DP1: Derivation and Execution of Cat-Bot Controller

Veranga, Joshua * and Young, Ella †
AE 353 - Design Project 1

The following report presents the development and implementation of a state-feedback controller for a wheeled robot, referred to as the "Cat-Bot." Using a differential-drive mechanism, the robot is stabilized and allowed to maneuver along an axis toward the predicted landing positions of cat pilots. To achieve this, a state-space model was developed along with subsequent simulations in PyBullet to validate the controller's effectiveness [1]. The correct values for the controller were determined through trial and error. In the final simulation, three cats were launched and all three were caught successfully.

I. Nomenclature

r_w	=	Wheel radius, 0.325m
m_w	=	Wheel mass, 2.4kg
J_w	=	Wheel moment of inertia, $0.12675\text{kg} \cdot \text{m}^2$
r_b	=	Body radius, 0.3m
m_b	=	Body mass, 12.0kg
J_b	=	Body moment of inertia, $0.8\text{kg} \cdot \text{m}^2$
g	=	Gravity, $9.81 \frac{\text{m}}{\text{s}^2}$
ζ	=	Wheel position, m
θ	=	Pitch angle, rad
$\dot{\square}, \ddot{\square}$	=	First and second derivatives with respect to time, $\frac{\text{d}}{\text{d}s}$, $\frac{\text{d}^2}{\text{d}s^2}$
τ	=	Wheel torque, $\text{N} \cdot \text{m}$
f, x, u	=	Variable matrices
A, B, K, S	=	Coefficient matrices

II. Introduction

Differential drive transmission controls the movement of a vehicle by varying the torques applied to each wheel. This transmission is favored in producing mobile robots of various sizes and specializations due to its lack of complexity. Advancements are being created regarding what these vehicles can do and where they can go. The biggest drawback to differential drive is the struggle with traversing rough and changing environments. The wheels can be prone to slipping, impeding precise movements. In aerospace engineering, discovering ways to improve terrain adaptation is important for future space exploration.

A. Design Project Objective

This project will focus on the design and implementation of a controller that will adjust the torque applied evenly to both wheels of the vehicle to reach a random target for each pilot on the platform. When testing the robot, the governing equations will display the wheel position and velocity, the pitch angle, and the pitch rate. The chosen equilibrium point and feedback gain matrix will keep the chassis upright when catching the pilots.

B. Engineering Challenges

The modeling process will require addressing engineering challenges:

- Linearize the model in the space-state form about an equilibrium point concerning the desired wheel position.

*veranga2
†ellamy2

- Design in theory and implement an asymptotically stable linear state feedback controller in simulation.
- Simulate the final launch system, making sure it has a high probability of catching the cat- pilots.

C. Significance

Ultimately, the goal of the model is to guarantee the safety of the cat- pilots to the highest degree. The process of creating the launch system utilizes control methods essential to keeping people safe.

III. Theory

A. Equations of Motion

The motion of the C at-Bot is governed by the ordinary differential equation [1] expressed as the following:

$$M(q)\ddot{q} + N(q, \dot{q}) = F(q)r. \tag{1}$$

Where given matrices for terms [1] can be substituted in to yield the expanded matrix form

$$\begin{bmatrix} \frac{J_w}{r_w^2} + m_b + m_w & m_b r_b \cos(\theta) \\ m_b r_b \cos(\theta) & J_b + m_b r_b^2 \end{bmatrix} \begin{bmatrix} \ddot{\zeta} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} -m_b r_b \sin(\theta)\dot{\theta}^2 \\ -g m_b r_b \sin(\theta) \end{bmatrix} = \begin{bmatrix} \frac{\tau}{r_w} \\ -\tau \end{bmatrix} r. \tag{2}$$

While representing the full nonlinear dynamics of the system, the expression can be modified to solve for the acceleration of the system by isolating the respective term. Thus, obtaining the following expression for $\ddot{q}$:

$$\ddot{q} = M^{-1}(q)\left(F(q)r - N(q, \dot{q})\right). \tag{3}$$

Additionally, we define the state-space representation to provide a mathematical framework for the dynamical system as the following matrix:

$$f = \begin{bmatrix} \dot{\zeta} & \dot{\theta} & \ddot{\zeta} & \ddot{\theta} \end{bmatrix}^T. \tag{4}$$

B. Linearize Around Equilibrium

To linearize the system, we define a constant equilibrium point with desired conditions denoted as: $\zeta_e, \theta_e, \dot{\zeta}_e, \dot{\theta}_e$, and τ_e. Additionally, we define expressions for the perturbations around the respective equilibrium:

$$x = m - m_e \quad u = n - n_e, \tag{5}$$

where m and n are defined as the respective matrices following the structure of the equilibrium points.

$$m = \begin{bmatrix} \zeta & \theta & \dot{\zeta} & \dot{\theta} \end{bmatrix}^T \quad n = \begin{bmatrix} \tau \end{bmatrix}. \tag{6}$$

The system is then defined as a linear equation of motion:

$$\dot{x} = Ax + Bu. \tag{7}$$

Where A and B are coefficient matrices that are calculated by taking the Jacobian of the state-space relation Eq.(4) concerning the expressions of perturbation Eq.(5):

$$A = \left.\frac{\partial f}{\partial x}\right|_{m_e, n_e} \quad B = \left.\frac{\partial f}{\partial u}\right|_{m_e, n_e}. \tag{8}$$

C. State Feedback Control

Commonly, state feedback controllers are designed using the following expression for linear state feedback:

$$u = -Kx. \tag{9}$$

Here, K is defined as a gain matrix to introduce stability to the controller, which is determined using a method utilizing Linear Quadratic Regulators (LQR)[2]. Therefore, substituting in the expression for u from Eq.(9) allows for the definition of a closed-loop system from the linearized equation of motion Eq.(7):

$$\dot{x} = (A - BK)\, x. \tag{10}$$

Where, subsequently, we can determine whether the system is stable or not by calculating the eigenvalues of the system matrix:

$$s = \lambda(A - BK). \tag{11}$$

D. Obtaining Control Values

To obtain a working controller for our system, an equilibrium point must be defined and subsequently used to derive the gain matrix. Doing this, the equilibrium point is defined with $\{\zeta_e = 1, \theta_e = 0, \dot{\zeta}_e = 0, \dot{\theta}_e = 0, \tau_e = 0\}$, stating the equilibrium point at the center of the plate, standing straight up, not moving or rotating, and not inducing any torque. This means the Cat-Bot is at the desired position, standing upright, and not moving. This is the best choice of equilibrium point because the pilot's target is randomized, starting at rest in the middle of the platform at a neutral angle is the most optimal to catch the first cat. Substituting into Eq.(6) yields

$$m_e = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \quad n_e = \begin{bmatrix} 0 \end{bmatrix} \tag{12}$$

which can then be used with Eq.(8) to determine the A and B matrices:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -7.77 & 0 & 0 \\ 0 & 33.67 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0.57 \\ -1.63 \end{bmatrix}. \tag{13}$$

As expressed in subsection III.C. State Feedback Control, the above matrices A and B are then used to solve for the gain matrix. The process enacted to do so is using the two matrices to solve an algebraic Riccati equation as expressed in LQR solutions [2]. The aforementioned process is can be found in an isolated file [3] expressing the code utilized to obtain the gain matrix:

$$K = \begin{bmatrix} -1, -46.56107063, -3.7334729111, -8.909737686 \end{bmatrix}, \tag{14}$$

which is then used to create the controller.

IV. Experimental Methods

To develop the Cat-Bot controller, the gain matrix is implemented in the respective code cell provided in the design project manual [1] by denoting the matrix multiplication seen in Eq.(9) and returning the control matrix, u, consisting of the necessary torque to approach or remain in the desired equilibrium. Where the desired equilibrium is one that changes due to the definition of the dynamic reference state:

$$\begin{bmatrix} \text{cat\_target} & 0 & 0 & 0 \end{bmatrix}, \tag{15}$$

allowing the Cat-Bot to move to the target position and successfully catch the cat pilots.

The simulation was then run for multiple trials under the condition of a duration of 18.99 seconds, as stated in the provided notebook [1] to catch three cat pilots and the initial conditions specified in Eq.(6); to successfully catch all three cat pilots. However, in the initial stages of running the Cat-Bot, it was evident that the calculated gain matrix was yielding poor results: not moving to target positions fast enough and/or falling over too easily after catching or colliding with a cat pilot. Therefore, further trials, with the same initial conditions and controller, were enacted with the only variation being the addition of a multiplied matrix to scale the values of the gain matrix to different values. The process of changing the scaling matrix and running the simulation was conducted until the desired outcome occurred: catching three consecutively launched cat pilots three times in a row.

After 76 trials of changing the scale matrix the desired outcome was obtained with a final scale matrix of

$$\begin{bmatrix} 10 & 4 & 9 & 10 \end{bmatrix}, \tag{16}$$

yielding a final gain matrix of:

$$\begin{bmatrix} -10 & -186.24428252 & -33.6012562 & -89.09737686 \end{bmatrix}. \tag{17}$$

V. Results and Discussion

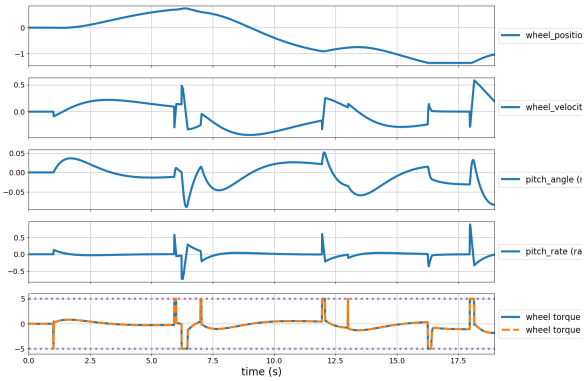

Fig. 1 Variable changes in successful launch.

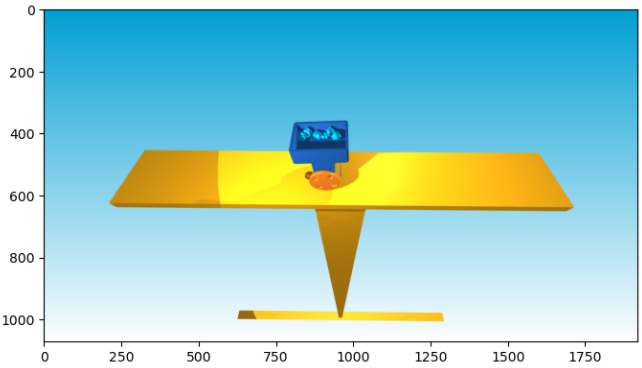

Fig. 2 Image of chassis and cat pilots after 18.99 seconds.

The testing phase involved 76 trials, each lasting 18.99 seconds, during which the Cat-Bot aimed to catch three cat pilots without tipping over or losing any passengers. The controller was designed to achieve this by monitoring the bot's motion and restoring equilibrium after each impact. In fig. 1 there are visible changes in all 5 charts which correspond to the collisions between the cat pilots and the chassis which occur every 6.3 seconds.

A. Position and Velocity

Specifically looking at the position graph (1), the slope of the line is the greatest right when a target is placed on the platform. The line begins to level out as the Cat-Bot approaches the target point, reaching the location at the right time to catch the pilot, while also staying upright. The controller keeps the chassis stable throughout the launches due to this gradual change in velocity. During initial adjustment trials, before the implementation of this feature, the torque applied to the cart would cause it to accelerate at a rate too substantial to maintain an upright position during the collision. The high velocity and heavy collision caused the chassis to tip over when the target was far from its initial position. The velocity portion of the chart reflects the findings from the position portion.

B. Pitch Angle and Pitch Rate

The same sort of relationship is shown in the pitch angle and pitch rate (1). In response to the forward movement, the chassis tips forward to not fall backward as the velocity increases. In addition to this, the chassis tips the most during the collisions themselves. The highest slope for the angle and the highest value for the pitch rate occur immediately after the collisions; The pilot is moving so fast as to shake the bot.

VI. Conclusion

The controller was designed through multiple trials and errors. The biggest obstacle to overcome in the code was to find a way to keep the chassis upright after collisions with the cat- pilots. This issue arises from the stability of the eigenvalues and the chosen gain matrix. After multiple attempts scaling values and watching simulations [3], the system began functioning correctly to catch three pilots simultaneously, and even multiple times in a row. Even so, the system

is not consistent. During the development of the controller, it was noted that the given moment of inertia was high compared to the maximum limit for torque. This caused difficulties in keeping the chassis upright, especially when the cats were landing. The overall system works but is not perfect enough to catch every pilot with certainty. To achieve this level of accuracy the controller must be iterated more to find the perfect values, displaying the importance of proper controllers when it comes to public safety.

Appendix

Throughout the test simulations and even the final launch, there were many instances where the cat pilots suffered casualties. These events occurred mainly due to the high speed of the contact between the pilot and the chassis of the cart, and the randomized nature and location of the cat-target. During the testing, the cat would make contact with the chassis of the cart, but then, many times, would bounce out depending on where the target ended up on the platform. Overall, the system is not consistent enough to be considered safe for the pilots since there is low predictability.

Acknowledgments

We would like to sincerely express our thanks to Dmitriy Shvydkoy, for his unique and invaluable contribution to this project. As both a partner to one team member and a roommate to the other, his presence in our lives fostered a relationship in a collaborative and supportive environment, overall improving the efficacy of this project in addition to keeping us motivated throughout this endeavor.

References

[1] Bretl, T., "Design Project 1 (wheeled cat-catching robot)," February 13, 2025. URL https://tbretl.github.io/ae353-sp25/projects/01-catbot.html.

[2] How, J., and Frazzoli, E., "Deterministic LQR," 2010. URL https://ocw.mit.edu/courses/16-30-feedback-control-systems-fall-2010/resources/mit16_30f10_lec18/, lecture Notes - MIT16_30F10_lec18.pdf.

[3] Veranga, J., "SP25_AE353_Code/Design_Project_1," February 14, 2025. URL https://github.com/muraSHUki/SP25_AE353_Code.git.