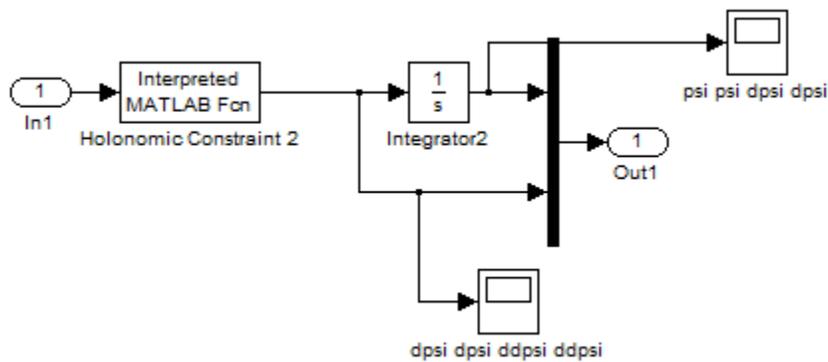
This shows the m-file used in the *NonHolonomic Constraint* subsystem.

```
function y = NonHolonomic_Constraint(u)
```

```
v = u(1);  
phi = u(2);
```

```
Dx = v*cos(phi);  
Dy = v*sin(phi);  
y = [Dx Dy phi]';
```

psicalc_d Subsystem



This subsystem inputs the desired translational and rotational velocities from the *Navigational Control*, and maps them to state coordinates that can be used by the sliding mode control. The main m-file follows.

```
function y = NonHolonomic_Constraint2(u)

R = 4*2.54/100;
W = 15/100;

v = u(1);
Dv = u(2);
Dphi = u(3);
DDphi = u(4);

mapper = [1/R W/(2*R);1/R -W/(2*R)];

Dpsis = mapper*[v Dphi]';
DDpsis = mapper*[Dv DDphi]';

y = [Dpsis;DDpsis]';
```

Appendix C - Implementation

Appendix C first contains the derivation of the actual Segbot dynamics and the non-linear control law values used. Next the Labview VI used for implementing the tracking control are presented. The Labview VI consists of three main parts: initialization, a data collection loop, and a timed loop. Parts of a previous project by Sunaina, K. (2013, August 1) was borrowed to implement the MPU-6050 accelerometer and gyro.

Maple File for Dynamics and Sliding Mode Control Values

Derive Lagrangian

Kinetic Energy of the Wheels and Motor

The "Dphi inertia contribution" for the wheels is included in Iphi for the pendulum body.

$$> KEwr := \frac{1}{2} \cdot (Jwm) \cdot Dpsir^2 + \frac{1}{2} \cdot (mw \cdot R^2) \cdot Dpsir^2;$$

$$KEwr := \frac{1}{2} Jwm Dpsir^2 + \frac{1}{2} mw R^2 Dpsir^2$$

$$> KEwl := \frac{1}{2} \cdot (Jwm) \cdot Dpsil^2 + \frac{1}{2} \cdot (mw \cdot R^2) \cdot Dpsil^2;$$

$$KEwl := \frac{1}{2} Jwm Dpsil^2 + \frac{1}{2} mw R^2 Dpsil^2$$

> # position of base center of mass w.r.t. axel center

$$rp := \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} + \begin{bmatrix} hc \cdot \sin(\theta) \cdot \cos(\phi) \\ hc \cdot \sin(\theta) \cdot \sin(\phi) \\ R + hc \cdot \cos(\theta) \end{bmatrix}$$

$$rp := \begin{bmatrix} x + hc \sin(\theta) \cos(\phi) \\ y + hc \sin(\theta) \sin(\phi) \\ R + hc \cos(\theta) \end{bmatrix}$$

> ## pendulum center of mass velocities

$$rp := \text{diff}~(\%, x) \cdot Dx + \text{diff}~(\%, y) \cdot Dy + \text{diff}~(\%, \theta) \cdot Dtheta + \text{diff}~(\%, \phi) \cdot Dphi :$$

$$Drp := \%$$

$$Drp := \begin{bmatrix} Dx + Dtheta \, hc \cos(\theta) \cos(\phi) - Dphi \, hc \sin(\theta) \sin(\phi) \\ Dy + Dtheta \, hc \cos(\theta) \sin(\phi) + Dphi \, hc \sin(\theta) \cos(\phi) \\ -Dtheta \, hc \sin(\theta) \end{bmatrix}$$

Find pendulum body centered rotational velocities:

body = Map * Global

$$> Rtheta := \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} :$$

$$> Omega_Pendulum := Rtheta \cdot \begin{bmatrix} 0 \\ 0 \\ Dphi \end{bmatrix} + \begin{bmatrix} 0 \\ Dtheta \\ 0 \end{bmatrix}$$

$$Omega_Pendulum := \begin{bmatrix} -\sin(\theta) \, Dphi \\ Dtheta \\ \cos(\theta) \, Dphi \end{bmatrix}$$

$$> Pend_Rotational := \frac{1}{2} \cdot (I1 \cdot (Omega_Pendulum(1))^2 + I2 \cdot (Omega_Pendulum(2))^2 + I3 \cdot (Omega_Pendulum(3))^2)$$

$$Pend_Rotational := \frac{1}{2} I1 \sin(\theta)^2 Dphi^2 + \frac{1}{2} I2 Dtheta^2 + \frac{1}{2} I3 \cos(\theta)^2 Dphi^2$$

$$> \text{## Pend Translational KE + Rotational KE} \\ \frac{1}{2} \cdot (mp \cdot (Drp(1)^2 + Drp(2)^2 + Drp(3)^2)) + Pend_Rotational :$$

KEpen := %

$$KEpen := \frac{1}{2} mp \left((Dx + Dtheta \, hc \cos(\theta) \cos(\phi) - Dphi \, hc \sin(\theta) \sin(\phi))^2 + (Dy + Dtheta \, hc \cos(\theta) \sin(\phi) + Dphi \, hc \sin(\theta) \cos(\phi))^2 + Dtheta^2 \, hc^2 \sin(\theta)^2 \right) + \frac{1}{2} I1 \sin(\theta)^2 Dphi^2 + \frac{1}{2} I2 Dtheta^2 + \frac{1}{2} I3 \cos(\theta)^2 Dphi^2$$

> ## variable PEPE := hc*cos(theta)*mp*g

$$PE := hc \cos(\theta) mp g$$

KE of the motor is now included in the kinetic energy of the wheel.

Build the Lagrangian

>

$KE_{wl} + KE_{wr} + KE_{pen} - PE :$

NON HOLONOMIC CONSTRAINTS: CAN NOT BE SUBSTITUTED YET

Holonomic Constraints: Can be substituted

$subs\left(Dphi = \frac{R \cdot (Dpsir - Dpsil)}{W}, \%\right) :$

$subs\left(phi = \frac{R \cdot (psir - psil)}{W}, \%\right) :$

$L := \% : simplify(\%, size)$

$$\begin{aligned} & \frac{1}{2} J_{wm} Dpsil^2 + \frac{1}{2} m_w R^2 Dpsil^2 + \frac{1}{2} J_{wm} Dpsir^2 + \frac{1}{2} m_w R^2 Dpsir^2 + \frac{1}{2} m_p \left(\left(D_x \right. \right. \\ & \quad \left. \left. + Dtheta hc \cos(\theta) \cos\left(\frac{R(psir - psil)}{W}\right) \right. \right. \\ & \quad \left. \left. - \frac{R(Dpsir - Dpsil) hc \sin(\theta) \sin\left(\frac{R(psir - psil)}{W}\right)}{W} \right)^2 + \left(D_y \right. \right. \\ & \quad \left. \left. + Dtheta hc \cos(\theta) \sin\left(\frac{R(psir - psil)}{W}\right) \right. \right. \\ & \quad \left. \left. + \frac{R(Dpsir - Dpsil) hc \sin(\theta) \cos\left(\frac{R(psir - psil)}{W}\right)}{W} \right)^2 + Dtheta^2 hc^2 \sin^2(\theta) \right)^2 \\ & + \frac{1}{2} \frac{I_1 \sin^2(\theta) R^2 (Dpsir - Dpsil)^2}{W^2} + \frac{1}{2} I_2 Dtheta^2 \\ & + \frac{1}{2} \frac{I_3 \cos^2(\theta) R^2 (Dpsir - Dpsil)^2}{W^2} - hc \cos(\theta) mp g \end{aligned}$$

Apply Lagrange's Equation

First 5 equations with respect to generalized coordinates.

>

$(diff(L, Dx, Dx)) * DDx + (diff(L, Dx, Dy)) * DDy + (diff(L, Dx, Dpsir)) * DDpsir$
 $+ (diff(L, Dx, Dpsil)) * DDpsil + (diff(L, Dx, Dtheta)) * DDtheta + (diff(L, Dx,$
 $Dphi)) * DDphi + (diff(L, Dx, x)) * Dx + (diff(L, Dx, y)) * Dy + (diff(L, Dx,$
 $psir)) * Dpsir + (diff(L, Dx, psil)) * Dpsil + (diff(L, Dx, theta)) * Dtheta$
 $+ (diff(L, Dx, phi)) * Dphi - diff(L, x) - lambda[1] :$
 $eq1 := \% : simplify(\%, size)$

$$\begin{aligned} & \frac{1}{W^2} \left(-mp \left((Dtheta^2 W^2 + R^2 (-Dpsir + Dpsil)^2) \sin(\theta) \right. \right. \\ & \quad \left. \left. - \cos(\theta) DDtheta W^2 \right) hc \cos\left(\frac{R (psir - psil)}{W}\right) + \left(R \left((-DDpsir \right. \right. \right. \\ & \quad \left. \left. + DDpsil) \sin(\theta) + 2 Dtheta \cos(\theta) (-Dpsir + Dpsil) \right) \right. \\ & \quad \left. mp hc \sin\left(\frac{R (psir - psil)}{W}\right) - (-mp DDx + \lambda_1) W \right) W \end{aligned}$$

>

$$\begin{aligned} & (diff(L, Dy, Dx)) * DDx + (diff(L, Dy, Dy)) * DDy + (diff(L, Dy, Dpsir)) * DDpsir \\ & + (diff(L, Dy, Dpsil)) * DDpsil + (diff(L, Dy, Dtheta)) * DDtheta + (diff(L, Dy, \\ & Dphi)) * DDphi + (diff(L, Dy, x)) * Dx + (diff(L, Dy, y)) * Dy + (diff(L, Dy, \\ & psir)) * Dpsir + (diff(L, Dy, psil)) * Dpsil + (diff(L, Dy, theta)) * Dtheta \\ & + (diff(L, Dy, phi)) * Dphi - diff(L, y) - lambda[2] : \\ eq2 := \% : simplify(\%, size) \end{aligned}$$

$$\begin{aligned} & \frac{1}{W^2} \left(-hc \left((Dtheta^2 W^2 + R^2 (-Dpsir + Dpsil)^2) \sin(\theta) \right. \right. \\ & \quad \left. \left. - \cos(\theta) DDtheta W^2 \right) mp \sin\left(\frac{R (psir - psil)}{W}\right) - W \left(hc \left((-DDpsir \right. \right. \right. \\ & \quad \left. \left. + DDpsil) \sin(\theta) + 2 Dtheta \cos(\theta) (-Dpsir + Dpsil) \right) \right. \\ & \quad \left. mp R \cos\left(\frac{R (psir - psil)}{W}\right) + W (-mp DDy + \lambda_2) \right) \end{aligned}$$

>

$$\begin{aligned} & (diff(L, Dpsir, Dx)) * DDx + (diff(L, Dpsir, Dy)) * DDy + (diff(L, Dpsir, Dpsir)) \\ & * DDpsir + (diff(L, Dpsir, Dpsil)) * DDpsil + (diff(L, Dpsir, Dtheta)) * DDtheta \\ & + (diff(L, Dpsir, Dphi)) * DDphi + (diff(L, Dpsir, x)) * Dx + (diff(L, Dpsir, y)) \\ & * Dy + (diff(L, Dpsir, psir)) * Dpsir + (diff(L, Dpsir, psil)) * Dpsil + (diff(L, \\ & Dpsir, theta)) * Dtheta + (diff(L, Dpsir, phi)) * Dphi - diff(L, psir) + \frac{R}{2} \\ & \cdot (\lambda[1] \cdot \cos(\phi) + \lambda[2] \cdot \sin(\phi)) + bwm \cdot (Dpsir) - taur : \\ eq3 := \% : simplify(\%, size) \end{aligned}$$

$$\begin{aligned} & \frac{1}{2} \frac{1}{W^2} \left(-4 \sin(\theta) \left(\left(-\frac{1}{2} DDpsir + \frac{1}{2} DDpsil \right) \sin(\theta) + Dtheta \cos(\theta) (-Dpsir \right. \right. \\ & \quad \left. \left. + Dpsil) \right) R^2 hc^2 mp \cos\left(\frac{R (psir - psil)}{W}\right)^2 \right. \\ & \quad \left. + 2 mp R hc \sin(\theta) \cos\left(\frac{R (psir - psil)}{W}\right) DDy W - 4 \sin(\theta) \left(\left(-\frac{1}{2} DDpsir \right. \right. \right. \\ & \quad \left. \left. + \frac{1}{2} DDpsil \right) \sin(\theta) + Dtheta \cos(\theta) (-Dpsir + Dpsil) \right) \right. \\ & \quad \left. R^2 hc^2 mp \sin\left(\frac{R (psir - psil)}{W}\right)^2 \right. \\ & \quad \left. - 2 mp R hc \sin(\theta) \sin\left(\frac{R (psir - psil)}{W}\right) DDx W - 2 R^2 II (-DDpsir \right. \\ & \quad \left. + DDpsil) \sin(\theta)^2 - 4 R^2 Dtheta \cos(\theta) (II - I3) (-Dpsir + Dpsil) \sin(\theta) \right. \\ & \quad \left. - 2 R^2 I3 (-DDpsir + DDpsil) \cos(\theta)^2 + 2 W^2 \left(\frac{1}{2} \sin(\phi) \lambda_2 R + Jwm DDpsir \right. \right. \\ & \quad \left. \left. - taur + DDpsir mw R^2 + bwm Dpsir + \frac{1}{2} \cos(\phi) \lambda_1 R \right) \right) \end{aligned}$$

>

$$\begin{aligned}
& (\text{diff}(L, Dpsil, Dx)) * DDx + (\text{diff}(L, Dpsil, Dy)) * DDy + (\text{diff}(L, Dpsil, Dpsir)) \\
& * DDpsir + (\text{diff}(L, Dpsil, Dpsil)) * DDpsil + (\text{diff}(L, Dpsil, Dtheta)) * DDtheta \\
& + (\text{diff}(L, Dpsil, Dphi)) * DDphi + (\text{diff}(L, Dpsil, x)) * Dx + (\text{diff}(L, Dpsil, y)) \\
& * Dy + (\text{diff}(L, Dpsil, psir)) * Dpsir + (\text{diff}(L, Dpsil, psil)) * Dpsil + (\text{diff}(L, \\
& Dpsil, theta)) * Dtheta + (\text{diff}(L, Dpsil, phi)) * Dphi - \text{diff}(L, psil) + \frac{R}{2} \\
& \cdot (\text{lambda}[1] \cdot \cos(\text{phi}) + \text{lambda}[2] \cdot \sin(\text{phi})) + \text{bwm} \cdot (Dpsil) - \text{taul} : \\
\text{eq4} := \% : \text{simplify}(\%, \text{size})
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2} \frac{1}{W^2} \left(4 \sin(\theta) \left(\left(-\frac{1}{2} DDpsir + \frac{1}{2} DDpsil \right) \sin(\theta) + Dtheta \cos(\theta) (-Dpsir \right. \right. \\
& \left. \left. + Dpsil) \right) R^2 hc^2 mp \cos\left(\frac{R(psir - psil)}{W}\right)^2 \\
& - 2 mp R hc \sin(\theta) \cos\left(\frac{R(psir - psil)}{W}\right) DDy W + 4 \sin(\theta) \left(\left(-\frac{1}{2} DDpsir \right. \right. \\
& \left. \left. + \frac{1}{2} DDpsil \right) \sin(\theta) + Dtheta \cos(\theta) (-Dpsir + Dpsil) \right) \\
& R^2 hc^2 mp \sin\left(\frac{R(psir - psil)}{W}\right)^2 \\
& + 2 mp R hc \sin(\theta) \sin\left(\frac{R(psir - psil)}{W}\right) DDx W + 2 R^2 II (-DDpsir \\
& + DDpsil) \sin(\theta)^2 + 4 R^2 Dtheta \cos(\theta) (II - I3) (-Dpsir + Dpsil) \sin(\theta) \\
& + 2 R^2 I3 (-DDpsir + DDpsil) \cos(\theta)^2 + 2 W^2 \left(-\text{taul} + \frac{1}{2} \sin(\phi) \lambda_2 R \right. \\
& \left. + \text{bwm} Dpsil + Jwm DDpsil + \frac{1}{2} \cos(\phi) \lambda_1 R + DDpsil mw R^2 \right)
\end{aligned}$$

>

$$\begin{aligned}
& (\text{diff}(L, Dtheta, Dx)) * DDx + (\text{diff}(L, Dtheta, Dy)) * DDy + (\text{diff}(L, Dtheta, Dpsir)) \\
& * DDpsir + (\text{diff}(L, Dtheta, Dpsil)) * DDpsil + (\text{diff}(L, Dtheta, Dtheta)) \\
& * DDtheta + (\text{diff}(L, Dtheta, Dphi)) * DDphi + (\text{diff}(L, Dtheta, x)) * Dx \\
& + (\text{diff}(L, Dtheta, y)) * Dy + (\text{diff}(L, Dtheta, psir)) * Dpsir + (\text{diff}(L, Dtheta, \\
& psil)) * Dpsil + (\text{diff}(L, Dtheta, theta)) * Dtheta + (\text{diff}(L, Dtheta, phi)) * Dphi \\
& - \text{diff}(L, theta) + \frac{(\text{taul} + \text{taur})}{N} + \text{bwm} \cdot (-Dpsir - Dpsil) : \\
\text{eq5} := \% :
\end{aligned}$$

Non-Holonomic Constraints

>

$$\begin{aligned}
Cc1 & := Dx - \frac{R}{2} \cdot (Dpsir + Dpsil) \cdot \cos(\text{phi}); \\
Cc2 & := Dy - \frac{R}{2} \cdot (Dpsir + Dpsil) \cdot \sin(\text{phi});
\end{aligned}$$

$$Cc1 := Dx - \frac{1}{2} R (Dpsir + Dpsil) \cos(\phi)$$

$$Cc2 := Dy - \frac{1}{2} R (Dpsir + Dpsil) \sin(\phi)$$

Time Derivative of Non-Holonomic Constraints

$$\begin{aligned} > \text{diff}(Cc1, Dx) \cdot DDx + \text{diff}(Cc1, Dpsir) \cdot DDpsir + \text{diff}(Cc1, Dpsil) \cdot DDpsil \\ &+ \text{diff}(Cc1, phi) \cdot Dphi : eq7 := \% \end{aligned}$$

$$\begin{aligned} \text{diff}(Cc2, Dy) \cdot DDy + \text{diff}(Cc2, Dpsir) \cdot DDpsir + \text{diff}(Cc2, Dpsil) \cdot DDpsil \\ + \text{diff}(Cc2, phi) \cdot Dphi : eq8 := \% \end{aligned}$$

$$\begin{aligned} eq7 := DDx - \frac{1}{2} R \cos(\phi) DDpsir - \frac{1}{2} R \cos(\phi) DDpsil + \frac{1}{2} R (Dpsir \\ + Dpsil) \sin(\phi) Dphi \end{aligned}$$

$$\begin{aligned} eq8 := DDy - \frac{1}{2} R \sin(\phi) DDpsir - \frac{1}{2} R \sin(\phi) DDpsil - \frac{1}{2} R (Dpsir \\ + Dpsil) \cos(\phi) Dphi \end{aligned}$$

Solve for the Lagrange multipliers, and time derivatives of X and Y.

Use the constraint equations, time derivative of the constraint equations, and the Lagrange Equations w.r.t. x and y

$$\begin{aligned} > Lsol := \text{solve}(\{eq1=0, eq2=0, eq7=0, eq8=0, Cc1=0, Cc2=0\}, \{\text{lambda}[1], \\ \text{lambda}[2], DDx, DDy, Dx, Dy\}) : \end{aligned}$$

Finds the final 3 equations of motion:

Eliminate the Lagrange multipliers, velocity of x and y, acceleration of x and y.

$$\begin{aligned} > \begin{bmatrix} eq3 \\ eq4 \\ eq5 \end{bmatrix} : \\ \text{map}(\text{eval}, \% , Lsol) : \\ \text{subs}\left(Dphi = \frac{R \cdot (Dpsir - Dpsil)}{W}, \% \right) : \\ \text{subs}\left(phi = \frac{R \cdot (psir - psil)}{W}, \% \right) : \\ \text{simplify}(\%, \text{trig}) : \\ eq1f := \% (1); \\ eq2f := \% (2); \\ eq3f := \% (3); \end{aligned}$$

$$\begin{aligned}
eq1f := & \frac{1}{4} \frac{1}{W^2} (4 bwm Dpsir W^2 + 4 DDpsir Jwm W^2 - 4 taur W^2 + 4 DDpsir mw R^2 W \\
& + 4 DDpsir I3 \cos(\theta)^2 R^2 - 4 R^2 DDpsil I3 \cos(\theta)^2 \\
& + 8 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsir - 8 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsil \\
& - 8 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsir + 8 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsil \\
& - 4 mp R^3 hc \sin(\theta) Dpsil^2 - 4 DDpsir mp R^2 hc^2 \cos(\theta)^2 \\
& + 4 R^2 DDpsil mp hc^2 \cos(\theta)^2 - 4 DDpsir I1 R^2 \cos(\theta)^2 + 4 R^2 DDpsil I1 \cos(\theta)^2 \\
& - 8 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsil \\
& + 8 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsir - 2 R mp hc Dtheta^2 W^2 \sin(\theta) \\
& + 4 R^3 mp hc Dpsir \sin(\theta) Dpsil + 2 R mp hc \cos(\theta) DDtheta W^2 \\
& + 4 DDpsir mp R^2 hc^2 - 4 R^2 DDpsil mp hc^2 - 4 R^2 DDpsil I1 + 4 DDpsir I1 R^2 \\
& + R^2 mp W^2 DDpsil + R^2 mp W^2 DDpsir)
\end{aligned}$$

$$\begin{aligned}
eq2f := & \frac{1}{4} \frac{1}{W^2} (-4 DDpsir I3 \cos(\theta)^2 R^2 + 4 R^2 DDpsil I3 \cos(\theta)^2 \\
& - 8 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsir + 8 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsil \\
& + 8 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsir - 8 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsil \\
& + 4 DDpsil mw R^2 W^2 + 4 bwm Dpsil W^2 + 4 DDpsil Jwm W^2 - 4 taul W^2 \\
& + 4 DDpsir mp R^2 hc^2 \cos(\theta)^2 - 4 R^2 DDpsil mp hc^2 \cos(\theta)^2 \\
& + 4 DDpsir I1 R^2 \cos(\theta)^2 - 4 R^2 DDpsil I1 \cos(\theta)^2 \\
& + 8 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsil \\
& - 8 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsir - 2 R mp hc Dtheta^2 W^2 \sin(\theta) \\
& + 4 R^3 mp hc Dpsir \sin(\theta) Dpsil + 2 R mp hc \cos(\theta) DDtheta W^2 \\
& - 4 DDpsir mp R^2 hc^2 + 4 R^2 DDpsil mp hc^2 + 4 R^2 DDpsil I1 - 4 DDpsir I1 R^2 \\
& + R^2 mp W^2 DDpsil - 4 mp R^3 hc \sin(\theta) Dpsir^2 + R^2 mp W^2 DDpsir)
\end{aligned}$$

$$\begin{aligned}
eq3f := & \frac{1}{2} \frac{1}{W^2 N} (2 taur W^2 + 4 mp hc^2 \sin(\theta) N R^2 Dpsir \cos(\theta) Dpsil + 2 taul W^2 \\
& - 2 mp hc^2 \sin(\theta) N R^2 Dpsir^2 \cos(\theta) + mp hc \cos(\theta) R W^2 N DDpsir \\
& - 2 mp hc^2 \sin(\theta) N R^2 Dpsil^2 \cos(\theta) + mp hc \cos(\theta) R W^2 N DDpsil \\
& + 2 DDtheta W^2 N I2 - 2 bwm W^2 N Dpsir - 2 bwm W^2 N Dpsil \\
& - 2 I1 \sin(\theta) R^2 \cos(\theta) N Dpsir^2 - 2 I1 \sin(\theta) R^2 \cos(\theta) N Dpsil^2 \\
& + 2 I3 \cos(\theta) R^2 \sin(\theta) N Dpsir^2 + 2 I3 \cos(\theta) R^2 \sin(\theta) N Dpsil^2 \\
& - 2 hc \sin(\theta) mp g W^2 N + 2 DDtheta W^2 N mp hc^2 \\
& + 4 I1 \sin(\theta) R^2 \cos(\theta) N Dpsir Dpsil - 4 I3 \cos(\theta) R^2 \sin(\theta) N Dpsir Dpsil)
\end{aligned}$$

Simplify the Equations

$$\begin{array}{l}
> \left[\begin{array}{ccc}
coeff(eq1f, DDpsir) & coeff(eq1f, DDpsil) & coeff(eq1f, DDtheta) \\
coeff(eq2f, DDpsir) & coeff(eq2f, DDpsil) & coeff(eq2f, DDtheta) \\
coeff(eq3f, DDpsir) & coeff(eq3f, DDpsil) & coeff(eq3f, DDtheta)
\end{array} \right] : simplify~(%, \\
size) :
\end{array}$$

M := %

$$\begin{aligned}
M := & \left[\left[\frac{1}{4} \frac{1}{W^2} (-4 R^2 (-I3 + mp hc^2 + I1) \cos(\theta)^2 + ((4 mw + mp) W^2 + 4 I1 \right. \right. \\
& \left. \left. + 4 mp hc^2) R^2 + 4 Jwm W^2), \right. \right. \\
& \left. \frac{1}{4} \frac{((-4 I3 + 4 mp hc^2 + 4 I1) \cos(\theta)^2 + (W^2 - 4 hc^2) mp - 4 I1) R^2}{W^2}, \right. \\
& \left. \frac{1}{2} R mp hc \cos(\theta) \right], \\
& \left[\frac{1}{4} \frac{((-4 I3 + 4 mp hc^2 + 4 I1) \cos(\theta)^2 + (W^2 - 4 hc^2) mp - 4 I1) R^2}{W^2}, \right. \\
& \left. \frac{1}{4} \frac{1}{W^2} (-4 R^2 (-I3 + mp hc^2 + I1) \cos(\theta)^2 + ((4 mw + mp) W^2 + 4 I1 \right. \right. \\
& \left. \left. + 4 mp hc^2) R^2 + 4 Jwm W^2), \frac{1}{2} R mp hc \cos(\theta) \right], \\
& \left[\frac{1}{2} R mp hc \cos(\theta), \frac{1}{2} R mp hc \cos(\theta), I2 + mp hc^2 \right] \Big]
\end{aligned}$$

> $M - M^{%T}$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

> $\begin{bmatrix} eq1f \\ eq2f \\ eq3f \end{bmatrix} : \text{coeff}(\%, g) \cdot g : G := \%$

$$G := \begin{bmatrix} 0 \\ 0 \\ -hc \sin(\theta) mp g \end{bmatrix}$$

C vector

> $\begin{bmatrix} eq1f \\ eq2f \\ eq3f \end{bmatrix} - M \cdot \begin{bmatrix} DDpsir \\ DDpsil \\ DDtheta \end{bmatrix} :$

$eval(\%, \{g=0, taur=0, taul=0\}) :$

$factor(\%) :$

$Cv := \%$

$$\begin{aligned}
Cv := & \left[\left[-\frac{1}{2} \frac{1}{W^2} (-2 bwm Dpsir W^2 - 4 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsir \right. \right. \\
& + 4 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsil + 4 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsir \\
& - 4 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsil - 2 R^3 mp hc Dpsir \sin(\theta) Dpsil \\
& + R mp hc Dtheta^2 W^2 \sin(\theta) - 4 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsir \\
& \left. + 4 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsil + 2 mp R^3 hc \sin(\theta) Dpsil^2 \right], \\
& \left[\frac{1}{2} \frac{1}{W^2} (-4 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsir \right. \\
& + 4 R^2 \cos(\theta) Dtheta I1 \sin(\theta) Dpsil + 4 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsir \\
& - 4 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsil + 2 bwm Dpsil W^2 \\
& + 2 R^3 mp hc Dpsir \sin(\theta) Dpsil - R mp hc Dtheta^2 W^2 \sin(\theta) \\
& - 2 mp R^3 hc \sin(\theta) Dpsir^2 - 4 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsir \\
& \left. + 4 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsil \right], \\
& \left[-\frac{1}{W^2} (-2 mp hc^2 \sin(\theta) R^2 Dpsir \cos(\theta) Dpsil \right. \\
& + mp hc^2 \sin(\theta) R^2 Dpsir^2 \cos(\theta) + mp hc^2 \sin(\theta) R^2 Dpsil^2 \cos(\theta) \\
& + bwm Dpsir W^2 + bwm Dpsil W^2 + I1 \sin(\theta) R^2 \cos(\theta) Dpsir^2 \\
& + I1 \sin(\theta) R^2 \cos(\theta) Dpsil^2 - I3 \cos(\theta) R^2 \sin(\theta) Dpsir^2 \\
& - I3 \cos(\theta) R^2 \sin(\theta) Dpsil^2 - 2 I1 \sin(\theta) R^2 \cos(\theta) Dpsir Dpsil \\
& \left. + 2 I3 \cos(\theta) R^2 \sin(\theta) Dpsir Dpsil \right] \right]
\end{aligned}$$

For the control law we want $(\dot{M} - 2*C)$ to be skew- symmetric. This determines the terms in the C matrix.

> $M: \text{diff}(\%, \theta) \cdot Dtheta : Mdot := \%$

$$\begin{aligned}
Mdot := & \left[\left[\frac{2 Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta)}{W^2}, \right. \right. \\
& -\frac{1}{2} \frac{Dtheta (-4 I3 + 4 mp hc^2 + 4 I1) \cos(\theta) \sin(\theta) R^2}{W^2}, \\
& \left. -\frac{1}{2} Dtheta R mp hc \sin(\theta) \right], \\
& \left[-\frac{1}{2} \frac{Dtheta (-4 I3 + 4 mp hc^2 + 4 I1) \cos(\theta) \sin(\theta) R^2}{W^2}, \right. \\
& \left. \frac{2 Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta)}{W^2}, -\frac{1}{2} Dtheta R mp hc \sin(\theta) \right], \\
& \left[-\frac{1}{2} Dtheta R mp hc \sin(\theta), -\frac{1}{2} Dtheta R mp hc \sin(\theta), 0 \right] \right]
\end{aligned}$$

Make $\dot{M} - 2*C$ Skew-Symmetric

> $C1 := \begin{bmatrix} c11 & c12 & c13 \\ c21 & c22 & c23 \\ c31 & c32 & c33 \end{bmatrix} :$

> Mdot - 2 · C1 : asm := %

$$asm := \left[\left[\frac{2 Dtheta R^2 (-I3 + mp hc^2 + II) \cos(\theta) \sin(\theta)}{W^2} - 2 c11, \right. \right. \\ \left. \left. - \frac{1}{2} \frac{Dtheta (-4 I3 + 4 mp hc^2 + 4 II) \cos(\theta) \sin(\theta) R^2}{W^2} - 2 c12, \right. \right. \\ \left. \left. - \frac{1}{2} Dtheta R mp hc \sin(\theta) - 2 c13 \right], \right. \\ \left[- \frac{1}{2} \frac{Dtheta (-4 I3 + 4 mp hc^2 + 4 II) \cos(\theta) \sin(\theta) R^2}{W^2} - 2 c21, \right. \\ \left. \frac{2 Dtheta R^2 (-I3 + mp hc^2 + II) \cos(\theta) \sin(\theta)}{W^2} - 2 c22, \right. \\ \left. - \frac{1}{2} Dtheta R mp hc \sin(\theta) - 2 c23 \right], \\ \left[- \frac{1}{2} Dtheta R mp hc \sin(\theta) - 2 c31, - \frac{1}{2} Dtheta R mp hc \sin(\theta) - 2 c32, - 2 c33 \right] \\ \left. \right]$$

> Dq := $\begin{bmatrix} Dpsir \\ Dpsil \\ Dtheta \end{bmatrix}$:

> C1.Dq - Cv : eval(% , bwm = 0) : Ccon := %

$$Ccon := \left[\left[c11 Dpsir + c12 Dpsil + c13 Dtheta + \frac{1}{2} \frac{1}{W^2} (\right. \right. \\ - 4 R^2 \cos(\theta) Dtheta II \sin(\theta) Dpsir + 4 R^2 \cos(\theta) Dtheta II \sin(\theta) Dpsil \\ + 4 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsir - 4 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsil \\ - 2 R^3 mp hc Dpsir \sin(\theta) Dpsil + R mp hc Dtheta^2 W^2 \sin(\theta) \\ - 4 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsir \\ + 4 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsil + 2 mp R^3 hc \sin(\theta) Dpsil^2 \left. \right) \right], \\ \left[c21 Dpsir + c22 Dpsil + c23 Dtheta - \frac{1}{2} \frac{1}{W^2} (\right. \\ - 4 R^2 \cos(\theta) Dtheta II \sin(\theta) Dpsir + 4 R^2 \cos(\theta) Dtheta II \sin(\theta) Dpsil \\ + 4 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsir - 4 R^2 \cos(\theta) Dtheta I3 \sin(\theta) Dpsil \\ + 2 R^3 mp hc Dpsir \sin(\theta) Dpsil - R mp hc Dtheta^2 W^2 \sin(\theta) \\ - 2 mp R^3 hc \sin(\theta) Dpsir^2 - 4 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsir \\ + 4 R^2 \cos(\theta) Dtheta mp hc^2 \sin(\theta) Dpsil \left. \right) \right], \\ \left[c31 Dpsir + c32 Dpsil + c33 Dtheta + \frac{1}{W^2} (\right. \\ - 2 mp hc^2 \sin(\theta) R^2 Dpsir \cos(\theta) Dpsil + mp hc^2 \sin(\theta) R^2 Dpsir^2 \cos(\theta) \\ + mp hc^2 \sin(\theta) R^2 Dpsil^2 \cos(\theta) + II \sin(\theta) R^2 \cos(\theta) Dpsir^2 \\ + II \sin(\theta) R^2 \cos(\theta) Dpsil^2 - I3 \cos(\theta) R^2 \sin(\theta) Dpsir^2 \\ - I3 \cos(\theta) R^2 \sin(\theta) Dpsil^2 - 2 II \sin(\theta) R^2 \cos(\theta) Dpsir Dpsil \\ + 2 I3 \cos(\theta) R^2 \sin(\theta) Dpsir Dpsil \left. \right) \right]$$

> solve({asm(1, 1) = 0, asm(2, 2) = 0, asm(3, 3) = 0, asm(1, 2) = -asm(2, 1), asm(1, 3) = -asm(3, 1), asm(2, 3) = -asm(3, 2), Ccon(1) = 0, Ccon(2) = 0, Ccon(3) = 0}, {c11, c12, c13, c21, c22, c23, c31, c32, c33}) :
Csol1 := %

$$\begin{aligned}
Csol1 := & \left\{ c11 = \frac{Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta)}{W^2}, c12 = \right. \\
& - \frac{1}{Dpsir W^2} (R^2 \sin(\theta) (Dtheta \cos(\theta) mp hc^2 Dpsil + Dtheta \cos(\theta) I1 Dpsil \\
& - Dtheta \cos(\theta) I3 Dpsil - R mp hc Dpsir^2 + R mp hc Dpsir Dpsil)) \\
& - \frac{Dtheta c32}{Dpsir}, c13 = \frac{1}{2} \frac{1}{Dpsir W^2} (\sin(\theta) R (-Dtheta mp hc Dpsir W^2 \\
& - 4 R \cos(\theta) mp hc^2 Dpsir Dpsil + 2 R \cos(\theta) mp hc^2 Dpsir^2 \\
& + 2 R \cos(\theta) mp hc^2 Dpsil^2 + 2 R \cos(\theta) I1 Dpsir^2 + 2 R \cos(\theta) I1 Dpsil^2 \\
& - 2 R \cos(\theta) I3 Dpsir^2 - 2 R \cos(\theta) I3 Dpsil^2 - 4 R \cos(\theta) I1 Dpsir Dpsil \\
& + 4 R \cos(\theta) I3 Dpsir Dpsil) + \frac{Dpsil c32}{Dpsir}, c21 = \frac{Dtheta c32}{Dpsir} \\
& + \frac{1}{Dpsir W^2} (R^2 \sin(\theta) (-2 \cos(\theta) Dtheta mp hc^2 Dpsir \\
& + Dtheta \cos(\theta) mp hc^2 Dpsil + Dtheta \cos(\theta) I1 Dpsil - Dtheta \cos(\theta) I3 Dpsil \\
& - 2 \cos(\theta) Dtheta I1 Dpsir + 2 \cos(\theta) Dtheta I3 Dpsir - R mp hc Dpsir^2 \\
& + R mp hc Dpsir Dpsil)), c22 = \frac{Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta)}{W^2}, \\
c23 = & - \frac{1}{2} Dtheta R mp hc \sin(\theta) - c32, c31 = - \frac{Dpsil c32}{Dpsir} \\
& - \frac{1}{Dpsir W^2} (R^2 \sin(\theta) \cos(\theta) (-2 mp hc^2 Dpsir Dpsil + mp hc^2 Dpsir^2 \\
& + mp hc^2 Dpsil^2 + I1 Dpsir^2 + I1 Dpsil^2 - I3 Dpsir^2 - I3 Dpsil^2 - 2 I1 Dpsir Dpsil \\
& + 2 I3 Dpsir Dpsil)), c32 = c32, c33 = 0 \}
\end{aligned}$$

> C1 : eval(% , Csol1) : simplify(% , size) : C := % : simplify~(% , size)

$$\left[\left[\frac{Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta)}{W^2}, \frac{1}{Dpsir W^2} (- (Dtheta Dpsil (-I3 + mp hc^2 + I1) \cos(\theta) + R hc mp Dpsir (-Dpsir + Dpsil)) R^2 \sin(\theta) - Dtheta W^2 c32), \frac{1}{2} \frac{1}{Dpsir W^2} \left(2 \left(R (-Dpsir + Dpsil)^2 (-I3 + mp hc^2 + I1) \cos(\theta) - \frac{1}{2} Dtheta mp hc Dpsir W^2 \right) R \sin(\theta) + 2 c32 Dpsil W^2 \right) \right], \left[\frac{1}{Dpsir W^2} ((Dtheta (-2 Dpsir + Dpsil) (-I3 + mp hc^2 + I1) \cos(\theta) + R hc mp Dpsir (-Dpsir + Dpsil)) R^2 \sin(\theta) + Dtheta W^2 c32), \frac{Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta)}{W^2}, -\frac{1}{2} Dtheta R mp hc \sin(\theta) - c32 \right], \left[\frac{-R^2 \sin(\theta) (-Dpsir + Dpsil)^2 (-I3 + mp hc^2 + I1) \cos(\theta) - c32 Dpsil W^2}{Dpsir W^2}, c32, 0 \right] \right]$$

Show skew-symmetric.

> `Mdot - 2 * C : % + %T : simplify(%)`

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The Dpsir in the denominator of some C elements can present a problem if is it ever 0. Get it out of the denominator.

> `C : coeff~(%, 1/Dpsir); isolate%(1, 2), c32); eval(%%, %)`

$$\left[\left[0, \frac{-Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta) Dpsil - Dtheta W^2 c32}{W^2}, \right. \right. \\ \left. \left. \frac{1}{2} \frac{2 R^2 Dpsil^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta) + 2 c32 Dpsil W^2}{W^2} \right], \right. \\ \left. \left[\frac{Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta) Dpsil + Dtheta W^2 c32}{W^2}, 0, 0 \right], \right. \\ \left. \left[\frac{-R^2 Dpsil^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta) - c32 Dpsil W^2}{W^2}, 0, 0 \right] \right]$$

$$c32 = - \frac{R^2 \sin(\theta) Dpsil (-I3 + mp hc^2 + I1) \cos(\theta)}{W^2}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Final form of C Matrix

> $C := eval\left(\% c32 = - \frac{R^2 \sin(\theta) Dpsil (-I3 + mp hc^2 + I1) \cos(\theta)}{W^2}\right) : simplify(\% size) : C := \%$

$$C := \left[\left[\frac{Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta)}{W^2}, \right. \right. \\ \left. \left. - \frac{R^3 \sin(\theta) hc mp (-Dpsir + Dpsil)}{W^2}, - \frac{1}{W^2} \left(2 \left(R (-I3 + mp hc^2 + I1) \left(Dpsil - \frac{1}{2} Dpsir \right) \cos(\theta) + \frac{1}{4} Dtheta mp hc W^2 \right) R \sin(\theta) \right) \right], \right. \\ \left[\frac{1}{W^2} \left((-2 Dtheta (-I3 + mp hc^2 + I1) \cos(\theta) + R hc mp (-Dpsir + Dpsil)) R^2 \sin(\theta) \right), \frac{Dtheta R^2 (-I3 + mp hc^2 + I1) \cos(\theta) \sin(\theta)}{W^2}, \right. \\ \left. \frac{\left(R Dpsil (-I3 + mp hc^2 + I1) \cos(\theta) - \frac{1}{2} Dtheta mp hc W^2 \right) \sin(\theta) R}{W^2} \right], \\ \left[\frac{R^2 \sin(\theta) (-I3 + mp hc^2 + I1) \cos(\theta) (-Dpsir + 2 Dpsil)}{W^2}, \right. \\ \left. - \frac{R^2 \sin(\theta) Dpsil (-I3 + mp hc^2 + I1) \cos(\theta)}{W^2}, 0 \right] \right]$$

Remaining Viscous Friction terms

> $C_v - C.Dq : \text{simplify}(\%)$;

$$\begin{bmatrix} bwm \, Dpsir \\ bwm \, Dpsil \\ -bwm \, (Dpsir + Dpsil) \end{bmatrix}$$

Viscous Friction Matrix

> $Cd := \text{map} \left(\text{eval}, \begin{bmatrix} \text{coeff}(\%1), Dpsir & \text{coeff}(\%1), Dpsil & 0 \\ \text{coeff}(\%2), Dpsir & \text{coeff}(\%2), Dpsil & 0 \\ \text{coeff}(\%3), Dpsir & \text{coeff}(\%3), Dpsil & 0 \end{bmatrix} \right)$;

$$Cd := \begin{bmatrix} bwm & 0 & 0 \\ 0 & bwm & 0 \\ -bwm & -bwm & 0 \end{bmatrix}$$

Linear Control

Build the state space matrix of $\dot{x} = [\text{psir } \text{psil } \text{theta } \text{Dpsir } \text{Dpsil } \text{Dtheta}]' = Ax + Bu$

>

$-M^{-1} \cdot (C_v + G)$:

$\text{map}(\text{mtaylor}, \%, [\text{theta}, \text{Dtheta}, \text{Dpsir} = 0, \text{Dpsil} = 0], 2)$:

THESE ARE OK, I ERASED THE PICTURE

$\text{eval} \left(\%, \left\{ I3 = \frac{89}{100000}, I2 = \frac{111}{100000}, I1 = \frac{153}{100000} \right\} \right)$:

$\text{eval} \left(\%, \left\{ W = \frac{16375}{100000}, mw = \frac{332}{10000}, R = 2 \cdot \frac{254}{10000}, mp = 1 + \frac{2398}{10000}, hc = \frac{746}{10000}, \right. \right.$
 $\left. \left. g = 9 + \frac{81}{100}, Jwm = \frac{157}{100000}, bwm = 0 \right\} \right)$:

$\text{map}(\text{eval}, [[0, 0, \text{coeff}(\%1), \text{theta}], \text{coeff}(\%1), \text{Dpsir}], \text{coeff}(\%1), \text{Dpsil}], \text{coeff}(\%1), \text{Dtheta}],$
 $[0, 0, \text{coeff}(\%2), \text{theta}], \text{coeff}(\%2), \text{Dpsir}], \text{coeff}(\%2), \text{Dpsil}], \text{coeff}(\%2),$
 $\text{Dtheta}],$
 $[0, 0, \text{coeff}(\%3), \text{theta}], \text{coeff}(\%3), \text{Dpsir}], \text{coeff}(\%3), \text{Dpsil}], \text{coeff}(\%3),$
 $\text{Dtheta}]]$:

$\text{evalf}(\%)$:

$A_{cg} := \%$

$$A_{cg} := \begin{bmatrix} 0. & 0. & -141.7482519 & 0. & 0. & 0. \\ 0. & 0. & -141.7482519 & 0. & 0. & 0. \\ 0. & 0. & 196.4264769 & 0. & 0. & 0. \end{bmatrix}$$

State Matrix

```
> <<< [ [ 0 0 0 1 0 0 ]
         [ 0 0 0 0 1 0 ]
         [ 0 0 0 0 0 1 ] ] >>>, (Acg) : simplify(% , size) : A := % : evalf(%)
```

$$\begin{bmatrix} 0. & 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 0. & 0. & 1. \\ 0. & 0. & -141.7482519 & 0. & 0. & 0. \\ 0. & 0. & -141.7482519 & 0. & 0. & 0. \\ 0. & 0. & 196.4264769 & 0. & 0. & 0. \end{bmatrix}$$

Input Matrix

```
> M^-1. [ [ 1 0 ]
          [ 0 1 ]
          [-1 -1 ] ] :
<<< [ [ 0 0 ]
       [ 0 0 ]
       [ 0 0 ] ] , simplify(% , size) :
eval(% , { I3 = 89/100000 , I2 = 111/100000 , I1 = 153/100000 } ) :
eval(% , { W = 16375/100000 , mw = 332/10000 , R = 2 * 254/10000 , mp = 1 + 2398/10000 , hc = 746/10000 ,
          g = 9 + 81/100 , Jwm = 157/100000 , bwm = 0 } ) :
map(mtaylor, % , { theta, Dtheta }, 2) :
evalf(% ) :
B := % : map(simplify, % , size)
```

$$\begin{bmatrix} 0. & 0. \\ 0. & 0. \\ 0. & 0. \\ 696.2319049 & 148.8829572 \\ 148.8829572 & 696.2319049 \\ -372.7191298 & -372.7191298 \end{bmatrix}$$

Non-Linear Control

Reference and Sliding Mode Vectors

$$\begin{aligned}
 > \quad DDq_r := \begin{bmatrix} DDpsir_d - \lambda_1 \cdot (Dpsir - Dpsir_d) \\ DDpsil_d - \lambda_2 \cdot (Dpsil - Dpsil_d) \\ DDtheta_d - \lambda_3 \cdot (Dtheta - Dtheta_d) \end{bmatrix} : \\
 Dq_r := \begin{bmatrix} Dpsir_d - \lambda_1 \cdot (psir - psir_d) \\ Dpsil_d - \lambda_2 \cdot (psil - psil_d) \\ Dtheta_d - \lambda_3 \cdot (theta - theta_d) \end{bmatrix} :
 \end{aligned}$$

$$s := Dq - Dq_r :$$

$$> \quad Km := \begin{bmatrix} Km11 & Km12 & Km13 \\ Km21 & Km22 & Km23 \\ Km31 & Km32 & Km33 \end{bmatrix} :$$

> *## Underactuated constraint for DDtheta_d*

(M.DDq_r + C.Dq_r + G) - Km.s :
%(1) + %(2) + %(3) :
#eval(%s, {psir_d=psir, psil_d=psil}) :
DDthetadsol := isolate(%s, DDtheta_d) :

eval(%s, {mp = mp, hc = hc, Dtheta = theta, Dtheta_d = theta_dot, DDtheta_d = theta_double_dot, Dpsir = psi_dot_r, Dpsir_d = psi_dot_rd, Dpsil = psi_dot_r, Dpsil_d = psi_dot_rd, theta_d = theta, DDpsil_d = psi_double_dot_ld, DDpsir_d = psi_double_dot_rd, mw = mw, Jwm = J}) :
simplify(%s, size) :

Find DDtheta_d

> *M.DDq_r + C.Dq_r + G - Km.s :*
isolate(%(1) + %(2) + %(3), DDtheta_d) :
#eval(%s, {psir_d=psir, psil_d=psil}) :
eval(%s, {I3 = 89/100000, I2 = 111/100000, I1 = 153/100000}) :
*eval(%s, {W = 16375/100000, mw = 332/10000, R = 2 * 254/10000, mp = 1 + 2398/10000, hc = 746/10000, g = 9 + 81/100, Jwm = 157/100000}) :*

$$\begin{aligned}
DDtheta_d = & \frac{1}{\frac{293652829}{62500000000} \cos(\theta) + \frac{1001210671}{125000000000}} \left(\frac{2268294687}{25000000000} \sin(\theta) \right. \\
& + Km11 (Dpsir - Dpsir_d + \lambda l (psir - psir_d)) + Km12 (Dpsil - Dpsil_d \\
& + \lambda 2 (psil - psil_d)) + Km13 (Dtheta - Dtheta_d + \lambda 3 (\theta - theta_d)) \\
& + Km21 (Dpsir - Dpsir_d + \lambda l (psir - psir_d)) + Km22 (Dpsil - Dpsil_d \\
& + \lambda 2 (psil - psil_d)) + Km23 (Dtheta - Dtheta_d + \lambda 3 (\theta - theta_d)) \\
& + Km31 (Dpsir - Dpsir_d + \lambda l (psir - psir_d)) + Km32 (Dpsil - Dpsil_d \\
& + \lambda 2 (psil - psil_d)) + Km33 (Dtheta - Dtheta_d + \lambda 3 (\theta - theta_d)) \\
& + \frac{1001210671}{125000000000} \lambda 3 Dtheta - \frac{1001210671}{125000000000} \lambda 3 Dtheta_d \\
& + \frac{293652829}{62500000000} \cos(\theta) \lambda 3 Dtheta - \frac{293652829}{62500000000} \cos(\theta) \lambda 3 Dtheta_d \\
& + \frac{4736326478941}{10474243164062500} \sin(\theta) (-Dpsir + Dpsil) (Dpsil_d - \lambda 2 (psil - psil_d)) \\
& + \frac{65024}{17161} \left(\frac{119692505217}{312500000000000} \left(Dpsil - \frac{1}{2} Dpsir \right) \cos(\theta) \right. \\
& + \left. \frac{39680127547}{64000000000000} Dtheta \right) \sin(\theta) (Dtheta_d - \lambda 3 (\theta - theta_d)) \\
& - \frac{1032256}{10725625} \left(-\frac{942460671}{62500000000} Dtheta \cos(\theta) - \frac{293652829}{62500000000} Dpsir \right. \\
& + \left. \frac{293652829}{62500000000} Dpsil \right) \sin(\theta) (Dpsir_d - \lambda l (psir - psir_d)) \\
& - \frac{32512}{17161} \left(\frac{119692505217}{312500000000000} Dpsil \cos(\theta) \right. \\
& - \left. \frac{39680127547}{32000000000000} Dtheta \right) \sin(\theta) (Dtheta_d - \lambda 3 (\theta - theta_d)) \\
& - \frac{293652829}{125000000000} \cos(\theta) (DDpsir_d - \lambda l (Dpsir - Dpsir_d)) \\
& - \frac{293652829}{125000000000} \cos(\theta) (DDpsil_d - \lambda 2 (Dpsil - Dpsil_d)) - \frac{160000}{17161} \left(\right. \\
& - \left. \frac{15200948162559}{19531250000000000} \cos(\theta)^2 + \frac{4379858184058151}{125000000000000000} \right) (DDpsir_d \\
& - \lambda l (Dpsir - Dpsir_d)) - \frac{160000}{17161} \left(-\frac{15200948162559}{19531250000000000} \cos(\theta)^2 \right. \\
& + \left. \frac{4379858184058151}{125000000000000000} \right) (DDpsil_d - \lambda 2 (Dpsil - Dpsil_d)) \\
& - \frac{258064}{10725625} \left(\frac{942460671}{31250000000} \cos(\theta)^2 - \frac{949333569}{200000000000} \right) (DDpsil_d \\
& - \lambda 2 (Dpsil - Dpsil_d)) - \frac{258064}{10725625} \left(\frac{942460671}{31250000000} \cos(\theta)^2 \right. \\
& - \left. \frac{949333569}{200000000000} \right) (DDpsir_d - \lambda l (Dpsir - Dpsir_d)) \\
& - \frac{15200948162559}{20948486328125000} Dtheta \cos(\theta) \sin(\theta) (Dpsir_d - \lambda l (psir - psir_d)) \\
& - \frac{15200948162559}{20948486328125000} Dtheta \cos(\theta) \sin(\theta) (Dpsil_d - \lambda 2 (psil - psil_d)) \\
& - \frac{15200948162559}{20948486328125000} \sin(\theta) \cos(\theta) (-Dpsir + 2 Dpsil) (Dpsir_d - \lambda l (psir \\
& - psir_d)) + \frac{15200948162559}{20948486328125000} \sin(\theta) Dpsil \cos(\theta) (Dpsil_d - \lambda 2 (psil \\
& - psil_d)) \left. \right)
\end{aligned}$$

tau input to LabView

>

($M.DDq_r + C.Dq_r + G) - Km.s :$

$eval\left(\%, \left\{ I3 = \frac{89}{100000}, I2 = \frac{111}{100000}, I1 = \frac{153}{100000} \right\}\right) :$

$eval\left(\%, \left\{ W = \frac{16375}{100000}, mw = \frac{332}{10000}, R = 2 \cdot \frac{254}{10000}, mp = 1 + \frac{2398}{10000}, hc = \frac{746}{10000}, \right. \right.$
 $\left. \left. g = 9 + \frac{81}{100}, Jwm = \frac{157}{100000}, bwm = 0 \right\}\right) :$

$eval(\%, \{psir_d = psir, psil_d = psil, theta_d = 0, Dtheta_d = 0\}) :$
 $\%(2);$

$$\begin{aligned} & \frac{258064}{10725625} \left(\frac{942460671}{3125000000} \cos(\theta)^2 - \frac{949333569}{2000000000000} \right) (DDpsir_d - \lambda 1 (Dpsir \\ & - Dpsir_d)) + \frac{160000}{17161} \left(-\frac{15200948162559}{1953125000000000} \cos(\theta)^2 \right. \\ & \left. + \frac{4379858184058151}{125000000000000000} \right) (DDpsil_d - \lambda 2 (Dpsil - Dpsil_d)) \\ & + \frac{293652829}{125000000000} \cos(\theta) (DDtheta_d - \lambda 3 Dtheta) + \frac{1032256}{10725625} \left(\right. \\ & \left. - \frac{942460671}{62500000000} Dtheta \cos(\theta) - \frac{293652829}{62500000000} Dpsir + \frac{293652829}{62500000000} Dpsil \right) \\ & \sin(\theta) Dpsir_d + \frac{15200948162559}{20948486328125000} Dtheta \cos(\theta) \sin(\theta) Dpsil_d \\ & - \frac{32512}{17161} \left(\frac{119692505217}{312500000000000} Dpsil \cos(\theta) \right. \\ & \left. - \frac{39680127547}{32000000000000} Dtheta \right) \sin(\theta) \lambda 3 \theta - Km21 (Dpsir - Dpsir_d) \\ & - Km22 (Dpsil - Dpsil_d) - Km23 (Dtheta + \lambda 3 \theta) \end{aligned}$$

Non-Holonomic Tracking Sliding Mode Control Values

>

($M \cdot DDq_r + C \cdot Dq_r + G$) - $Km \cdot s$:

$eval(\%, DDtheta_{d_{sol}})$:

$eval(\%, \{psir_d=0, psil_d=0, DDpsir_d=0, DDpsil_d=0, theta_d=0, Dtheta_d=0\})$:

$eval(\%, \left\{ I3 = \frac{89}{100000}, I2 = \frac{111}{100000}, I1 = \frac{153}{100000} \right\})$:

$eval(\%, \left\{ W = \frac{16375}{100000}, mw = \frac{332}{10000}, R = 2 \cdot \frac{254}{10000}, mp = 1 + \frac{2398}{10000}, hc = \frac{746}{10000}, \right.$
 $\left. g = 9 + \frac{81}{100}, Jwm = \frac{157}{100000}, bwm = 0 \right\})$:

$map(mtaylor, \%, [theta, Dtheta, Dpsir, Dpsil], 2)$:

$Flin := \%$:

$solve\left(\left\{ \begin{aligned} &coeff(Flin(1), theta) = 10 + \frac{27}{100}, coeff(Flin(1), Dpsir) = \frac{26}{1000}, \\ &coeff(Flin(1), Dpsil) = \frac{14}{100}, coeff(Flin(1), Dtheta) = \frac{19}{100}, coeff(Flin(2), theta) \\ &= 10 + \frac{27}{100}, coeff(Flin(2), Dpsir) = \frac{14}{100}, coeff(Flin(2), Dpsil) = \frac{26}{1000}, \\ &coeff(Flin(2), Dtheta) = \frac{19}{100} \end{aligned} \right\}\right)$:

$Flinsol := \%$

$$\begin{aligned} Flinsol := & \left\{ Km11 = \frac{38249697408529166609}{520249765518549250} Km21 + \frac{3589190400874451176083}{260124882759274625000} \right. \\ & - \frac{87133557464106883}{4096454846602750} Km31, Km12 = Km12, Km13 = Km23, Km21 = Km21, Km2: \\ & = \frac{3589190400874451176083}{260124882759274625000} + \frac{38249697408529166609}{520249765518549250} Km12 \\ & - \frac{87133557464106883}{4096454846602750} Km32, Km23 = Km23, Km31 = Km31, Km32 = Km32, Km3: \\ & = \frac{1001210671}{293652829} Km23 + \frac{30181810251}{29365282900}, \lambda1 = - \frac{1073861020314437500000}{27052987806964561} Km21 \\ & - \frac{202624070903137125000}{27052987806964561} + \frac{2480007971687500000}{213015652023343} Km31, \lambda2 = \\ & - \frac{202624070903137125000}{27052987806964561} - \frac{1073861020314437500000}{27052987806964561} Km12 \\ & \left. + \frac{2480007971687500000}{213015652023343} Km32, \lambda3 = \frac{13373021865077260159}{251515085425000000} \right\} \end{aligned}$$

>

$lambda3 : eval(\%, Flinsol) : evalf(\%);$

53.16985994

>

```
(M.DDq_r + C.Dq_r + G) - Km.s :
eval(% DDtheta_dsol) :
#eval(% {psir_d=0, psil_d=0, DDpsir_d=0, DDpsil_d=0, theta_d=0, Dtheta_d
=0}) :
eval(% {I3 = 89/100000, I2 = 111/100000, I1 = 153/100000}) :
eval(% {W = 16375/100000, mw = 332/10000, R = 2 * 254/10000, mp = 1 + 2398/10000, hc = 746/10000,
g = 9 + 81/100, Jwm = 157/100000, bwm = 0}) :
```

```
### solves for lamda1 and lambda2 and builds Kd matrix: copy and paste these
lines for Kd matrix
eval(eval(Km, Flinsol), solve( {eval(lambda1, Flinsol) = 1, eval(lambda2, Flinsol)
= 1})) :
eval(% solve( {%(1, 1) = 1, %(1, 3) = 1, %(3, 1) = %(3, 2)})) :
####
```

```
### eval contrl with the Km values
eval(%% {Km11 = %(1, 1), Km12 = %(1, 2), Km13 = %(1, 3), Km21 = %(2, 1),
Km22 = %(2, 2), Km23 = %(2, 3), Km31 = %(3, 1), Km32 = %(3, 2), Km33 = %(3,
3)}):
eval(% {lambda1 = 1, lambda2 = 1}) : eval(% Flinsol) :
##
```

```
map(mtaylor, %, {theta, Dtheta, Dpsir, Dpsil}, 2) :
```

```
map(eval, [coeff(%(1), psir), coeff(%(1), psil), coeff(%(1), theta), coeff(%(1), Dps
), coeff(%(1), Dpsil), coeff(%(1), Dtheta)]) :
evalf(%)
```

[0.02750512959, 0.1396781415, 10.27000000, 0.02600000000, 0.1400000000,
0.1900000000]

> lambda1 = 1; lambda2 = 1;

$$\lambda_1 = 1$$

$$\lambda_2 = 1$$

> lambda3 : eval(% Flinsol) : lambda3 = evalf(%)

$$\lambda_3 = 53.16985994$$

>

```
eval(eval(Km, Flinsol), solve( {eval(lambda1, Flinsol) = 5, eval(lambda2, Flinsol)
= 5})) :
eval(% solve( {%(1, 1) = 1, %(1, 3) = 1, %(3, 1) = %(3, 2)})) :
Kd = evalf(%)
```

$$K_d = \begin{bmatrix} 1. & 0.8951349404 & 1. \\ 0.8951349404 & 1. & 1. \\ 3.695726805 & 3.695726805 & 4.437310473 \end{bmatrix}$$

Desired Theta Tracking Sliding Mode Control Values

Solve for the Non-Linear Control Law Values

>

($MDDq_r + C.Dq_r + G$) - $Km.s$:

$eval(\%, DDthetaadsol)$:

$eval(\%, \{psir_d=0, psil_d=0, DDpsir_d=0, DDpsil_d=0, theta_d=0, Dtheta_d=0\})$:

$eval(\%, \{I3 = \frac{89}{100000}, I2 = \frac{111}{100000}, I1 = \frac{153}{100000}\})$:

$eval(\%, \{W = \frac{16375}{100000}, mw = \frac{332}{10000}, R = 2 \cdot \frac{254}{10000}, mp = 1 + \frac{2398}{10000}, hc = \frac{746}{10000}, g = 9 + \frac{81}{100}, Jwm = \frac{157}{100000}, bwm = 0\})$:

$map(maylor, \%, [theta, Dtheta, Dpsir, Dpsil], 2)$:

$Flin := \%$:

$solve(\{coeff(Flin(1), theta) = 15, coeff(Flin(1), Dpsir) = \frac{26}{1000}, coeff(Flin(1), Dpsil) = \frac{14}{100}, coeff(Flin(1), Dtheta) = \frac{20}{100}, coeff(Flin(2), theta) = 15, coeff(Flin(2), Dpsir) = \frac{14}{100}, coeff(Flin(2), Dpsil) = \frac{26}{1000}, coeff(Flin(2), Dtheta) = \frac{20}{100}\})$:

$Flinsol := \%$

$$\begin{aligned}
 Flinsol := & \left\{ Km11 = \frac{38249697408529166609}{520249765518549250} Km21 + \frac{3589190400874451176083}{260124882759274625000} \right. \\
 & - \frac{87133557464106883}{4096454846602750} Km31, Km12 = Km12, Km13 = Km23, Km21 = Km21, Km22 = \\
 & = \frac{3589190400874451176083}{260124882759274625000} + \frac{38249697408529166609}{520249765518549250} Km12 \\
 & - \frac{87133557464106883}{4096454846602750} Km32, Km23 = Km23, Km31 = Km31, Km32 = Km32, Km33 = \\
 & = \frac{1001210671}{293652829} Km23 + \frac{1588516329}{1468264145}, \lambda1 = -\frac{1073861020314437500000}{27052987806964561} Km21 \\
 & - \frac{202624070903137125000}{27052987806964561} + \frac{2480007971687500000}{213015652023343} Km31, \lambda2 = \\
 & - \frac{202624070903137125000}{27052987806964561} - \frac{1073861020314437500000}{27052987806964561} Km12 \\
 & \left. + \frac{2480007971687500000}{213015652023343} Km32, \lambda3 = \frac{19634423728552260159}{264752721500000000} \right\}
 \end{aligned}$$

>

$$\text{solve}\left(\left\{\begin{array}{l} \text{coeff}(\text{Flin}(1), \text{theta}) = 15, \text{coeff}(\text{Flin}(1), \text{Dpsir}) = \frac{26}{1000}, \text{coeff}(\text{Flin}(1), \\ \text{Dpsil}) = \frac{14}{100}, \text{coeff}(\text{Flin}(1), \text{Dtheta}) = \frac{40}{100}, \text{coeff}(\text{Flin}(2), \text{theta}) = 15, \\ \text{coeff}(\text{Flin}(2), \text{Dpsir}) = \frac{14}{100}, \text{coeff}(\text{Flin}(2), \text{Dpsil}) = \frac{26}{1000}, \text{coeff}(\text{Flin}(2), \\ \text{Dtheta}) = \frac{40}{100} \end{array}\right\}\right):$$

Flinsol2 := %:

>

$$\text{solve}\left(\left\{\begin{array}{l} \text{coeff}(\text{Flin}(1), \text{theta}) = 15, \text{coeff}(\text{Flin}(1), \text{Dpsir}) = \frac{26}{1000}, \text{coeff}(\text{Flin}(1), \\ \text{Dpsil}) = \frac{14}{100}, \text{coeff}(\text{Flin}(1), \text{Dtheta}) = \frac{60}{100}, \text{coeff}(\text{Flin}(2), \text{theta}) = 15, \\ \text{coeff}(\text{Flin}(2), \text{Dpsir}) = \frac{14}{100}, \text{coeff}(\text{Flin}(2), \text{Dpsil}) = \frac{26}{1000}, \text{coeff}(\text{Flin}(2), \\ \text{Dtheta}) = \frac{60}{100} \end{array}\right\}\right):$$

Flinsol3 := %:

Desired path $\theta_d = a \sin\left(\frac{2\pi}{T}t\right)$

Sliding Mode Control Gains:

Path 1: a = 0.8 , T = 0.2

>

```
eval(eval(Km, Flinsol), solve( {eval(lambda1, Flinsol) = 1, eval(lambda2, Flinsol)
= 1})):
eval(% solve( {%(1, 1) = 1, %(1, 3) = 1, %(3, 1) = %(3, 2)})):
```

```
Kd = evalf(%);
lambda3 = evalf(eval(lambda3, Flinsol));
```

$$Kd = \begin{bmatrix} 1. & 0.8878269881 & 1. \\ 0.8878269881 & 1. & 1. \\ 3.670466736 & 3.670466736 & 4.491405519 \end{bmatrix}$$

$$\lambda_3 = 74.16136694$$

Path 2: $a = 0.25$, $T = 0.6$

>

```
eval(eval(Km, Flinsol2), solve( {eval(lambda1, Flinsol2) = 1, eval(lambda2, Flinsol2) = 1} )) :  
eval(% solve( {%(1, 1) = 1, %(1, 3) = 1, %(3, 1) = %(3, 2)} )) :
```

```
Kd = evalf(%);  
lambda3 = evalf(eval(lambda3, Flinsol2));
```

$$Kd = \begin{bmatrix} 1. & 0.8878269881 & 1. \\ 0.8878269881 & 1. & 1. \\ 3.670466736 & 3.670466736 & 5.573306439 \end{bmatrix}$$

$$\lambda_3 = 37.08068347$$

Path 2: $a = 0.325$, $T = 0.625$

>

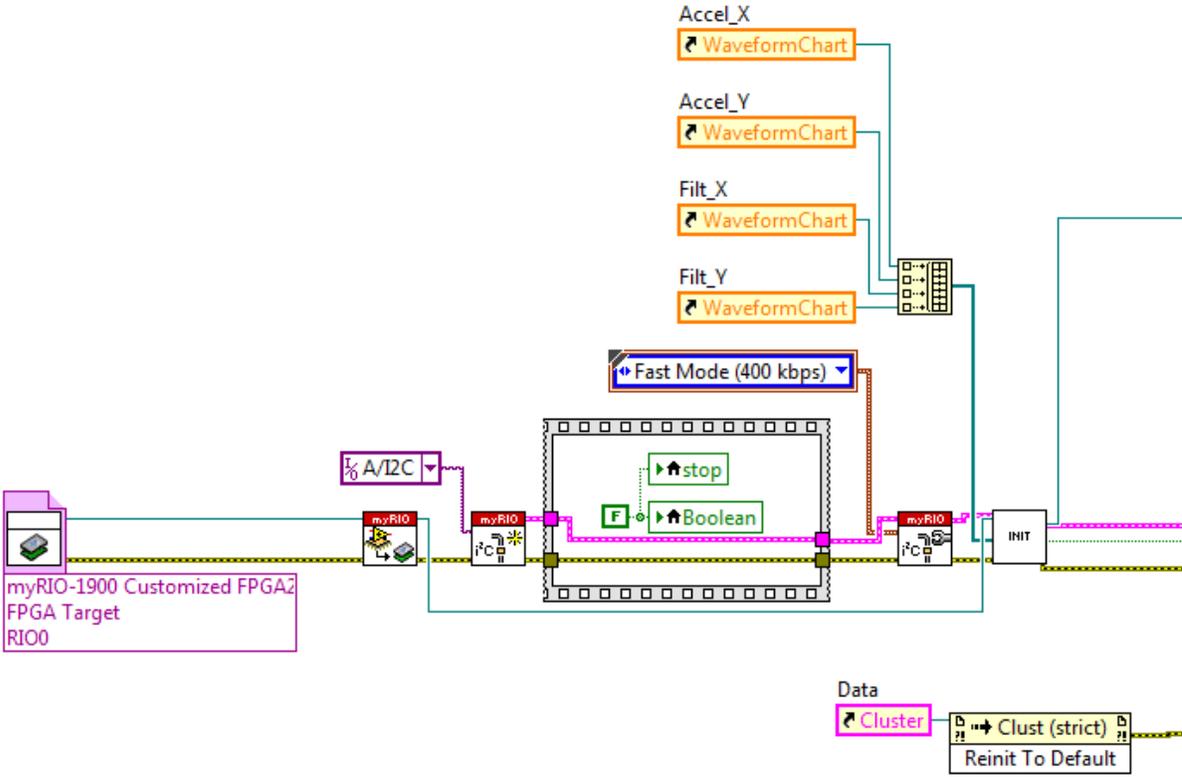
```
eval(eval(Km, Flinsol3), solve( {eval(lambda1, Flinsol3) = 1, eval(lambda2, Flinsol3) = 1} )) :  
eval(% solve( {%(1, 1) = 1, %(1, 3) = 1, %(3, 1) = %(3, 2)} )) :
```

```
Kd = evalf(%);  
lambda3 = evalf(eval(lambda3, Flinsol3));
```

$$Kd = \begin{bmatrix} 1. & 0.8878269881 & 1. \\ 0.8878269881 & 1. & 1. \\ 3.670466736 & 3.670466736 & 6.655207358 \end{bmatrix}$$

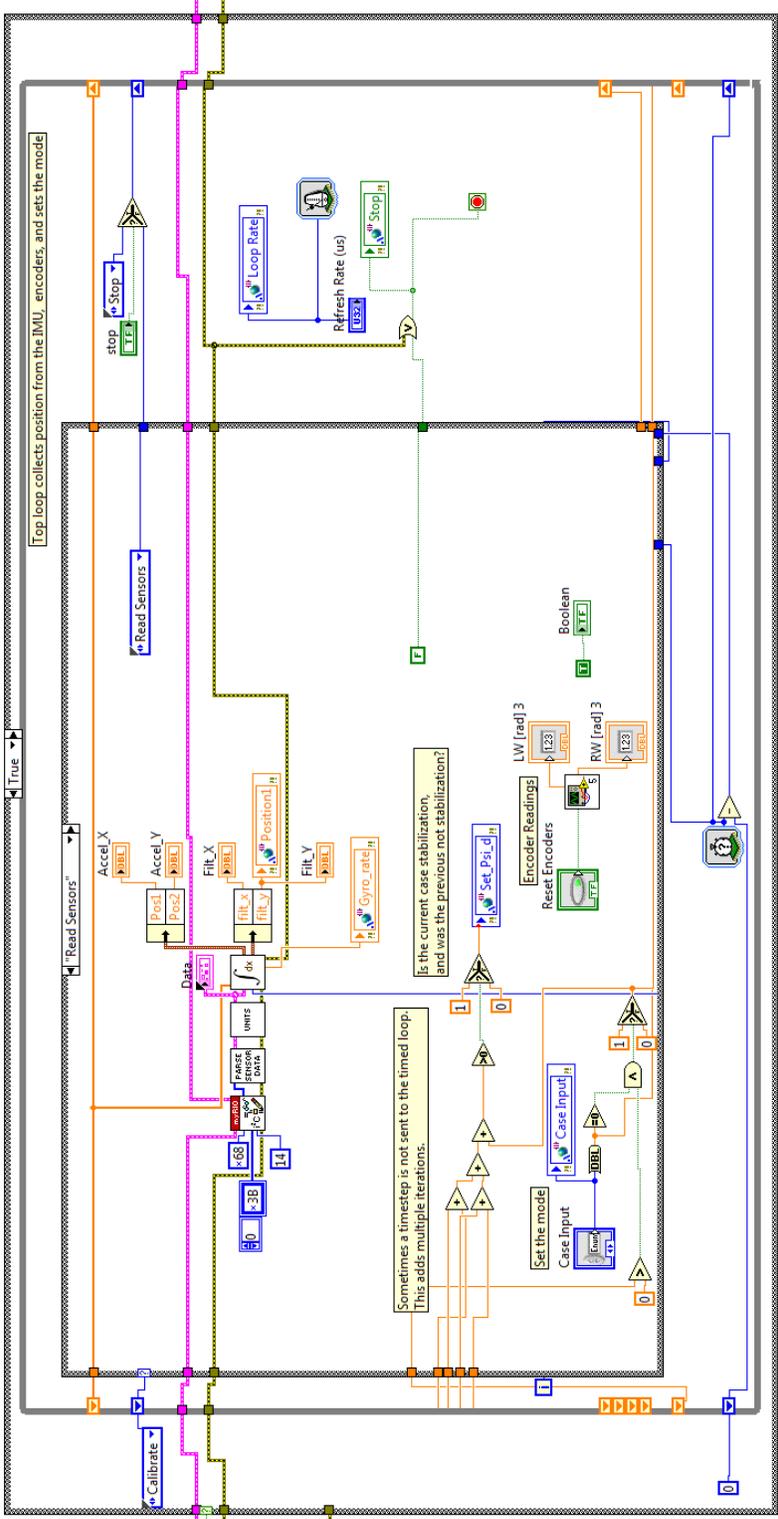
$$\lambda_3 = 24.72045565$$

Labview Initialization



This shows the portion of the VI that initializes the I2C protocol used with the MPU-6050 accelerometer/gyro.

Labview State Acquisition



This shows the portion of the VI used for the state acquisition.

The timed loop uses the encoder measurements to estimate the wheel velocities through a linear discrete observer. The bulk of the timed loop is implemented in a math script node. The Segbot has three modes that it can be put in: Stationary, Velocity Mode, and Tracking. Stationary Mode keeps the Segbot at its current XY coordinates. The velocity mode operates like a remote control car where the Segbot can be steered forwards, backwards, left, and right from the front panel. When in tracking mode the math script node calls two m-file functions. These m-files define the desired trajectory to follow and the navigational control. The sliding mode control derived in this thesis then calculates a desired torque. The desired torque is then converted to a duty cycle to be applied to the motor driver. The math script node and m-file functions are given below.

Timed Loop Math Script Node

```

%% NEED TO MAKE SURE FEEDBACK ARRAY ELEMENTS ARE ALL INITIALIZED

%% Switches Desired wheel angles to current angle
%% When Entering "Stationary" Control From a Different One
psir_reset = fb(1);
psil_reset = fb(2);
if Set_Psi_d > 0 | tstart<2
psir_reset = states(1);
psil_reset = states(2);
fb(3) =x0;
fb(4) =0;
end

%% Dynamic Properties
R = 2*2.54/100;      %% Wheel Radius

% Defines Current States
psir = states(1)-psir_reset;
psil = states(2)-psil_reset;
%% Angle Offset - Already Added in top loop %%
theta = states(3)+thetaoff(1);
%%%%%%%%%%%%%%
Dpsir = states(4);
Dpsil = states(5);
Dtheta = states(6);

% Holonomic Constraints
phi = (psir - psil)*R/W+pi/2;
Dphi = (Dpsir - Dpsil)*R/W;
Vel = (Dpsir + Dpsil)*R/2;
%% Set the Control Mode
Mode = DesiredStates(5);

%% Non-Holonomic Constraints
Dx = Vel*cos(phi);
Dy = Vel*sin(phi);

```

```

x_old = fb(3);
y_old = fb(4);

%%% Let everything settle when first starting VI %%%
if tstart<4
x =x0;y=0;
else
x = x_old + Dx*DT;
y = y_old + Dy*DT;
end

if (Mode==2) & (PathChoice==2)
phi = pi/2; x=0.25;
end

%% Defined variables
taur_e=0;taul_e=0;

%% Stationary Mode
if Mode == 0
phi_d = DesiredStates(1);
psir_d = W/(2*R)*phi_d;
psil_d = -W/(2*R)*phi_d;
theta_d = DesiredStates(2);
v_d = DesiredStates(3);
Dphi_d = DesiredStates(4);
Dpsir_d = 0;Dpsil_d = 0;Dtheta_d = 0;

%% Linear
%taur = (psir-psir_d)*taulincon(1) + (psil-psil_d)*taulincon(2) + (theta-
theta_d)*taulincon(3) + (Dpsir-Dpsir_d)*taulincon(4) + (Dpsil-
Dpsil_d)*taulincon(5) + (Dtheta-Dtheta_d)*taulincon(6);
%taul = (psir-psir_d)*taulincon(2) + (psil-psil_d)*taulincon(1) + (theta-
theta_d)*taulincon(3) + (Dpsir-Dpsir_d)*taulincon(5) + (Dpsil-
Dpsil_d)*taulincon(4) + (Dtheta-Dtheta_d)*taulincon(6);
taur = (psir-psir_d)*0+ (psil-psil_d)*.12+ (theta-theta_d)*10+ (Dpsir-
Dpsir_d)*.02 + (Dpsil-Dpsil_d)*.14+ (Dtheta-Dtheta_d)*0.1;
taul = (psir-psir_d)*0 + (psil-psil_d)*.12+ (theta-theta_d)*10 + (Dpsir-
Dpsir_d)*.14+ (Dpsil-Dpsil_d)*.02+ (Dtheta-Dtheta_d)*0.1;

end

%% Remote Control Mode %%%
if Mode == 1
psir_d = psir;psil_d = psil;
Dtheta_d = 0;theta_d =0;
v_d = DesiredStates(3);
Dphi_d = DesiredStates(4);
Dpsis = inv([R/2 R/2; R/W -R/W])* [v_d;Dphi_d];
Dpsir_d = Dpsis(1);
Dpsil_d = Dpsis(2);

%% Linear Control %%%

```

```

if Tracking_Choice==0
    taur = (psir-psir_d)*0 + (psil-psil_d)*0.12 + (theta-theta_d)*10 + (Dpsir-
Dpsir_d)*0.02 + (Dpsil-Dpsil_d)*0.14 + (Dtheta-Dtheta_d)*0.2;
    taul = (psir-psir_d)*0.12 + (psil-psil_d)*0 + (theta-theta_d)*10 + (Dpsir-
Dpsir_d)*0.14 + (Dpsil-Dpsil_d)*0.02 + (Dtheta-Dtheta_d)*0.2;
end

%% Non-Linear Control %%
if Tracking_Choice==1
    Km11 = Km(1); Km12 = Km(2); Km13 =Km(3);
    Km21 = Km(4); Km22 = Km(5); Km23 =Km(6);
    Km31 = Km(7); Km32 = Km(8); Km33 =Km(9);
    lambda1 = Km(10); lambda2 = Km(11); lambda3 =Km(12);

    ddthet anum1a= ((2 * Dpsil - Dtheta - Dpsir) * Dpsir_d + (lambda3 * theta -
Dpsil_d) * Dpsil - lambda3 * theta * Dpsir + Dtheta * Dpsil_d) * cos(theta);
    ddthet anum1b= (-0.587305658e9 / 0.942460671e9 * Dpsir + 0.587305658e9 /
0.942460671e9 * Dpsil) * Dpsir_d - 0.587305658e9 / 0.942460671e9 * Dpsil *
Dpsil_d + 0.24800079716875e14 / 0.3830160166944e13 * lambda3 * theta * Dtheta
- 0.202740651685453125e18 / 0.162143447067296e15 + 0.587305658e9 /
0.942460671e9 * Dpsil_d * Dpsir;
    ddthet anum1= -0.972860682403776e15 *
sin(theta)*(ddthet anum1a+ddthet anum1b);
    ddthet anum2 = (6299220248086250 * lambda3 * Dtheta + 3149610124043125 *
lambda1 * Dpsir - 3149610124043125 * lambda1 * Dpsir_d + 3149610124043125 *
lambda2 * Dpsil - 3149610124043125 * lambda2 * Dpsil_d) * cos(theta) + ((-
1340703125000000000 * Km11 - 1340703125000000000 * Km21 - 1340703125000000000
* Km31 - 4364546382923750 * lambda1) * Dpsir_d) + ((1340703125000000000 *
Km12 + 1340703125000000000 * Km22 + 4364546382923750 * lambda2 +
1340703125000000000 * Km32) * Dpsil);
    ddthet anum3= (-1340703125000000000 * Km12 - 1340703125000000000 * Km22 -
1340703125000000000 * Km32 - 4364546382923750 * lambda2) * Dpsil_d +
(1340703125000000000 * Km11 + 1340703125000000000 * Km31 +
1340703125000000000 * Km21 + 4364546382923750 * lambda1) * Dpsir +
(10738610203144375 * Dtheta + (1340703125000000000 * Km13 +
1340703125000000000 * Km23 + 1340703125000000000 * Km33) * theta) * lambda3 +
1340703125000000000 * Dtheta * (Km13 + Km23 + Km33);
    ddthet aden = 0.6299220248086250e16 * cos(theta) + 0.10738610203144375e17;
    ddthet anum = ddthet anum1 + ddthet anum2 + ddthet anum3;
    DDtheta_d = ddthet anum/ddthet aden;

    taur = -0.9323465998e1 * (-0.7782885459e-4 * cos(theta) ^ 2 +
0.3503886547e-3) * lambda1 * (Dpsir - 0.1e1 * Dpsir_d) - 0.2406050929e-1 *
(0.3015874147e-1 * cos(theta) ^ 2 - 0.4746667845e-3) * lambda2 * (Dpsil -
0.1e1 * Dpsil_d) + 0.2349222632e-2 * cos(theta) * (DDtheta_d - 0.1e1 *
lambda3 * Dtheta) + 0.7256346795e-3 * Dtheta * cos(theta) * sin(theta) *
Dpsir_d - 0.4521879438e-3 * sin(theta) * (-0.1e1 * Dpsir + Dpsil) * Dpsil_d +
0.1451269359e-2 * sin(theta) * lambda3 * theta * ((0.1000000000e1 * Dpsil -
0.5000000000e0 * Dpsir) * cos(theta) + 0.1618736465e1 * Dtheta) - Km11 *
(Dpsir - 0.1e1 * Dpsir_d) - Km12 * (Dpsil - 0.1e1 * Dpsil_d) - Km13 * (Dtheta
+ lambda3 * theta);
    taul = (-0.7256346793e-3 * lambda1 * Dpsir + 0.7256346793e-3 * lambda1
* Dpsir_d + 0.7256346794e-3 * lambda2 * Dpsil - 0.7256346794e-3 * lambda2 *
Dpsil_d) * cos(theta) ^ 2 + ((-0.7256346795e-3 * Dpsil * lambda3 * theta -
0.1451269359e-2 * Dtheta * Dpsir_d + 0.7256346795e-3 * Dtheta * Dpsil_d) *
sin(theta) - 0.2349222632e-2 * lambda3 * Dtheta + 0.2349222632e-2 *

```

```

DDtheta_d) * cos(theta) + ((0.4521879438e-3 * Dpsil - 0.4521879438e-3 *
Dpsir) * Dpsir_d + 0.2349222632e-2 * lambda3 * theta * Dtheta) * sin(theta) +
(0.1e1 * Km21 - 0.1142072458e-4 * lambda1) * Dpsir_d - 0.1e1 * Km23 * Dtheta
+ (-0.1e1 * Km22 - 0.3266836708e-2 * lambda2) * Dpsil + (0.1e1 * Km22 +
0.3266836708e-2 * lambda2) * Dpsil_d - 0.1e1 * Km23 * lambda3 * theta +
0.1142072458e-4 * lambda1 * Dpsir - 0.1e1 * Km21 * Dpsir;

%DDpsir_d=0.0;DDpsil_d=0.0;
%tauinput = [Km psir psir_d psil psil_d theta theta_d Dpsir Dpsir_d Dpsil
Dpsil_d Dtheta Dtheta_d DDpsir_d DDpsil_d]';
%tauoutput = taucalc(tauinput);
%taur=tauoutput(1);
%taul=tauoutput(2);
end
end

%% Tracking Mode %%
if Mode ==2
%% Set the time offset
if Tracking_Reset ==1
t_reset = tstart;
else
t_reset = fb(5);
end
t = tstart-t_reset;

omega=0;vdes=1;x_d=0;y_d=0;
k1=0;k2=0;k3=0;k4=0;k5=0;
theta_d=0;Dtheta_d=0;
Dphi_d=0;phi_d=0;DDphi_d=0;
Dphi_d_Nav=0; DDphi_d_Nav=0;
Dv_d=0;v_d=0;DDv_d=0;
theta_d =0;Dtheta_d = 0; x_d=0;
v_d_Nav=0;Dv_d_Nav=0;
y_d =0;Dpsir_d=0;Dpsil_d=0;

if PathChoice==0
psir_d = psir;psil_d = psil;
%% Makes Input for Figure-8 Path
vdes = DesPath(1);omega = DesPath(2);
u = [t vdes omega]';
%% Set the Desired Trajectory to Track %%
DesiredTracking = DesiredCoordsLemis3(u);
x_d = DesiredTracking(1);
y_d = DesiredTracking(2);
phi_d = DesiredTracking(5);
Dphi_d = DesiredTracking(6);
v_d =DesiredTracking(7);
DDphi_d = DesiredTracking(8);
Dv_d = DesiredTracking(9);
end

if PathChoice == 1
psir_d = psir;psil_d = psil;

```

```

theta_d =0;Dtheta_d = 0;
vdes=1;omega=SquiggleIn(2);
t = tstart-t_reset;
u = [t SquiggleIn(1) SquiggleIn(2) SquiggleIn(3) SquiggleIn(4)]';

DesiredTracking = SquigglePath(u);
x_d = DesiredTracking(1);
y_d = DesiredTracking(2);
phi_d = DesiredTracking(3);
Dphi_d = DesiredTracking(4);
v_d = DesiredTracking(5);
DDphi_d=0;Dv_d=0;
end

if PathChoice ==2
a = sinpath( 2);
period = sinpath(1);
omega = 2*pi/period;

y_d= a*sin(2*pi/period*t);
v_d = a*(2*pi/period)*cos(2*pi/period*t);
Dv_d= -a*(2*pi/period)^2*sin(2*pi/period*t);

psir_d=y_d/R;psil_d=psir_d;

Dpsir_d=v_d/R;Dpsil_d=Dpsir_d;
DDpsir_d=Dv_d/R;DDpsil_d=DDpsir_d;
end

if PathChoice ==3
a = sinpath( 2);
period = sinpath(1);
omega = 2*pi/period;

theta_d = a*sin(2*pi/period*t);
Dtheta_d = a*(2*pi/period)*cos(2*pi/period*t);

psir_d =0;psil_d=0;
Dpsir_d=0; Dpsil_d=0;
DDpsir_d=0;DDpsil_d=0;

    if Tracking_Choice==1
        if Holo_nav ==1
            Dpsir_d = -NavCon(1)*y;
            Dpsil_d=Dpsir_d;
            psir_d=psir;psil_d=psil;
        end
    end
end

```

```

if PathChoice ==0 | PathChoice ==1
%% Navigational Control Gains %%
k1 = NavCon(1);k2 = NavCon(2); k3 = 2*sqrt(k2); k4 = NavCon(3); k5 =
NavCon(4);
NavIn = [x_d y_d phi_d Dphi_d v_d x y phi k1 k2 k3 k4 k5 DDphi_d Dv_d]';
%% Navigational Control m-file %%
NavOut = NavigationalControl_DDpsid(NavIn);
%% Navigational Control Output %%
v_d_Nav = NavOut(1);
Dphi_d_Nav = NavOut(2);
Dv_d_Nav = NavOut(3);
DDphi_d_Nav = NavOut(4);

mapper = [1/R W/(2*R);1/R -W/(2*R)];
Dpsis = mapper*[v_d_Nav Dphi_d_Nav]';
Dpsir_d = Dpsis(1);
Dpsil_d = Dpsis(2);
DDpsis = mapper*[Dv_d_Nav DDphi_d_Nav]';
DDpsir_d = DDpsis(1);
DDpsil_d = DDpsis(2);

if Tracking_Choice == 0
%% Navigational Control Gains %%
k1 = NavCon(1);k2 = NavCon(2); k3 = 2*sqrt(k2); k4 = NavCon(3);
k5 = NavCon(4);
NavIn = [x_d y_d phi_d Dphi_d v_d x y phi k1 k2 k3 k4 k5 DDphi_d
Dv_d]';

%% Navigational Control m-file %%
NavOut = NavigationalControl_DDpsidlin(NavIn);
%% Navigational Control Output %%
v_d_Nav = NavOut(1);
Dphi_d_Nav = NavOut(2);
Dv_d_Nav = NavOut(3);
DDphi_d_Nav = NavOut(4);

mapper = [1/R W/(2*R);1/R -W/(2*R)];
Dpsis = mapper*[v_d_Nav Dphi_d_Nav]';
Dpsir_d = Dpsis(1);
Dpsil_d = Dpsis(2);
DDpsis = mapper*[Dv_d_Nav DDphi_d_Nav]';
DDpsir_d = DDpsis(1);
DDpsil_d = DDpsis(2);

end
end

% linear
%if Tracking_Choice == 0
taur_L = (psir-psir_d)*taulincon(1) + (psil-psil_d)*taulincon(2) +
(theta-theta_d)*taulincon(3) + (Dpsir-Dpsir_d)*taulincon(4) + (Dpsil-
Dpsil_d)*taulincon(5) + (Dtheta-Dtheta_d)*taulincon(6);
taul_L = (psir-psir_d)*taulincon(2) + (psil-psil_d)*taulincon(1) +
(theta-theta_d)*taulincon(3) + (Dpsir-Dpsir_d)*taulincon(5) + (Dpsil-
Dpsil_d)*taulincon(4) + (Dtheta-Dtheta_d)*taulincon(6);
%end

```

```

    % Non-Linear
    %if Tracking_Choice == 1
    %% Calculate Torque
    tauinput = [Km psir psir_d psil psil_d theta theta_d Dpsir Dpsir_d Dpsil
Dpsil_d Dtheta Dtheta_d DDpsir_d DDpsil_d]';
    tauoutput = taucalc(tauinput);
    taur_NL=tauoutput(1);
    taul_NL=tauoutput(2);
    % end

if Tracking_Choice ==0
taur = taur_L;
taul = taul_L;
end

if Tracking_Choice == 1
taur = taur_NL;
taul = taul_NL;
end

end

%% make sure everything is defined if not in tracking mode
if Mode~=2
x_d =0;x=x0;y=0;
    if PathChoice == 1|PathChoice==2
        x = 0;
    end
taur1=0;DDtheta_d =0;taul1=0;y_d =0;ddthet anum1=0;
v_d=0;phi_d=0;Dphi_d=0;v_d_Nav =0;Dphi_d_Nav =0;k1
=0;k2=0;k3=0;k4=0;k5=0;DDtheta_d=0;
Dpsir_d=0;Dpsil_d=0;t_reset=0;vdes=0;omega=0;t = 0;DesiredTracking=0;omega=0;
end

%% Make sure motors are off when first starating VI
if tstart<4
    Dutyr =0;Vr=0;taur=0;
    Duty1 = 0;V1=0;taul=0;
else
    %% Calculate Voltage and Duty Cycle
    Vr = (Vcon(1))* taur + Vcon(2)*Dpsir;
    Dutyr = Vr/11.94;

    V1 = (Vcon(3))* taul + Vcon(4)*Dpsil;
    Duty1 = V1/11.94;
end

if abs(theta)>0.75
Duty1 =0;

```

```

Dutyx = 0;
end

%% Motor Duty Cycle Outputs
output = [Dutyx Dutyy]';

%% Feedback
fb = [psir_reset psil_reset x y t_reset]';

%% Set Loop so data can be recorded on specified loop
if Mode == 2
Loop = floor(t/(2*pi/omega));
    if Loop < 10
        if abs(theta) > 0.5
            light_off = 1;
        end
    end
else Loop = 99999;
end

% Reset for the Light
if Light_Reset == 1
light_off = 0;
end

%% Values to be saved in Spreadsheet and realtime XY plots
spreadsheet = [t k1 k2 k3 k4 k5 x_d y_d x y Tracking_Choice Loop Holo_nav
theta_d theta]';

StationaryReady = max(psir-psir_d,psil-psil_d);

```

Desired Trajectory M-file

```

function Y=DesiredCoordsLemis3(u)

a      = u(2);      %%% Figure Eight Width
omega  = u(3);      %%% Omega*pi= max(Dphi)
t      = u(1);

x = a * cos(omega * t) / (0.1e1 + sin(omega * t) ^ 2);
dx = -a * sin(omega * t) * omega / (0.1e1 + sin(omega * t) ^ 2) - 0.2e1 * a *
cos(omega * t) ^ 2 / (0.1e1 + sin(omega * t) ^ 2) ^ 2 * sin(omega * t) *
omega;
ddx = -a * cos(omega * t) * omega ^ 2 / (0.1e1 + sin(omega * t) ^ 2) + 0.6e1
* a * sin(omega * t) ^ 2 * omega ^ 2 / (0.1e1 + sin(omega * t) ^ 2) ^ 2 *
cos(omega * t) + 0.8e1 * a * cos(omega * t) ^ 3 / (0.1e1 + sin(omega * t) ^
2) ^ 3 * sin(omega * t) ^ 2 * omega ^ 2 - 0.2e1 * a * cos(omega * t) ^ 3 /
(0.1e1 + sin(omega * t) ^ 2) ^ 2 * omega ^ 2;

```

```

y = a * cos(omega * t) / (0.1e1 + sin(omega * t) ^ 2) * sin(omega * t);
dy = -a * sin(omega * t) ^ 2 * omega / (0.1e1 + sin(omega * t) ^ 2) - 0.2e1 *
a * cos(omega * t) ^ 2 / (0.1e1 + sin(omega * t) ^ 2) ^ 2 * sin(omega * t) ^
2 * omega + a * cos(omega * t) ^ 2 / (0.1e1 + sin(omega * t) ^ 2) * omega;
ddy = -0.4e1 * a * sin(omega * t) * omega ^ 2 / (0.1e1 + sin(omega * t) ^ 2)
* cos(omega * t) + 0.6e1 * a * sin(omega * t) ^ 3 * omega ^ 2 / (0.1e1 +
sin(omega * t) ^ 2) ^ 2 * cos(omega * t) + 0.8e1 * a * cos(omega * t) ^ 3 /
(0.1e1 + sin(omega * t) ^ 2) ^ 3 * sin(omega * t) ^ 3 * omega ^ 2 - 0.6e1 * a
* cos(omega * t) ^ 3 / (0.1e1 + sin(omega * t) ^ 2) ^ 2 * sin(omega * t) *
omega ^ 2;

phi=atan2(dy,dx);
if phi<-0.786
    phi = phi-sign(phi)*2*pi;
end
dphi=-(-ddy*dx+dy*ddx)/(dx^2+dy^2);

dddx = -a * sin(omega * t) * omega ^ 3 * (-0.1e1 + 0.3e1 * sin(omega * t) ^ 2
+ 0.9e1 * sin(omega * t) ^ 4 + 0.5e1 * sin(omega * t) ^ 6 - 0.20e2 *
cos(omega * t) ^ 2 + 0.8e1 * cos(omega * t) ^ 2 * sin(omega * t) ^ 2 + 0.28e2
* cos(omega * t) ^ 2 * sin(omega * t) ^ 4 + 0.24e2 * cos(omega * t) ^ 4 *
sin(omega * t) ^ 2 - 0.24e2 * cos(omega * t) ^ 4) / (0.1e1 + sin(omega * t) ^
2) ^ 4;

ddy = -0.6e1 * a * omega ^ 3 * (sin(omega * t) ^ 8 / 0.3e1 + 0.4e1 / 0.3e1 *
cos(omega * t) ^ 2 * sin(omega * t) ^ 6 + (-0.14e2 / 0.3e1 * cos(omega * t) ^
2 + cos(omega * t) ^ 4 - 0.1e1) * sin(omega * t) ^ 4 + (-0.2e1 / 0.3e1 -
0.16e2 / 0.3e1 * cos(omega * t) ^ 2 - 0.6e1 * cos(omega * t) ^ 4) * sin(omega
* t) ^ 2 + 0.2e1 / 0.3e1 * cos(omega * t) ^ 2 + cos(omega * t) ^ 4) / (0.1e1
+ sin(omega * t) ^ 2) ^ 4;

ddphi = (dx ^ 3 * dddy + (-2 * ddx * ddy - dy * dddx) * dx ^ 2 + dy * (-2 *
ddy ^ 2 + dy * dddy + 2 * ddx ^ 2) * dx + 2 * ddy * ddx * dy ^ 2 - dy ^ 3 *
dddx) / (dx ^ 2 + dy ^ 2) ^ 2;

Dvd = (x * dx + y * dy) * (x ^ 2 + y ^ 2) ^ (-0.1e1 / 0.2e1);

v = sqrt(dx^2+dy^2);
Y = [x y dx dy phi dphi v dphi Dvd]';

```

Navigational Control M-file

```

function qout = NavigationalControl_DDpsidlin(u)
%u = randn(15);

%% Imports %%
xd    = u(1);
yd    = u(2);
phid  = u(3);
Dphid = u(4);
vd    = u(5);
x     = u(6);

```

```

y      = u(7);
phi    = u(8);

k1     = u(9);
k2     = u(10);
k3     = u(11);
k4     = u(12);
k5     = u(13);

DDphid = u(14);
Dvd    = u(15);

%% Global Differences
e = [xd-x yd-y phid-phi]';
% Rotation to body
R = [cos(phi) sin(phi) 0; -sin(phi) cos(phi) 0; 0 0 1];
%% Local Error: [tangent lateral phi]'
pe = R*e;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
e1 = pe(1);
e2 = pe(2);
e3 = pe(3);

%% Velocity and Angular Velocity Control:
v = (vd+k1*e1) + k4*(e1)*e2^2;
Dphi = Dphid + vd*(k2*e2 + k3*(e3)) + k5*sin(e3)*e2^2;

%Dv = cos(e3) * Dvd - vd * sin(e3) * (Dphid - Dphi) + (k1 + e2 ^ 2 * k4) *
(Dphi * e2 - k1 * e1 - k4 * e1 * e2 ^ 2) + 0.2e1 * k4 * e1 * e2 * (-e1 * Dphi
+ vd * sin(e3));
Dv = 0;
DDphi=0;

%DDphi = (k2 * e2 + k3 * sin(e3)) * Dvd + (vd * k3 * cos(e3) + cos(e3) * k5 *
e2 ^ 2) * (Dphid - Dphi) + (vd * k2 + 0.2e1 * sin(e3) * k5 * e2) * (-e1 *
Dphi + vd * sin(e3)) + DDphid;

qout = [v Dphi Dv DDphi]';

```

Wheel Velocity Estimate

```

double wn;
wn = 2*pi*CutOffFrequency;

b0 = (dt*dt*wn*wn + 4*zeta*dt*wn + 4);
b1 = (2*dt*dt*wn*wn - 8)/b0;
b2 = (dt*dt*wn*wn - 4*zeta*dt*wn + 4)/b0;
a0 = 2*dt*wn*wn/b0;
a1 = 0/b0;
a2 = (-2*dt*wn*wn)/b0;

Speed = a0*pos+a1*Pk_1 + a2*Pk_2 - b1*Vk_1 - b2*Vk_2;

```

Torque Calculation

```
function output = tauca(u);

Km11=u(1);Km12=u(2);Km13=u(3);
Km21=u(4);Km22=u(5);Km23=u(6);
Km31=u(7);Km32=u(8);Km33=u(9);
lambda1=u(10);lambda2=u(11);lambda3=u(12);

psir=u(13);psir_d=u(14);
psil=u(15);psil_d=u(16);
theta=u(17);theta_d=u(18);
Dpsir=u(19);Dpsir_d=u(20);
Dpsil=u(21);Dpsil_d=u(22);
Dtheta=u(23);Dtheta_d=u(24);
DDpsir_d=u(25);DDpsil_d=u(26);

num1 = (-0.293652829e9 / 0.125000000000e12 * cos(theta) - 0.203463499e9 /
0.62500000000e11) * (DDpsir_d - lambda1 * (Dpsir - Dpsir_d) + DDpsil_d - lambda2 *
(Dpsil - Dpsil_d));

num2 = 0.1001210671e10 / 0.125000000000e12 * lambda3 * Dtheta + 0.2268294687e10 /
0.2500000000e10 * sin(theta) + 0.293652829e9 / 0.6250000000e11 * cos(theta) * lambda3
* Dtheta - 0.293652829e9 / 0.6250000000e11 * cos(theta) * lambda3 * Dtheta_d + Km11 *
(Dpsir - Dpsir_d + lambda1 * (psir - psir_d)) + Km12 * (Dpsil - Dpsil_d + lambda2 *
(psil - psil_d)) + Km13 * (Dtheta - Dtheta_d + lambda3 * (theta - theta_d)) + Km21 *
(Dpsir - Dpsir_d + lambda1 * (psir - psir_d)) + Km22 * (Dpsil - Dpsil_d + lambda2 *
(psil - psil_d)) + Km23 * (Dtheta - Dtheta_d + lambda3 * (theta - theta_d)) + Km31 *
(Dpsir - Dpsir_d + lambda1 * (psir - psir_d)) + Km32 * (Dpsil - Dpsil_d + lambda2 *
(psil - psil_d)) + Km33 * (Dtheta - Dtheta_d + lambda3 * (theta - theta_d));

num3 = 0.65024e5 / 0.17161e5 * (0.119692505217e12 / 0.31250000000000e15 * (Dpsil -
Dpsir / 0.2e1) * cos(theta) + 0.39680127547e11 / 0.640000000000e14 * Dtheta) *
sin(theta) * (Dtheta_d - lambda3 * (theta - theta_d)) - 0.32512e5 / 0.17161e5 *
(0.119692505217e12 / 0.31250000000000e15 * Dpsil * cos(theta) - 0.39680127547e11 /
0.32000000000000e14 * Dtheta) * sin(theta) * (Dtheta_d - lambda3 * (theta - theta_d))
- 0.1001210671e10 / 0.125000000000e12 * lambda3 * Dtheta_d - 0.15200948162559e14 /
0.20948486328125000e17 * Dtheta * cos(theta) * sin(theta) * (Dpsil_d - lambda2 * (psil
- psil_d));

num4 = -0.15200948162559e14 / 0.20948486328125000e17 * sin(theta) * cos(theta) * (-
Dpsir + 2 * Dpsil) * (Dpsir_d - lambda1 * (psir - psir_d)) + 0.15200948162559e14 /
0.20948486328125000e17 * sin(theta) * Dpsil * cos(theta) * (Dpsil_d - lambda2 * (psil
- psil_d)) - 0.15200948162559e14 / 0.20948486328125000e17 * Dtheta * cos(theta) *
sin(theta) * (Dpsir_d - lambda1 * (psir - psir_d)) - 0.1032256e7 / 0.10725625e8 * (-
0.942460671e9 / 0.62500000000e11 * Dtheta * cos(theta) - 0.293652829e9 /
0.62500000000e11 * Dpsir + 0.293652829e9 / 0.62500000000e11 * Dpsil) * sin(theta) *
```

```
(Dpsir_d - lambda1 * (psir - psir_d)) + 0.4736326478941e13 / 0.10474243164062500e17 *  
sin(theta) * (-Dpsir + Dpsil) * (Dpsil_d - lambda2 * (psil - psil_d));
```

```
den = 0.1001210671e10 / 0.125000000000e12 + 0.293652829e9 / 0.62500000000e11 *  
cos(theta);
```

```
DDtheta_d=(num1+num2+num3+num4)/den;
```

```
taur = 0.160000e6 / 0.17161e5 * (-0.15200948162559e14 / 0.195312500000000000e18 *  
cos(theta) ^ 2 + 0.4379858184058151e16 / 0.125000000000000000e20) * (DDpsir_d -  
lambda1 * (Dpsir - Dpsir_d)) + 0.258064e6 / 0.10725625e8 * (0.942460671e9 /  
0.31250000000e11 * cos(theta) ^ 2 - 0.949333569e9 / 0.200000000000e13) * (DDpsil_d -  
lambda2 * (Dpsil - Dpsil_d)) + 0.293652829e9 / 0.125000000000e12 * cos(theta) *  
(DDtheta_d - lambda3 * (Dtheta - Dtheta_d)) + 0.15200948162559e14 /  
0.20948486328125000e17 * Dtheta * cos(theta) * sin(theta) * (Dpsir_d - lambda1 * (psir  
- psir_d)) - 0.4736326478941e13 / 0.10474243164062500e17 * sin(theta) * (-Dpsir +  
Dpsil) * (Dpsil_d - lambda2 * (psil - psil_d)) - 0.65024e5 / 0.17161e5 *  
(0.119692505217e12 / 0.3125000000000000e15 * (Dpsil - Dpsir / 0.2e1) * cos(theta) +  
0.39680127547e11 / 0.64000000000000e14 * Dtheta) * sin(theta) * (Dtheta_d - lambda3 *  
(theta - theta_d)) - Km11 * (Dpsir - Dpsir_d + lambda1 * (psir - psir_d)) - Km12 *  
(Dpsil - Dpsil_d + lambda2 * (psil - psil_d)) - Km13 * (Dtheta - Dtheta_d + lambda3 *  
(theta - theta_d));
```

```
taul = 0.258064e6 / 0.10725625e8 * (0.942460671e9 / 0.31250000000e11 * cos(theta) ^ 2  
- 0.949333569e9 / 0.200000000000e13) * (DDpsir_d - lambda1 * (Dpsir - Dpsir_d)) +  
0.160000e6 / 0.17161e5 * (-0.15200948162559e14 / 0.195312500000000000e18 * cos(theta)  
^ 2 + 0.4379858184058151e16 / 0.125000000000000000e20) * (DDpsil_d - lambda2 *  
(Dpsil - Dpsil_d)) + 0.293652829e9 / 0.125000000000e12 * cos(theta) * (DDtheta_d -  
lambda3 * (Dtheta - Dtheta_d)) + 0.1032256e7 / 0.10725625e8 * (-0.942460671e9 /  
0.62500000000e11 * Dtheta * cos(theta) - 0.293652829e9 / 0.62500000000e11 * Dpsir +  
0.293652829e9 / 0.62500000000e11 * Dpsil) * sin(theta) * (Dpsir_d - lambda1 * (psir -  
psir_d)) + 0.15200948162559e14 / 0.20948486328125000e17 * Dtheta * cos(theta) *  
sin(theta) * (Dpsil_d - lambda2 * (psil - psil_d)) + 0.32512e5 / 0.17161e5 *  
(0.119692505217e12 / 0.3125000000000000e15 * Dpsil * cos(theta) - 0.39680127547e11 /  
0.3200000000000000e14 * Dtheta) * sin(theta) * (Dtheta_d - lambda3 * (theta - theta_d))  
- Km21 * (Dpsir - Dpsir_d + lambda1 * (psir - psir_d)) - Km22 * (Dpsil - Dpsil_d +  
lambda2 * (psil - psil_d)) - Km23 * (Dtheta - Dtheta_d + lambda3 * (theta - theta_d));
```

```
output = [taur taul]';
```