Implementing Flexible Evasion Dynamics in One Dimension Using a Tuned Second-Order Response

Z. Jin, A. Rogers, M. Allain

I. Objective and Significance

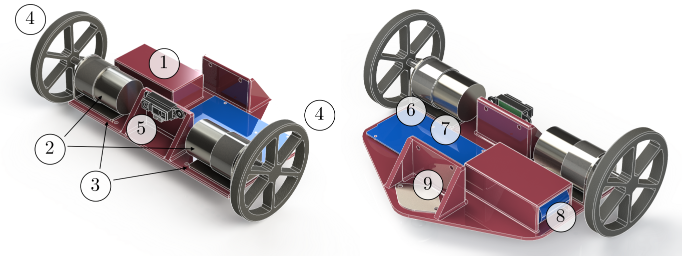
The designed and constructed robot is capable of evading obstacles in its environment. There are several applications for this device. A robot navigating an area of high traffic is required to avoid obstacles such as people and cars. An evading robot may be used in sports conditioning and exercising by giving a group the task of catching the evading robot. This could improve team synergy as well as individual physical fitness. A simple evading robot could be used as an exercise for young children developing motor skills by crawling and walking after the robot. Robot swarms must navigate without individual robots colliding. A robot evading those around it could also be used as an entertaining prank: picture the frustration that a dollar bill attached to a fishing line has caused the victims of this prank.

The dynamic control of a robot evading many surrounding obstacles would be beyond the scope of this class. Therefore, this problem has been reduced such that the concept may be demonstrated adequately. The created robot evades a single obstacle in a single, linear dimension. This simplified approach allows for adequate demonstration of the evading methodology used. The evading methodology is designed such that a robot of appropriate complexity could avoid any number of obstacles in full 3-D space.

The evasion methodology is such that the robot responds as if there were a spring and damper between itself and the obstacle. This allows for flexible design of the desired response. For example, the virtual spring between the robot and the obstacle can be specified as stiff such that the robot maintains an equal distance between itself and the obstacle. The spring constant and damping coefficient can be chosen such that the robots departure from the obstacle either includes or does not include overshoot and damping oscillations.

II. Design and Development

The final design utilizes a 3D-printed chassis, with hole arrays for each component. Detailed drawings of the chassis can be found in Appendix B. The 120 RPM motors (Pololu) are mounted via brackets to the chassis, and directly coupled to the wheel hubs (Pololu). A ball and socket caster was used as a third rolling contact in the rear of the assembly. A single 1500 mAh LiPo battery (Turnigy) lies within a pocket in the chassis. The IR sensor (Sharp) is mounted to a printed vertical support. Fig. 1 and Table I provide a CAD model and bill of materials of the final assembly.



(a) Front isometric view

(b) Rear isometric view

Fig. 1: CAD model of final design assembly

A. Dynamic Modeling

A diagram of the evading robot is shown in Fig. 2, and a free-body diagram is shown in Fig. 3. The car dynamics can be derived using Newton's Law as:

$$m_c \ddot{x} = F_x \tag{1}$$

Created as partial requirement for MEEN 667, Mechatronics, in the Dept. of Mechanical Engineering, Texas A&M University

Manuscript created November 26, 2016, revised December 5, 2016.

Part No.	Manufacturer	Model
1	N/A	3D printed chassis
2	Pololu	12 V 150 RPM DC motors
3	Pololu	Aluminum L-bracket motor mounts
4	Pololu	Plastic wheels, ∅90mm, with hubs
5	Sharp	Long range IR sensor
6	Arduino	Uno
7	Adafruit	Motor shield v2.3
8	Turnigy	1500 mAh LiPo Battery
9	N/A	Ball and socket caster

TABLE I: Bill of materials

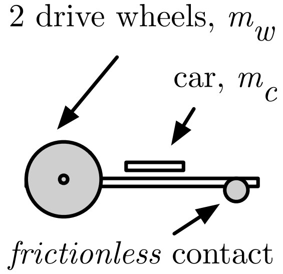


Fig. 2: The differential drive robot

and for the wheels:

$$2I_w\ddot{\theta} = 2T - fr \tag{2}$$

$$2m_w\ddot{x} = f - F_x \tag{3}$$

The no-slip condition gives us the kinematic constraint:

$$x = r\theta \tag{4}$$

Rearranging the equations of motion gives:

$$\left[\frac{I_w}{r} + \left(\frac{1}{2}m_c + m_w\right)r\right]\ddot{x} = T \tag{5}$$

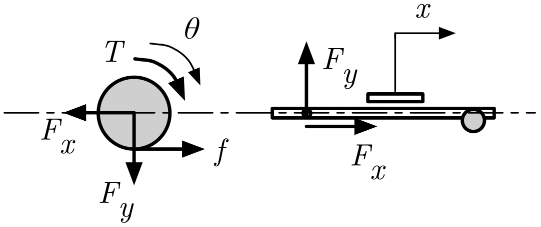


Fig. 3: Free body diagram of the robot

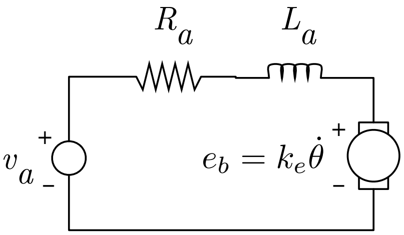


Fig. 4: DC motor circuit

The DC motor circuit is shown in Fig. 4. The inductance is very small and the motor dynamics are much faster than the vehicle dynamics, so neglecting the inductance gives:

$$v_a = R_a i_a + k_e\dot{\theta} \tag{6}$$

$$T = k_t i_a \tag{7}$$

Eqs. 5 and 6 give the open-loop system dynamics:

$$M\ddot{x} + C\dot{x} = v_a \tag{8}$$

where:

$$M = \left[\frac{I_w}{r} + \left(\frac{1}{2}m_c + m_w\right)r\right]\frac{R_a}{k_t}$$

$$C = \frac{k_e}{r}$$

The system transfer function can be written as:

$$G(s) = \frac{X(s)}{V_a(s)} = \frac{1}{Ms^2 + Cs} \tag{9}$$

B. System Parameters

The electrical resistance of the motor can be approximated by dividing the rated voltage by the stall current (at the rated voltage). The electromotive force constant, k_e, can be approximated by dividing the rated voltage by the free-run speed (at the rated voltage). To approximate the motor torque constant, k_t, one can divide the stall torque by the stall current. The relevant motor data is given in Table II, and the calculated system parameters are shown in Table III.

Parameter	Value		Converted Value
Gear ratio	70:1		
Free-run speed (12 V)	150 RPM	=	15.7 rad/s
Free-run current (12 V)	300 mA	=	0.3 A
Stall current (12 V)	5000 mA	=	5 A
Stall torque (12 V)	200 oz∗in	=	1.4 N∗m

TABLE II: Relevant Motor Parameters (Pololu)

These parameters yield the following transfer function coefficients for Eq. 9:

$$M = 0.1728$$

$$C = 16.98$$

Parameter	Value
k_t	0.28 Nm/A
k_e	0.764 Vs/rad
R_a	2.4 Ω
m_c	0.896 kg
$m_w = I_w$	0

TABLE III: System Parameters

C. Digital Filter Design

For the IR sensor data, a digital filter was designed using Tustin's discretization method for the following first-order continuous-time filter, [1]:

$$F(s) = \frac{1}{\frac{s}{\omega_c} + 1} \tag{10}$$

where ω_c is the filter cutoff freqeuncy. After discretization with the sampling period T, the Z-domain transfer function for the digital filter is given as:

$$F(z) = \frac{\omega_c T + \omega_c T z^{-1}}{(\omega_c + 2) + (\omega_c T - z)z^{-1}} \tag{11}$$

which corresponds to the following difference equation:

$$s_o[k] = C_1 s_i[k] + C_2 s_i[k-1] - C_3 s_o[k-1] \tag{12}$$

where

$$C_1 = \left(\frac{\omega_c T}{\omega_c + 2}\right) \tag{13}$$

$$C_2 = \left(\frac{\omega_c T}{\omega_c T + 2}\right) \tag{14}$$

$$C_3 = \left(\frac{\omega_c T - 2}{\omega_c T + 2}\right) \tag{15}$$

Sec. III demonstrates the sensor data before and after filtering.

D. Controller Design

In order to achieve a tunable, approximate second-order response from the closed-loop system, the necessary controller structure is given by:

$$D(s) = \frac{K}{s^2 + As + B} \tag{16}$$

where k, A, and B are design parameters. Using the plant transfer function in Eq. 9 yields a closed-loop transfer function:

$$G_{CL}(s) = \frac{D(s)G(s)}{1 + D(s)G(s)} = \frac{b(s)}{a(s)} \tag{17}$$

and a characteristic polynomial given by:

$$Ms^4 + (C + AM)s^3 + (AC + BM)s^2 + BCs + K$$

The desired closed-loop characteristic polynomial is:

$$a_d(s) = \theta(s + \alpha)(s + \beta)(s^2 + \gamma s + \phi) \tag{18}$$

where θ, α, β, γ, and ϕ are constrained by the following conditions:

$$\gamma = 2\bar{\zeta}\bar{\omega}_n \tag{19}$$

$$\phi = \bar{\omega}_n^2 \tag{20}$$

$$\alpha = 10\bar{\omega}_n \tag{21}$$

These conditions are selected such that the real poles (the control poles) are fast, and the complex pair (the design poles) follow the second order response dictated by design parameters $\bar{\omega}_n$ and $\bar{\zeta}$. For the rest of this paper, the design parameters $\bar{\omega}_n$ and $\bar{\zeta}$ will just be referred to as ω_n and ζ. Matching terms in Eq. 18 and the denominator of Eq. 17 and using conditions in Eqs. 19-21 fully determines the controller parameters, as shown in Appendix A.

Once controller parameters are determined, Tustin's approximation is used to discretize the controller, yielding the following Z-domain transfer function:

$$C(z)\frac{b_1 + b_2 z^{-1} + b_3 z^{-2}}{a_1 + a_2 z^{-1} + a_3 z^{-2}}$$

and corresponding difference equation:

$$u[k] = \frac{1}{a_1}\Big[-a_2 u[k-1] - a_3 u[k-2] \\ + b_1 e[k] + b_2 e[k-1] + b_3 e[k-2]\Big]$$

where

$$a_1 = \frac{4}{T^2} + \frac{2A}{T} + B$$

$$a_2 = 2B - \frac{8}{T^2}$$

$$a_3 = \frac{4}{T^2} - \frac{2A}{T} + B$$

$$b_1 = K$$

$$b_2 = 2K$$

$$b_3 = K$$

The closed-loop control system is shown in Fig. 5, see [2].

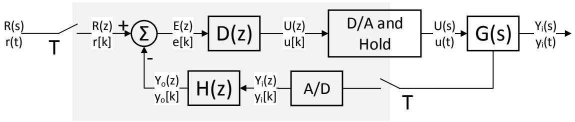


Fig. 5: The closed-loop digital control system.

E. Software Design

Fig. 6 depicts the logic flow used to control the robot. After declaring variables, the position of two potentiometers are determined on a scale of 0 to 100%. The desired closed-loop damping ratio and natural frequency are then determined from the potentiometer positions. One pot changes the damping ratio within the range of 0.4 to 1.0. The other pot changes the natural frequency in the range of 2 to 7 rad/sec. Note that the potentiometer position is only used in the setup portion of the logic. Once the controller is running, the potentiometer position will not be read again. The desired closed-loop natural frequency and damping ratio are used to calculate the control parameters and the coefficients in the digital control difference equation during the setup portion of the logic. Refer to Sec. II-D for details. The coefficients for the digital filter are also calculated during the setup portion of the logic.

When the setup portion is completed, the logic will switch between an interrupt service routine (ISR) and the main loop. The ISR is connected to a timer with a period of 10 ms. During the ISR, the sensor is read and filtered, the error and the control input are calculated, and bookkeeping is performed. Bookkeeping includes ensuring that the current values are saved as previous values for the next ISR iteration. When the ISR is completed, the main loop with be continually executed until the ISR is triggered again. The main loop implements the control input (voltage) to the DC motors and then checks if the total time of operation exceeds 25 seconds. If the total time of operation does exceed 25 seconds, the motors are stopped, and the robot waits for a manual reset. This condition is used for demonstration purposes in order to limit the time of demonstration, and avoid issues associated with catching a moving robot.

Once the robot receives a manual reset, the setup portion is executed again. This provides a convenient way for users to adjust the potentiometers and allow the robot to calculate the new controller based on the adjusted closed loop natural frequency and damping ratio.

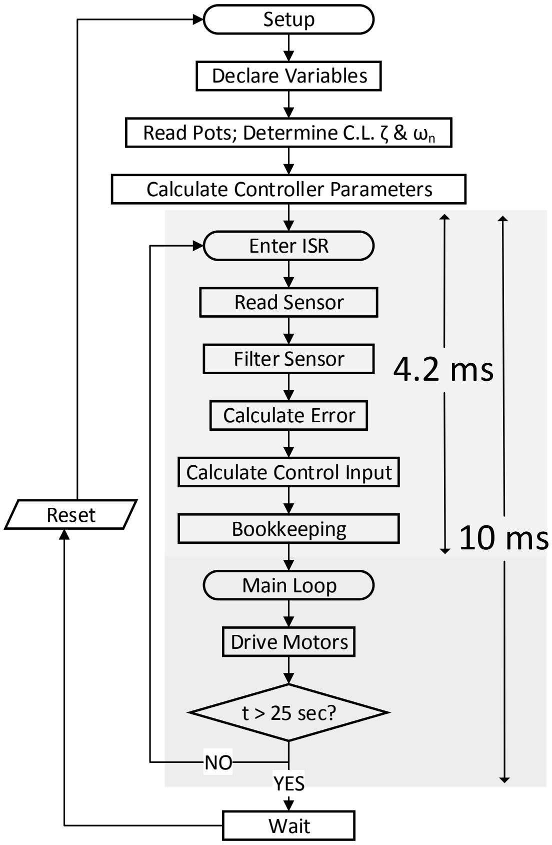


Fig. 6: The control logic for the tunable system.

III. Experimental Results

A. Sample Rate Testing

The time required to execute the logic and computations in the ISR was determined in order to select an appropriate sample rate. This was done by moving the code in the ISR to the main loop and allowing it to execute as fast as possible. A time stamp was recorded a the beginning and after 100 cycles. The difference between the time stamps was then divided by 100 in order to determine the time per cycle. This allows the maximum possible sample rate to be determined while adding minimal processing in addition to the ISR code. Additional processing includes incrementing an iteration counter and checking if this counter is equal to 100 at the end of each cycle. The time required to execute the ISR code was estimated using this methodology to be 4.1 ms. Therefore, the maximum possible sampling frequency is approximately 240 Hz. A sampling frequency of 100 Hz was chosen in order to leave sufficient time for additional processes and because a faster sampling rate is unnecessary for the current application.

B. *Digital Filter Results*

Before integrating the sensor into the control loop, tests were run to examine the effect of noise on the sensor output. The unfiltered sensor data and its corresponding discrete fourier transform (DFT) are shown in Fig. 7. Discretizing a first-order filter with $\omega_c = 1$ rad/s, and applying it to the measurement data gives the filtered time-series and DFT shown in Fig. 8.

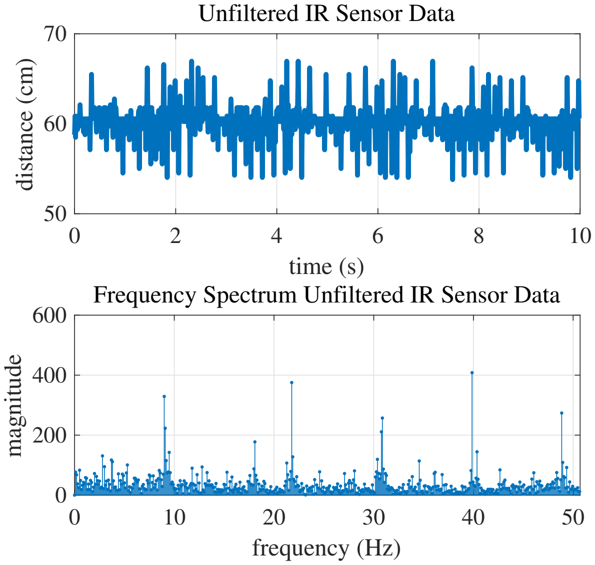


Fig. 7: Unfiltered sensor data

C. *Motion Estimation with Object Tracking*

Although the controllers performed well qualitatively, some measurement of the system output was necessary to validate the control performance characteristics. For the experimental responses, a step input was applied by placing an object in front of the sensor.

One available source of measurement data is the IR sensor. However, capturing this data would require modifying the real-time control program to facilitate data logging (at the expense of execution time), and the IR sensor lacks precision. One non-intrusive measurement alternative is motion estimation with computer vision (CV) algorithms. Our method leveraged a readily-available sensor (smartphone camera) and OpenCV[1], an open-source computer vision toolbox with Python bindings.

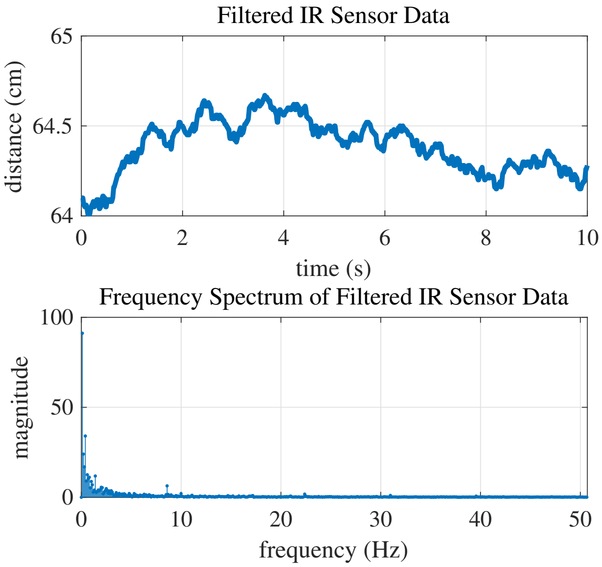


Fig. 8: Sensor data after a digital filter is applied

A small green color flag was adhered to the robot body before trials. The color value thresholds for this flag are passed to the CV algorithm, which masks each frame and computes the centroid of the flag.

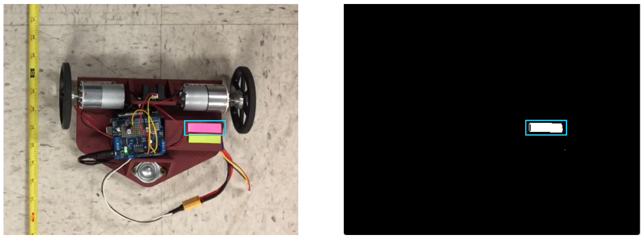
D. *Design of Experiment and Results*

To model a step input to the system, the robot is first allowed to come to steady-state against a static object. Then, another large flat object was placed abruptly in front of the robot a distance of 30 cm in front of the other object. This procedure represents a step input to the system on the order of 30 cm. Several responses are recorded with the camera, for the trials shown in Table IV. These trials were selected to test a range of damping ratios and rise times ($t_r \propto \omega_n$) with the real system. A scale is placed in the video frame in order to scale the motion capture data. A single frame of video and the corresponding color mask are shown in Fig. 9.

[1] http://opencv.org

Trial No.	ζ	ω_n (rad/s)
1	1	2
2	0.3	2
3	0.707	2
4	0.707	4
5	0.707	7

TABLE IV: Experimental trials


(a) Original frame of video

(b) Frame with mask

Fig. 9: Illustration of color masking a single frame

The color masking failed only in a few trials, where the robot left the video frame due to controller overshoot. Sample data from Trial 1 with several consecutive responses (back and forth) is shown in Fig. 10

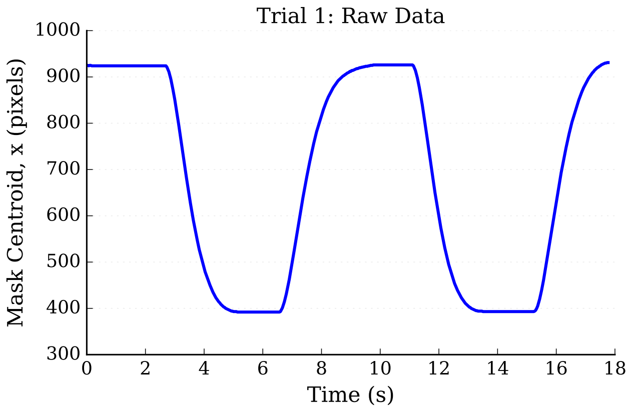


Fig. 10: Sample data from Trial 1.

One step response was taken from each trial and the steady-state positions were assumed perfect, i.e., zero steady-state error. The pixel locations, as in the y-axis of Fig. 10 were then mapped to the input distance of 30 cm. The resulting response for each controller trial is shown in Fig. 11. For comparison, a step response for the corresponding continuous-time version of these controllers is shown in Fig. 12.

Trials 1-3 agree very well with the simulation results. However, it is evident that the simulation is less faithful to the actual response at larger ω_n values. This corresponds to faster rise times and is caused by non-linear saturation of the motor voltage and the effect of filter dynamics on the closed-loop system behavior.

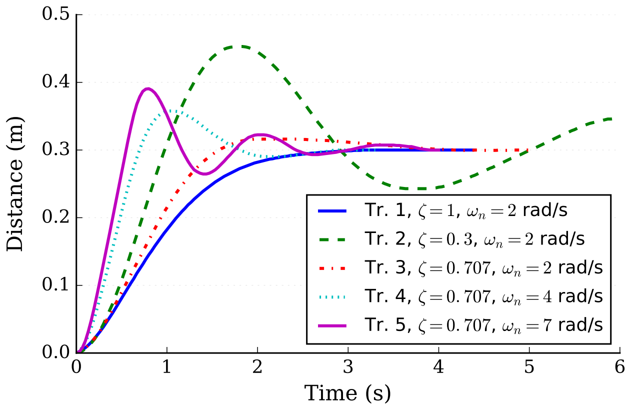


Fig. 11: Experimental motion capture results from all trials.

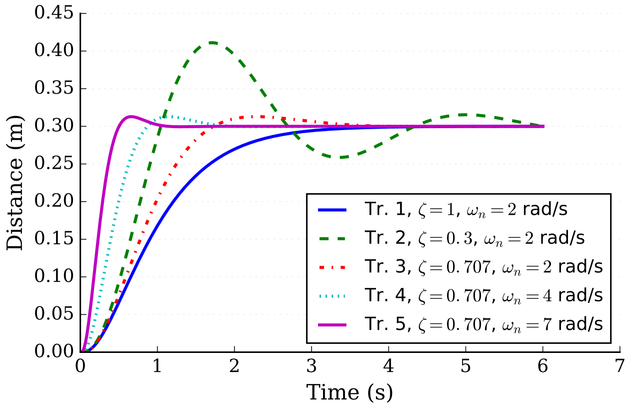


Fig. 12: Step responses with corresponding continuous-time controllers from each trial.

IV. Discussion and Conclusions

From the proposal, the anticipated results included (1) maintaining a desired steady state distance from a single object in one dimension and (2) a flexible dynamic response based on a virtual mass-spring-damper system. The proposed robot was to be capable of various damping ratios and response times. This report has demonstrated the anticipated results have been met entirely. The robot that has been designed and built uses a single IR sensor to measure the distance to a single object. The robot then uses two DC motors to maintain a distance from the object. The digital controller was created by transforming a continuous time controller using Tustin's method. The continuous time controller is entirely novel, and the controller successfully allows for a flexible dynamic response. The closed loop response of the robot to a step input was captured using computer vision methods. Several damping ratios and natural frequencies have been demonstrated. The flexible damping ratio allows the robot to exhibit overshoot, damped oscillations, and critically damped behavior. The flexible natural frequency allows the robot to exhibit fast and slow responses.

Appendix

A. *Individual Contributions*

Early in the project, I was responsible for the design and 3D printing of the robot chassis. I also prepared the final assembly CAD model. I developed the early pole-placement approach which allowed for a flexible second order response. In order to validate the controllers, I assembled a computer vision routine using OpenCV in Python and recorded and plotted the output data. Throughout the project, I also prepared two Jupyter notebooks to closely document my design and analysis.

Mitchell Allain

Dynamic modeling of the system. Motor coding (input mapping, step, ramp, harmonic functions) design and tests. Sensor coding (voltage to distance mapping), filter design, implementation and tests. System integration: 1 DOF controller and filter pairs (slow/fast) design, implementation and tests, 2 DOF tracking design, implementation and tests.

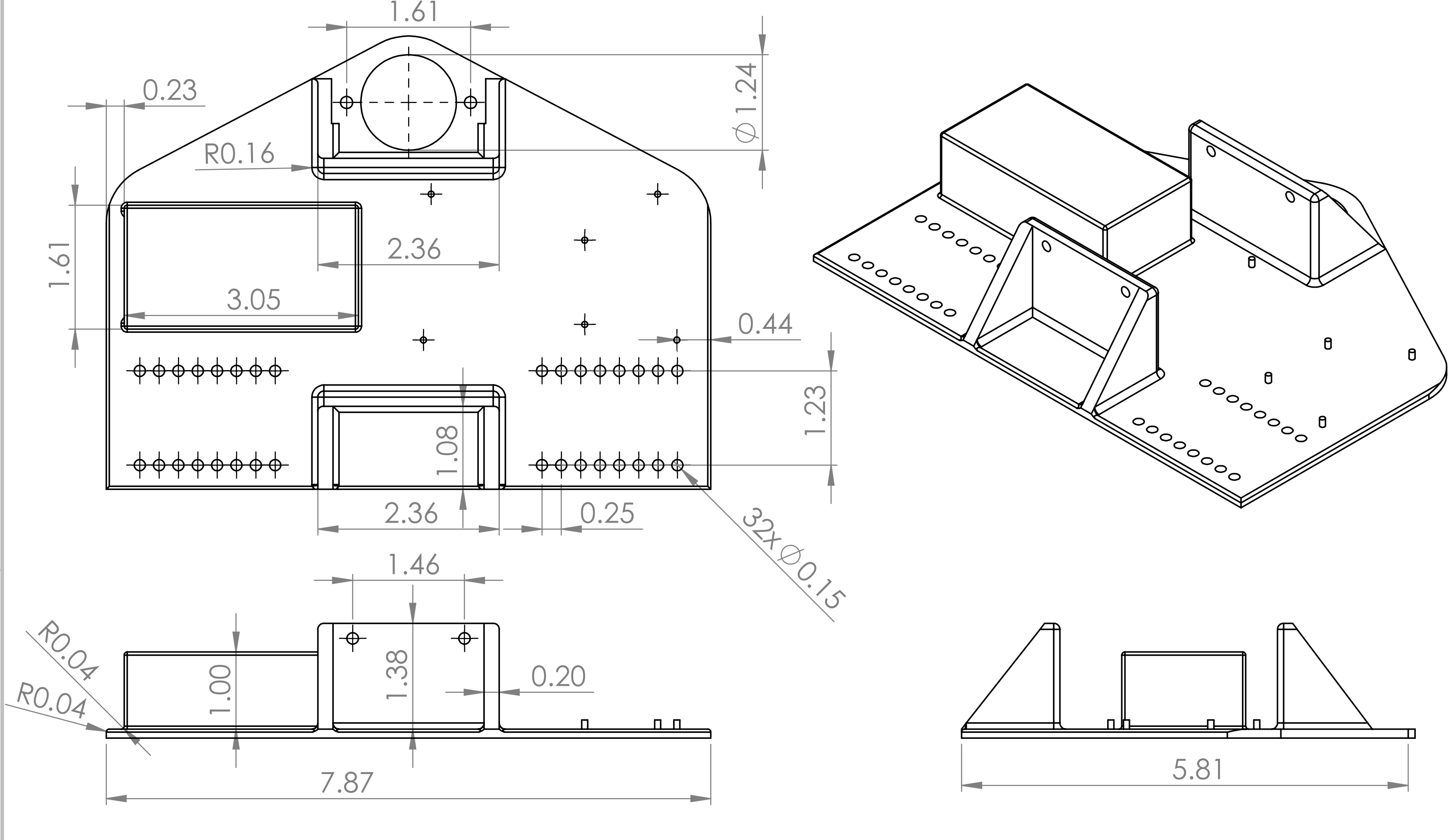
Zongyao Jin

I worked with the whole team to develop the project objectives. I then confirmed the model dynamics with Zongyao. My greatest contributions were made in developing the final software architecture. I took Zongyao's wrapper functions, filter, and initial controller, and implemented Mitch's pole placement control method. In doing this, I made the control robust to changes in the sampling rate and desired closed loop response by performing the necessary calculations in the setup portion of the code. I ensured that the controller would execute on a set time-interval by executing the control portion in timer-attached interrupt service routine. I performed the tests necessary to select an appropriate sample rate, and I implemented potentiometers for selecting the desired response.

Austin Rogers

B. *Engineering Drawings*

A dimensioned drawing of the 3D printed robot chassis can be found on the following page.



PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

NEXT ASSY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±

INTERPRET GEOMETRIC TOLERANCING PER:

MATERIAL

FINISH

DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	MA	11/7/16
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

TITLE:

SIZE	DWG. NO.	REV
A	chassis	

SCALE: 1:2 WEIGHT: SHEET 1 OF 1

C. Matching Controller Terms

Matching terms in $a_d(s)$, Eq. 18, and $a(s)$, Eq. 17 gives:

$$\theta = M \tag{22}$$

$$\tag{23}$$

$$\theta(\gamma + \alpha + \beta) = C + AM \tag{24}$$

$$\tag{25}$$

$$\theta(\phi + \alpha\beta + \gamma(\alpha + \beta)) = AC + BM \tag{26}$$

$$\tag{27}$$

$$\theta(\phi(\alpha + \beta) + \alpha\beta\gamma) = BC \tag{28}$$

$$\tag{29}$$

$$\theta\alpha\beta\phi = K \tag{30}$$

Rearranging the above equations yields a system of equations:

$$\begin{bmatrix} 1 & 0 & -1 & 0 \\ \frac{C}{M} & 1 & -(\alpha + \gamma) & 0 \\ 0 & \frac{C}{M} & -(\alpha\gamma + \phi) & 0 \\ 0 & 0 & \alpha\phi & -\frac{1}{M} \end{bmatrix} \begin{bmatrix} A \\ B \\ \beta \\ K \end{bmatrix} = \begin{bmatrix} -\frac{C}{M} + \gamma\alpha \\ \phi + \alpha\gamma \\ \phi\alpha \\ 0 \end{bmatrix}$$

which can be solved for controller parameters k, A, and B, and closed-loop pole β. In order for the closed-loop system behavior to be approximately second order, the inequality:

$$\beta > \alpha = 10\omega_n$$

must be valid; i.e., the real closed-loop poles are both significantly faster than the complex pair. This inequality was valid for all trials.

References

[1] K. Ogata, *Discrete-time control systems. 2nd ed. Katsuhiko Ogata.* Patpargani, Delhi, India : Pearson Education, 2002, 2002.

[2] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback control of dynamic systems. 5th ed. Gene F. Franklin, J. David Powell, Abbas Emami-Naeini.* Upper Saddle River, N.J. : Pearson Prentice Hall, [2006], 2006.