2015 American Control Conference
Palmer House Hilton
July 1-3, 2015. Chicago, IL, USA

Design, Build, and Test of an Autonomous myRIO Based Segbot

Matt Migchelbrink, Warren N. White, Lucas Gorentz, Jacob Wagner, and Brian Blankenau

***Abstract*—This paper describes the design, programming, and construction of a battery operated autonomous Segbot. The Segbot, a two wheeled robot, has some similarities to the inverted pendulum cart system in that it is underactuated, exhibits non-minimum phase behavior, and requires a stabilizing controller to stand upright. The Segbot differs from the inverted pendulum cart system in that the dynamics are nonholonomic. The National Instruments® myRIO microcontroller is used for control implementation. The myRIO allows WiFi communication between the Segbot and a host computer for supervisory control. The LabView® VI used for real-time data acquisition and control consists of two main loops. The first loop collects state measurements from an accelerometer, rate gyros, and wheel position encoders. These measurements are passed to the second loop which calculates the desired control law. Included is a summary of the system identification procedure used to determine parameters that cannot be directly measured and the use of a CAD program to estimate inertial values and mass center location. An example of the Segbot tracking a figure-8 path with a novel navigation controller, derived herein, is presented. Finally, a URL link is included for CAD device drawings, a parts list, control programs, wiring diagrams, and a video demonstration of the Segbot.**

I. Introduction

The Segbot and other two wheeled balancing robots have been used in both research and academic settings [1, 2]. Much like the inverted pendulum cart, the basis of the device is to maintain an underactuated dynamic system in the vertical position. Unlike the inverted pendulum cart system, the Segbot exhibits a reaction torque on the pendulum body. The system is nonlinear, underactuated, nonholonomic, non-minimum phase, and it serves as a physical example of several different systems.

One of the first Segbot designs [3] was produced at the University of Illinois. This early Segbot used a TI C2000F28335 Delfino® floating-point microcontroller coupled with rate gyros, accelerometers, and other systems that made it possible to stabilize the robot in an upright position. The Univ. of IL Segbot is also capable of robust, high speed wall following, and a remote control mode that uses TI eZ430-Chronos Wireless Watch Development Tool.

M. Migchelbrink is with the Mechanical and Nuclear Engineering Department at Kansas State University, Manhattan, KS 66506 USA (e-mail: mmigchelb@gmail.com).

W.N. White is with the Mechanical and Nuclear Engineering Department at Kansas State University, Manhattan, KS 66506 USA (phone: 785-532-2615; fax: 785-532-7057; e-mail: wnw@ksu.edu).

L. Gorentz is with the Mechanical and Nuclear Engineering Department at Kansas State University, Manhattan, KS 66506 USA (e-mail: lgorentz@ksu.edu).

J. Wagner is with the Mechanical Engineering Department at the University of Illinois, Champaign, IL 61820 USA (e-mail: wjwagne2@illinois.edu).

B. Blankenau is with the Mechanical Engineering Department at the University of Illinois, Champaign, IL 61820 USA (e-mail: bjblank2@illinois.edu).

This paper reports the replacement of the TI Delfino® with the National Instruments myRIO. This replacement allows students to incorporate the interfacing and programming advantages of LabView. The substitution includes real-time gain tuning, turning on/off the Segbot at a button click, the ability to work within a GUI oriented programming language, the ability to include script-files or C code, and the ability to save on a host computer real-time measurements transmitted wirelessly. Besides stabilization, this paper provides a simple control scheme for tracking a desired planar trajectory.

Examination of well-known literature databases shows a wealth of information on the control of two wheeled mobile robots. The purpose of this paper is to illustrate the application of the myRIO to the navigation and control of the device so that it can be used in a teaching environment. The Segbot does present an opportunity for many nonlinear control applications, however the thrust of this paper is to present the Segbot as a second control theory class teaching tool. A linear control design will be presented. We have developed nonlinear control applications for the Segbot and those applications will be presented elsewhere. At the time of the writing of this paper, the myRIO has been commercially available for less than a year. From an examination of the literature at this time, no other myRIO applications to the Segbot have appeared.

II. Segbot Dynamics and Modeling

A. Basic Geometry

Fig. 1 shows a representation of the Segbot. In Fig. 1, the *X-Y* plane models the laboratory floor and trajectories in this plane will be referred to as *XY* trajectories consisting of coordinate histories measured in this frame. The given *XY* trajectory is the desired path the Segbot axle center will follow. The origin of the body-attached coordinate system, given by *xyz,* is located at the Segbot axle center. The *x* axis is aligned with the Segbot heading direction, while the *z* axis is vertical. The *x*-axis is found by a right hand rotation of ϕ measured from the global *X*-axis about the global *Z*-axis.

Because the pendulum body can lean and rotate, the coordinate set (x',y',z') is the body-fixed system and its origin is located at the Segbot mass center. The angle θ is the angle the Segbot body is leaning from the vertical and is measured in the right hand sense about the y-axis. The Segbot was assembled so that the center of mass is located an equal distance from the center of the left and right wheels, i.e. somewhere in the *x-z* plane. The Segbot is also assembled so that the center of mass is directly above the axle center when θ is zero. The angles ψ_r and ψ_l measure the right and left wheel rotations, respectively. The wheel

displacements are measured in the right hand sense about the y-axis. The system parameters used to describe the Segbot geometry and other physical quantities are listed in Table 1.

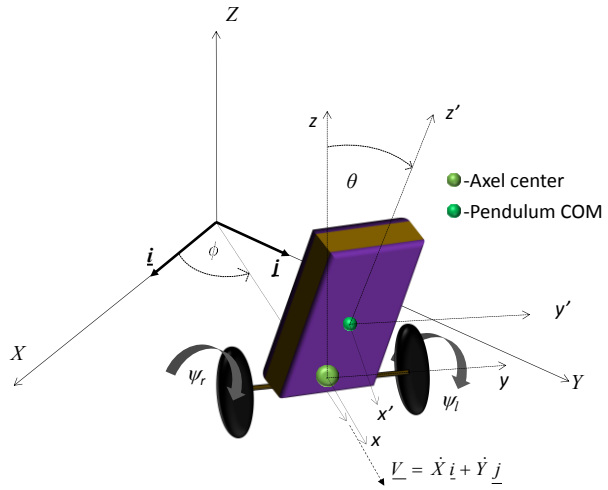

Figure 1: Segbot Coordinate Systems

Table 1: System Parameters

Symbol	Description	Value	Units
J	Mass Moment of Inertia for Wheel, Gearbox, and Motor	0.00157	kg-m^2
I_1	Mass Moment of Inertia of Body and Wheels about x' Axis	0.00153	kg-m^2
I_2	Mass Moment of Inertia of Body about y' Axis	0.00111	kg-m^2
I_3	Mass Moment of Inertia of Body and Wheels about 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 Pendulum	1.2398	kg
h_c	Distance from Axle Center to Pendulum Mass Center	0.0746	m
g	Acceleration of Gravity	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	ohms
N	Gear ratio	30	

B. *Dynamics*

The equations of motion were derived using Lagrangian mechanics in terms of two rigid body rotational angles, the two wheel angles, two X-Y axle center coordinates, and two Lagrangian multipliers associated with the nonholonomic constraint of the shaft center motion along the trajectory. The next step of the dynamic analysis is to reduce the number of variables. Note that the angle ϕ can be determined from the difference between the wheel displacements. Through substitution, the equations of motion can be reduced to three second order, nonlinear differential equations in time in terms of the tilt angle θ and the wheel rotational displacements ψ_r and ψ_l. The derivation details are contained in [4]. While the model development would be beyond the scope of a second controls class, the equations could be presented together with an outline of the derivation procedure. The end result would be to linearize the equations so that control design could be performed.

C. *Identification*

The Table 1 parameters W, m_w, R, m_p, and g can be directly measured. The CAD software Solidworks®, the same software used in the undergraduate graphics class, was used to create a detailed drawing of the Segbot. Part masses were measured using a laboratory balance. By finding the part volume from the CAD software, the density of the part could be estimated. This allowed for the estimation of inertial values and the mass center location. The Table 1 inertial quantities I_1, I_2, and I_3 together with the length h_c were found in this way. The procedure of mass, volume, and density determination was performed for all Segbot parts.

The Table 1 parameters K_m, K_t, R_a, and J were found through identification. The brushed DC motor torque is

$$\tau = K_t i_a \tag{1}$$

where τ is the developed torque in N-m and i_a is the armature current in amps. The DC motor voltage is

$$V = i_a R_a + K_m \dot{\psi} \tag{2}$$

where V is the voltage across the motor in volts and ψ is the motor shaft angular position in radians.

A regulated power supply provided a constant 12.08 volts to the motor driver. The duty cycle of a PWM input to the motor driver was then varied from 0 to 1. The wheel was allowed to spin freely while DC motor voltage and current were measured. Angular position was determined using an optical encoder and a discrete observer, using a fixed time step of 0.001 sec., estimated the motor angular velocity.

Motor characteristics were measured for two different motor operating configurations. The first configuration consisted of the wheel being placed in a locked rotor situation while various voltages were applied to the DC motor. The corresponding current was measured from the motor driver. Then the wheel was allowed to spin freely. The corresponding current, voltage, and wheel speed were measured.

The measurements from each motor configuration were inserted into (2) to calculate the best guess for the motor parameters via a least squares technique. The resistance is given in ohms, while the back emf constant has the units of volt-sec/rad. Table 1 shows the least squares analysis results. Because one motor model was desired, the least squares fit was performed together with data taken from both motors. Because SI units were used to find K_m, the torque constant K_t is the same value, but with units of N-m/amp.

The remaining unidentified Table 1 quantity is the motor, gearbox, and wheel collective mass moment of inertia. The Segbot was positioned so that the wheels were off the ground. There is no translation and each wheel rotates freely. A motor torque balance is written. Eqs.(1) and (2) are applied to this dynamic model providing the relation

$$J\ddot{\psi} = -C_d\dot{\psi} + \tau_{accel} = \frac{K_t}{R_a}\left(V - \left\{\frac{C_d R_a}{K_t} + K_m\right\}\dot{\psi}\right) \tag{3}$$

where C_d is the viscous damping coefficient and J is the collective inertia. Note that R_a, K_t, and K_m were determined earlier. Writing (3) in the frequency domain yields

$$\frac{\psi(s)}{V(s)} = \frac{\frac{K_t}{R_a J}}{s\left(s + \frac{C_d R_a + K_t K_m}{R_a J}\right)} = \frac{b_2}{s(s+b_1)} \tag{4}$$

where the b_i replace the ratios in the numerator and denominator. Providing a voltage step and inverting the resulting Laplace transform yields

$$\psi(t) = V\left(\frac{b_2}{b_1}t - \frac{b_2}{b_1^2} + \frac{b_2}{b_1^2}e^{-b_1 t}\right) \tag{5}$$

where t is the time in seconds, V is the magnitude of the voltage step input, e is the base of the natural logarithms,

$$b_1 = \frac{C_d R_a + K_t K_m}{R_a J}, \text{ and } b_2 = \frac{K_t}{R_a J}. \tag{6}$$

Again the least squares method is used to determine the unknown constants b_1 and b_2. By knowing these two constants, (6) provides an estimate for J. The estimate of the wheel position to a voltage step input is given by Equations (5) and (6). Comparing (5) to the measured result is the least squares basis. The value of J is listed in Table 1.

D. Linearized Dynamics

The dynamics of the Segbot were linearized about the point where the lean angle and lean rate are both zero, and the rate of change of both wheels is equal. Let the state vector be specified as $\underline{X} = \left[\psi_r \; \psi_l \; \theta \; \dot{\psi}_r \; \dot{\psi}_l \; \dot{\theta}\right]^T$. The linearized state equations are

$$\underline{\dot{X}} = \left[\dot{\psi}_r \quad \dot{\psi}_l \quad \dot{\theta} \quad \ddot{\psi}_r \quad \ddot{\psi}_l \quad \ddot{\theta}\right]^T = A\underline{X} + Bu =$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -141.75 & 0 & 0 & 0 \\ 0 & 0 & -141.75 & 0 & 0 & 0 \\ 0 & 0 & 196.43 & 0 & 0 & 0 \end{bmatrix} \underline{X} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 696.23 & 1248.88 \\ 148.88 & 696.23 \\ -372.72 & -372.72 \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}. \tag{7}$$

E. Torque Control

The voltage input is determined from the control law torque values. The torque applied by the left and right motors are used as inputs to the system as shown in (7). The torque input can be converted to a voltage input through the application of (1), (2), and using the desired torque and the motor angular velocity. Thus, the torque can be controlled by the application of the correct motor DC voltage.

A Pololu® dual MC33926 motor driver is used to apply the calculated voltages to the motors. The motor driver is configured for a drive-brake operation. In this configuration the current flows through the motor when the pulse width modulation (PWM) motor input is high, and the leads to the DC motor are shorted together when the PWM signal is low. This allows the linear relationship of $V_{out} = V_s D_{in}$ where D_{in} is the duty cycle of the PWM motor input which ranges over 0-1.0, V_{out} is the average voltage supplied to the DC motor, and V_s is the regulated motor driver supply voltage.

F. State Determination Loop

The myRIO has three data acquisition ports. The data acquisition is accomplished through an on-board field programmable gate array (FPGA) which has a factory shipped standard configuration or "personality". This FPGA can be reprogrammed to accommodate feedback transducers that are not part of the standard package. The IMU is one such device and more will be described later in this section.

There are two main loops used in the LabView Virtual Instrument (VI). These loops are shown in Fig. 2. The first is a feedback acquisition loop. This loop runs on available processor time. The main objective of this loop is to obtain and parse state measurements. The first loop runs faster than the second loop. The purpose of the second loop is to realize the real-time control where feedback information is used to evaluate the control law and issue the necessary voltages through the D/A converters. The second loop also allows for supervisory control where commands can be issued from the host computer to turn the controller on or off, change the nature of the control (tracking, stabilization, etc.), and change control gains. The second loop executes at 100 Hz.

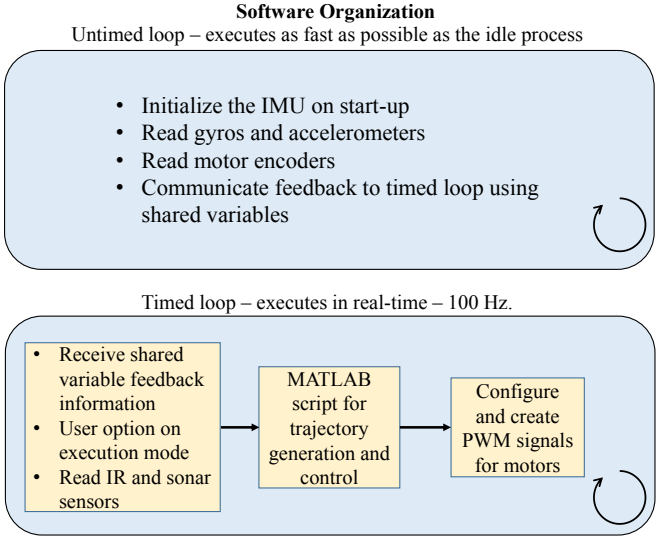

Figure 2: Control Code Structure

Optical encoders are mounted on each motors. The resolution is 420 counts per rev. before the X4 quadrature decode. The acquisition and decode is performed by the FPGA. The wheel angle calculated by the state acquisition loop is made a "shared variable" which means that it can be passed to the second loop which calculates the control law.

The state acquisition loop runs on available processor time. I^2C protocol is used to streamline data acquisition from a Sparkfun® MPU 6050 3-axis accelerometer and gyroscope. The FPGA had to be reprogrammed to allow for the IMU. The gyroscope measurements are directly used to estimate the pendulum body angular velocity. The estimate of θ is calculated by a complementary filter of the form

$$\theta_{actual} = 0.998\,\theta_{gyro} + 0.002\,\theta_{accel} \tag{8}$$

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 [5] for more information. The estimate for θ_{actual} was also made a shared variable so it could be passed to the control loop.

G. Timed Loop

The second loop is a timed loop. The main purpose of this loop is to allow the user the freedom to choose an operating mode and conditions, calculate desired trajectories if need be, and calculate the control law. Many individual parameters are passed to the myRIO timed loop from the host computer using LabVIEW utilities. Also passed are the state measurements determined by the state acquisition loop. The control loop also has a discrete observer to estimate of the angular wheel velocities using the angular displacement information passed via shared variables from the measurement loop to the timed loop. Currently, the IR and sonar sensors are read in the timed loop.

The control law evaluation is performed in the second loop by a math script node which consists of a MATLAB® script file. This provides convenience because the script file is easy to create and debug. This is the main function used in the timed control loop. A simple series of programming statements can be used to affect the control output of the timed loop, the simplest of which is the case of stabilizing the Segbot. The MATLAB script is also used to implement the tracking controller. Following the math script node, the required pulse width modulated (PWM) signals are created for the motor driver.

This control code started with a published NI example of the myRIO being used with the Sparkfun MPU-6050 IMU. The published code, documentation, and hardware details are found at: https://decibel.ni.com/content/docs/DOC-30443

III. Control Issues

A. Stabilization

Bryson's rule [6] was used to determine an initial set of linear state feedback gains to stabilize the dynamic system. The weighting matrices were defined as

$$Q = \begin{bmatrix} \frac{1}{\psi_{\max}^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\psi_{\max}^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\theta_{\max}^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\dot{\psi}_{\max}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\dot{\psi}_{\max}^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\dot{\theta}_{\max}^2} \end{bmatrix}$$

and

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{1}{b_2^2} \tag{9}$$

where $\psi_{max} = 10$, $\theta_{max} = 0.05$, $\dot{\psi}_{\max} = 10$, $\dot{\theta}_{\max} = 5$, b_2 is from (6), and $V_{max} = 10$ were used along with the MATLAB command *K=-lqr(A,B,Q,R)* to determine the linear quadratic regulator feedback gains of

$$K = \begin{bmatrix} 0 & 0.047 & 8.848 & 0.035 & 0.070 & 0.290 \\ 0.047 & 0 & 8.848 & 0.070 & 0.035 & 0.290 \end{bmatrix}. \tag{10}$$

In implementation, the state feedback gains from (10) could stabilize the Segbot. However, the values were altered slightly to yield a more desirable response. This process was performed by varying the feedback gain values as the Segbot operated by using the supervisory control functions on the host computer. The final linear gains used to stabilize the Segbot are

$$K = \begin{bmatrix} 0 & 0.12 & 10.27 & 0.026 & 0.14 & 0.19 \\ 0.12 & 0 & 10.27 & 0.14 & 0.026 & 0.19 \end{bmatrix}. \tag{11}$$

B. Trajectory Tracking

Once the Segbot was stabilized, the main control design task is the trajectory tracking control of the axle center.

The nonholonomic relationships between the wheel angular velocities and the global velocity, shown in Fig. 1, are

$$\dot{X} = \frac{R(\dot{\psi}_r + \dot{\psi}_l)}{2}\cos(\phi) \text{ and } \dot{Y} = \frac{R(\dot{\psi}_r + \dot{\psi}_l)}{2}\sin(\phi). \tag{12}$$

The Segbot is a nonholonomic dynamic system and as a result cannot move sideways without violating the nonholonomic constraint. In order to reduce the influence of disturbances on the tracking error, a navigational control must be added to the Segbot trajectory so the robot can eventually join the correct path. The navigational control receives the current axle center position, heading direction, and the desired trajectory. The navigational control then produces a compensating angular velocity for each wheel so that the path is regained. For the case of stabilization, the desired velocities in the control loop are set to zero.

The derivation for the navigational control is rather straight forward as outlined in [7]. A variation of the derivation is presented here that produces better tracking. The development of the navigational control begins with an object that is subject to the nonholonomic constraints of

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \dot{\phi} \end{bmatrix} \tag{13}$$

where v is the translational speed, ϕ is the heading direction, and $\dot{\phi}$ is the rate of change in heading. Fig. 3 shows the actual position of the object as a triangle.

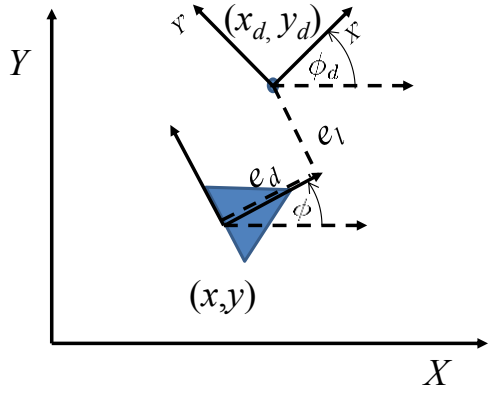

Figure 3: Navigational Control Posture Error

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

The technique of [7] 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 Fig. 3 as

$$P = \begin{bmatrix} e_d \\ e_l \\ \phi_e \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \phi_d - \phi \end{bmatrix}. \quad (14)$$

With some trigonometric simplifications the time derivative of the vector in (14) can be written as

$$\dot{P} = \begin{bmatrix} \dot{e}_d \\ \dot{e}_l \\ \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} e_l\dot{\phi} - v + v_d \sin(\phi_e) \\ -e_d\dot{\phi} + v_d \sin(\phi_e) \\ \dot{\phi}_d - \dot{\phi} \end{bmatrix} \quad (15)$$

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_d is the compensational translational speed found by the navigational control, v is the actual translational speed, and the subscript e stands for the desired coordinate minus the actual coordinate.

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

$$V = \frac{1}{2}\left(e_d^2 + e_l^2 + \frac{1-\cos(\phi_e)}{k_2}\right) \quad (16)$$

This can be observed to be non-negative in the errors and V describes a positive definite system. The time derivative of (16) while observing the results of (15) can be shown to be

$$\dot{V} = e_d\dot{e}_d + e_l\dot{e}_l + \frac{\sin(\phi_e)\dot{\phi}_e}{2k_2}$$
$$= e_d\left(e_l\dot{\phi} - v + v_d\cos(\phi_e)\right) + e_l\left(-e_d\dot{\phi} + v_d\sin(\phi_e)\right) \quad (17)$$
$$+ \frac{\sin(\phi_e)\left(\dot{\phi}_d - \dot{\phi}\right)}{2k_2}.$$

The development in [4] proposes the control law of

$$v = v_d\cos(\phi_e) + k_1 e_d + k_4 e_d e_l^2$$

and

$$\dot{\phi} = \dot{\phi}_d + v_d\left(k_2 e_l + k_3\sin(\phi_e)\right) + k_5\sin(\phi_e)e_l^2 \quad (18)$$

where the k_i are all positive constants used as control parameters. In [7] there are only three control parameters. Substituting the proposed control laws into the Lyapunov time derivative shows

$$\dot{V} = -k_1 e_d^2 - k_4 e_d^2 e_l^2 - \frac{k_3}{k_2}v_d\sin(\phi_e)^2 - \frac{k_5\sin(\phi_e)^2 e_l^2}{k_2} \quad (19)$$

which shows that V is positive definite and $\dot{V}$ is negative definite. In [7], $\dot{V}$ was only negative semi-definite.

The trajectory tracking control has the same linear feedback gains as the stabilizing control in (11). However, $\psi_{rd} = \psi_r$ and $\psi_{rd} = \psi_r$, while $\dot{\psi}_{rd}$ and $\dot{\psi}_{ld}$ are determined by the navigational control. For stabilization all of the desired states in the navigational control are set to zero.

The last step is to choose gains for the k_i that are used in the navigational control. In [4, 7] the gain k_3 is set as $k_3 = 2\sqrt{k_2}$. Then appropriate gains must be found for the other k_i. The Segbot was given the figure-8 trajectory of

$$X_d = r\frac{\cos(\omega t)}{1+\sin(\omega t)^2} \text{ and } Y_d = r\frac{\cos(\omega t)\sin(\omega t)}{1+\sin(\omega t)^2} \quad (20)$$

as the desired trajectory where ω is the angular frequency of the figure-8 and $2r$ is the width across the figure-8. The result in (20) can be differentiated to determine desired trajectory time derivatives and the desired heading direction can be found by rearranging the nonholonomic constraint as

$$\phi = \tan^{-1}\left(\frac{\dot{Y}}{\dot{X}}\right). \quad (21)$$

IV. Applications

After the Segbot completes a few figure-8 revolutions to overcome initial conditions, the trajectory converged to a steady-state path. The effects of changing k_1 and k_2 are shown in Figures 4 and 5. The intent of Figures 4 and 5 are to illustrate the influence of the control constants.

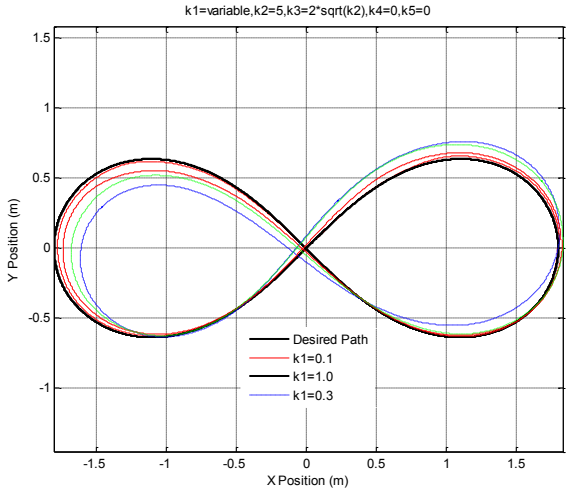

Figure 4: Effects of Modifying k_1

The Segbot tracking can be improved by the proper gain choices as shown in Figure 6 which demonstrates that the new tracking control technique produces a better result.

V. Segbot Construction

In building the Segbot, the aim was to replace the microcontroller. Use was made of the existing framework and peripheral parts described in [3]. Although we started with the Segbot shell, many design changes were introduced as the construction progressed. Fig. 7 shows the completed device. Information, such as CAD drawings of the Segbot,

pdf of the CAD drawings, a parts list, the LabVIEW controller, pictures, a wiring diagram, and video showing the Segbot operation, is available at [8].

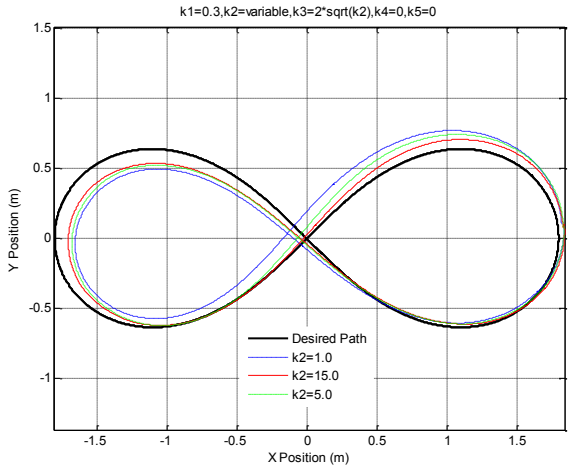

Figure 5: Effects of Modifying k_2

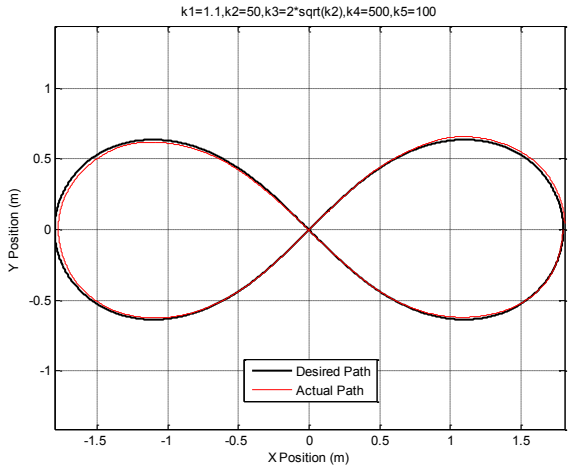

Figure 6: Good Segbot Tracking Performance

VI. Conclusions

An autonomous Segbot has been built and tested. The robot was constructed with off-the-shelf parts and the cost was kept to a minimum. The implementation of the NI myRIO as a microcontroller allows a convenient means of WiFi communication between the robot and host computer.

The KSU Segbot resulted from the work of a group of undergraduates and one graduate student. Most of the students volunteered their time in this effort.

This paper provides the details of model development, identification, stabilizing control, and tracking control.

The unique dynamics of the Segbot can appeal to a range of students. The Segbot can provide an opportunity for graduate study in non-linear controls, digital controls, or nonholonomic tracking. If the dynamics are simplified so the left and right wheel turn together, the Segbot dynamics behave similarly to the inverted pendulum cart which may be better suited for an introductory controls class.

The LabVIEW environment allows a simplification of programming and sensor interfacing details, an important consideration for students to complete their projects in a timely fashion. This simplification provides the opportunity to concentrate on control issues. The material presented here is better suited for a controls class as opposed to a mechatronics class because of the fewer interface details.

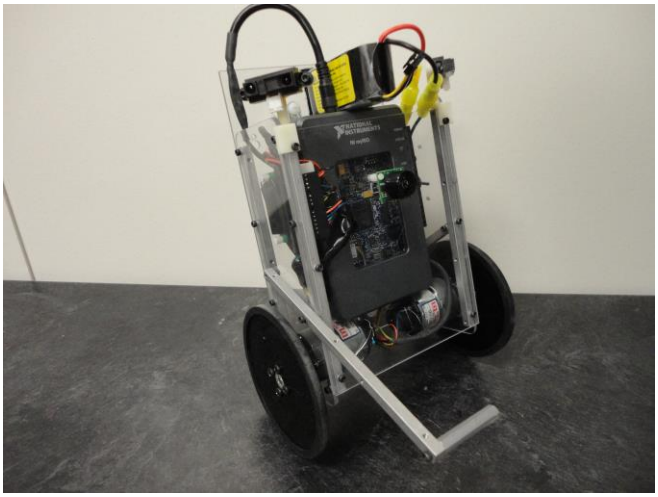

Figure 7: Completed Segbot

VII. Acknowledgements

The authors gratefully acknowledge the KSU students: Kris Larson and Skyler Butler for our website video, Sergio Ortiz and Alan Ramirez for their creation and management of our website, and Cameron Lucero for the wiring diagram.

VIII. References

[1] Hau-Shiue Juang and Kai-Yew Lum, "Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller Board, "2013 10th IEEE International Conference on Control and Automation (ICCA), June 12-14, 2013, pp. 634-639.

[2] He Bin, Liu Wen Zhen, and Lv Hai Feng, "The Kinematics Model of a Two-wheeled Self-balancing Autonomous Mobile Robot and Its Simulation," 2010 Second International IEEE Conference on Computer Engineering and Applications, pp. 64-68.

[3] The Segbot Home Page – College of Engineering Control Systems Laboratory, http://coecsl.ece.illinois.edu/segbot/segbot.html, last accessed 03/15/2015.

[4] Matt Migchelbrink, "Sliding Mode Control Trajectory Tracking Implementation on Underactuated Dynamic System," M.S. thesis, Mechanical and Nuclear Engr. Dept., Kansas State University, Manhattan, KS August, 2014.

[5] "Gyroscopes and Accelerometers on a Chip, http://www.geekmomprojects.com/gyroscopes-and-accelerometers-on-a-chip/ , last accessed 03/15/2015.

[6] G.F. Franklin, J.D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Upper Saddle River, NJ: Pearson, 6th ed., 2010, p. 463.

[7] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A Stable Tracking Control Method for an Autonomous Mobile Robot," 1990 IEEE International Conference on Robotics and Automation, May 13-18, Cincinnati, OH, pp. 384-389.

[8] The KSU Mechanical and Nuclear Engr. Dept. Segbot Website - http://www.mne.ksu.edu/research/laboratories/dynamic-systems-controls-laboratory-1/segbot, last accessed 03/15/2015.