AIAA SciTech Forum
January 3-7, 2022, San Diego, CA & Virtual
AIAA SCITECH 2022 Forum

10.2514/6.2022-1914

BeagleRover: An Open-Source 3D-Printable Robotic Platform for Engineering Education and Research

Pengcheng Cao*, James Strawson†, Xuebin Zhu‡, Everbrook Zhou§, Chase Lazar¶, Dominique Meyer‖, Zhaoliang Zheng**, Thomas Bewley ††, and Falko Kuester ‡‡
University of California San Diego, La Jolla, CA 92093, USA

Recent years have witnessed the rapid development of robotic platforms specialized in engineering education. And more robotic courses are being designed and taught in the curricula of different school levels and college programs. Among a number of the educational mobile robotic platforms, however, one can rarely find a small-sized, 4-wheel-driven mobile robot which can be considered as a chassis design for extraterrestrial or off-road rovers. In addition, it could difficult for the learners to understand the circuits configuration and electrical actuation of embedded systems from an off-the-shelf robotic platform. In this paper, we propose a low-cost open-source robotic platform named BeagleRover. BeagleRover is a 4-wheel-driven mobile robot of the size $236 \times 223 \times 85mm$ ($L \times W \times H$). It includes 4 DC servo motors and 4 DC gear motors to over-actuate its motion in order to realize more precise path-tracking on uneven terrains. Most of its components are 3D-printable using hobbyist printers or can be procured at relatively low costs from hobbyist websites. The controller board which BeagleRover uses is BeagleBone® Blue by default but can be replaced by some other SoC computers due to BeagleRover's multiple sets of mounting holes. In addition, We utilized BeagleRovers during the MAE 40 and MAE 144 classes at UC San Diego to help instruct electrical circuit fundamentals and classical control theories, respectively. Next, code-based and Simulink-based software approaches are discussed for users to choose and start with for their own projects. Last but not least, we explore the feasibility of using BeagleRover as a research platform for indoor navigation and SLAM and object detection studies.

I. Introduction

Micro robot platforms have been involved in pre-college and college STEM educations since 1960s. In 1969, Seymour Papert first developed the idea of using Logo programming language and Turtle robots to teach mathematics [1] to pupils. During the last 50 years, the development of educational robots has been boosted by both the advancements in technologies and increasing market demands from both parents and educational institutes. Some of the most successful educational robots include Lego Mindstorms EV3 [2] to teach robotic motions and programming, humanoid robot NAO [3] to interact with children, and MakeBlock mBot [4] as one of the low-cost STEM robots. For students during their college education in engineering, the Renaissance Robotics and UCSD Coordinates Robotics Lab have developed the EduMIP robot [5] which is a 2-wheel mobile inverted pendulum driven by a BeagleBone Blue SoC computer[6].

Apart from educational robotics, 3D printing has also become an emerging area with both fast progressing research and applications. For robotics, 3D printing is implemented mainly for novel design and fast prototyping. In educational robotics, de Souza and Elisiário [7] designed multiple 3D-printable robotic projects to educate teenagers in STEM disciplines. In addition, Gonzalez et al. [8] and Lapeyre et al. [9] studied 3D-printable mobile and humanoid robots for

*Ph.D. candidate in Mechanical Engineering, University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92093; p5cao@eng.ucsd.edu. AIAA student member
†Robotics Engineer, ModalAI, Inc., 10855 Sorrento Valley Rd Ste. 2, San Diego, CA 92121; jstrawso@gmail.com
‡Undergraduate student in Mechanical Engineering, University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92093; xuz157@ucsd.edu.
§Undergraduate student in Mechanical Engineering, University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92093; e8zhou@ucsd.edu.
¶Undergraduate student in Aerospace Engineering, University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92093; cjlazar@ucsd.edu.
‖Postdoctoral researcher in Computer Science and Engineering, University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92093; demeyer@eng.ucsd.edu.
**Ph.D. student in Electrical and Computer Engineering, University of California, Los Angeles, Los Angeles, CA 90095; zhz03@g.ucla.edu.
††Professor, Department of Mechanical and Aerospace Engineering, University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92093; bewley@eng.ucsd.edu.
‡‡Professor, Department of Structural Engineering, University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92093; fkuester@eng.ucsd.edu.

educational uses, respectively. In soft robotics, Gul el al. [10] implemented fused deposition modeling (FDM) based 3D printing and post-processing to manufacture a cylindrical actuator as a steerable catheter. In aerial robots, Strawson et al. [11] [12] and De Vivo et al. [13] attempted on implementing large-scale 3D printers to manufacture aerodynamically functional components with structural and mechanical optimization for multirotor and fixed-wing unmanned aerial vehicles (UAVs), respectively.

Among a number of the educational mobile robots or 3-D printable robots, however, one can rarely find a small-sized, 4-wheel-driven mobile robot which can be considered as a base design for extraterrestrial or off-road rovers. And one can hardly find the literature deploying mobile robots with complex kinematics design fro educational and research uses. In this case, this paper introduces the robotic platform BeagleRover designed and tested through years of collaborations between researchers of different levels (undergraduates, graduates, post-docs and faculties) in CHEI Lab and Coordinated Robotics Lab at University of California, San Diego. This platform is mainly aimed at serving educational purposes, while it also plays a role in several research projects. This project is currently open-sourced and published on GitHub at `https://github.com/p5cao/BeagleRover`.

The structure of this paper is as follows: in Section II we discuss the design motivations and workflow of BeagleRover; in Section III we derive the kinematics equations for both 4-wheel steering and 2-wheel-driving with self-balancing of the robot; in Section IV we talk about two primary software approaches for users to drive the robot and start their own projects; in Section V we discuss the educational uses of BeagleRover in college curricula at UC San Diego; and in Section VI, we introduce the research deployment of BeagleRover.

II. Design Workflow

A. Design Motivation

The concept of BeagleRover was first inspired by the double Ackermann steering configuration which enables 4-wheel steering and proves to provide better path-tracking ability on rough terrains compared to front-wheels-steering [14], and a similar design of an active suspension and independent steering can also be found on the extraterrestrial rover VIPER by NASA [15]. As shown in Fig. 1, the double Ackermann steering significantly reduces the minimum turn radius compared to the front wheel steering on the same vehicle. Second, another consideration for this design is to enable the BeagleRover to balance itself in the upright pose with 2 wheels and drive on both top and bottom sides with 4 wheels in the event of rollover like a RC stunt car [16]. Third, from educational perspective, it is intended to become a practical learning tool for UCSD engineering students to better understand embedded systems and control of a robotic vehicle. Last but not least, there has also been a need in the UCSD CHEI Lab to develop a small-sized robotic rover providing high agility and maneuverability to travel across the uneven surfaces on excavation sites. The aforementioned motivations have been driving our team of student researchers to improve the design of BeagleRover to the current version. In this section, we introduce and discuss the design phases, double Ackermann steering, two-wheeled inverted pendulum, chassis design, and layered manufacturing of BeagleRover.

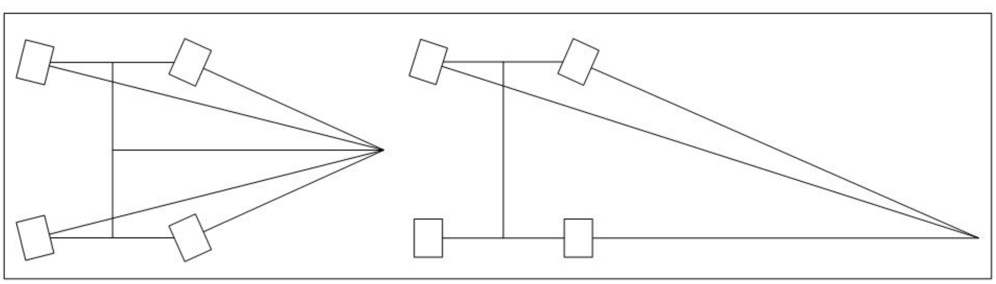


Fig. 1 4-wheel and front-wheel steerings.

B. Design Phases

The entire design process starts from determining the dimensions and parameters of the BeagleRover, including wheel and body masses and dimensions, total mass and dimension, and moments of inertia of interest. The list of dimensions and parameters are given in Table 1. After that, we select the electrical actuators for the BeagleRover.

Table 2 enlists the parameters of four motors to drive the wheels and four servo motors to actuate the steering. We also select the 7.4V 2-cell LiPo battery as power supply of the robot and BeagleBone Blue as its controller board [6]. Meanwhile, we build the each component of 3D-printable wheels and steering mechanism in SolidWorks according to the components and actuators we select. Next, we design the chassis which mounts and supports all of above components. This chassis will also have mounting holes to fit with some other off-the-shelf controller boards. Both wheels and steering design and chassis design will be discussed in the following subsections.

After finishing chassis design, all the components either come from additive manufacturing or from procurement. We then assemble and test the BeagleRover, and design and implement the BeagleBone or Jetson Nano-based software for different uses. The whole design workflow is illustrated as in Fig. 2.

Table 1 BeagleRover dimensions and parameters

Parameter	Value	Unit	Explanation
g	9.810	m/s^2	Gravitational acceleration
m_w	0.089	kg	Wheel assembly mass
D_w	0.085	m	Wheel diameter
w_w	0.040	m	Wheel width
J_w	$mD^2/8 = 3.133 \times 10^{-5}$	kgm^2	Wheel moment of inertial
d_{cw}	0.089	m	Distance between vehicle center of mass and wheel base center
d_{pw}	0.007	m	Distance between steering pin (shoulder screw) and wheel base center
m_B	0.250	kg	Vehicle body block mass (chassis total + battery)
w_B	0.142	m	Vehicle body block width
l_B	0.183	m	Vehicle body block length
h_B	0.048	m	Vehicle body block height
M	0.607	kg	Vehicle total mass including battery (without camera)
W	0.223	m	Vehicle total width
L	0.236	m	Vehicle total length
H	0.085	m	Vehicle total height
J_ψ	$Md_{cw}^2/3 = 1.592 \times 10^{-3}$	kgm^2	Vehicle pitch moment of inertia
J_ϕ	$M(w_B^2 + h_B^2)/12 = 1.137 \times 10^{-3}$	kgm^2	Vehicle yaw moment of inertia

Table 2 Actuators parameters

Parameter	Value	Unit	Explanation
m_s	1.060×10^{-2}	kg	Servo motor mass
m_m	9.410×10^{-3}	kg	Wheel motor mass
T_m	0.176	Nm	Wheel motor maximum torque
J_m	2.767×10^{-7}	kgm^2	Wheel motor moment of inertia
I_m	1.6	A	Wheel motor working current
V_m	7.4	V	Wheel motor nominal voltage
R_m	2.7	Ω	Wheel motor resistance
K_b	7.407×10^{-4}	$V \cdot s/rad$	Wheel motor back EMF constant
K_t	$T_m/I_m = 0.110$	Nm/A	Wheel motor torque constant
K_g	100 : 1		Wheel motor gear ratio

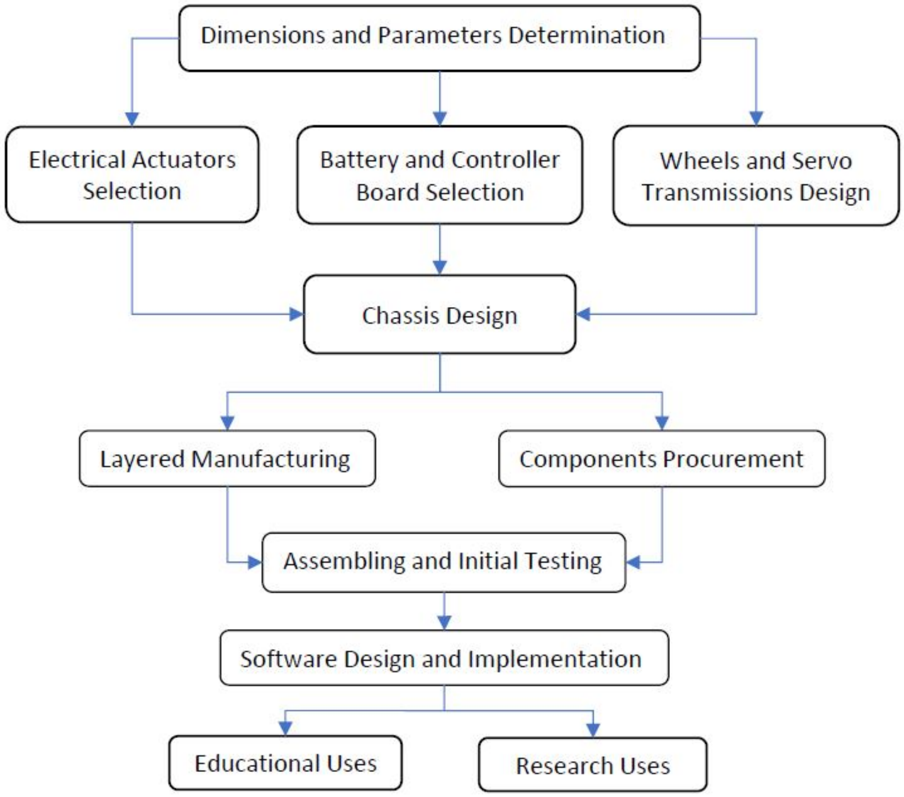


Fig. 2 The flowchart illustrating the design phases of BeagleRover

C. Wheels and Steering Design

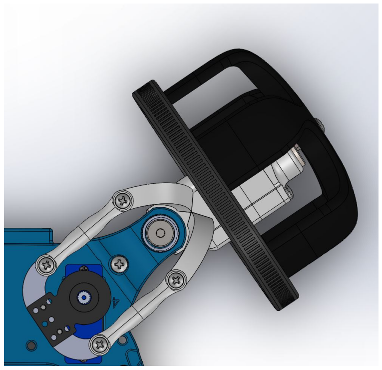
At each corner, the corner sub-assembly mounted on the chassis consists of a servo motor and a wheel motor with encoder, as shown in Fig. 3. The in-total 8 actuators on a BeagleRover vehicle control its 3 Degrees of Freedom when traveling on a 2-D plane, which results in over-actuation. With proper control allocation, over-actuation can help realize more precise path-tracking on uneven terrains [17]. For each servo, the black plastic part called servo saver is attached and fixed on to the servo output gear as can be seen at the lower left corner of Fig. 3. The use of servo saver here is to prevent the servo motor from being damaged by vibrations or impacts.

The wheel itself is of a quasi-cylindrical shape with $85mm$ diameter and $45mm$ width or height. The base of the wheel has an extra $8mm$ in radius and is wrapped by a rubber tire. The reason for this quasi-cylindrical design is to allow enough space to accommodate different motors with or without gearbox and rotary encoder attached to one side of the motor, and offer protections for these parts as well. The diameter of this wheel is greater than the body block height $h_B = 48mm$ on both top and bottom sides so that it can prevent damage to the electronics or servos.

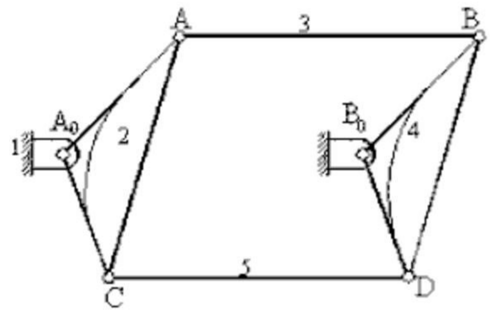
The steering of each wheel is realized via a 4-bar parallelogram mechanism as shown in Fig. 3. The 2 fixed pivot joints of this 4-bar linkage are the center of servo shaft and the center of stainless steel pin (shoulder screw), respectively. With the servo motor driving the triangular rigid body ΔB_0BD, the first triangle ΔA_0AC rotates around A_0, and $\overline{AC}$ bar and $\overline{BD}$ bar are parallel to each other.

D. Chassis Design

Once we selected electrical actuators, battery, controller board, and hardware, we design the chassis to connect and support these components after designing the corner sub-assembly. Fig. 4c illustrates the top view of the combined bottom, front, and back pieces of chassis. Considering the low-profile tool-path slicer and 3-D printers used in this study, we round off most of the sharp edges for better layered manufacturing effect and to avoid cutting the users in future deployment.

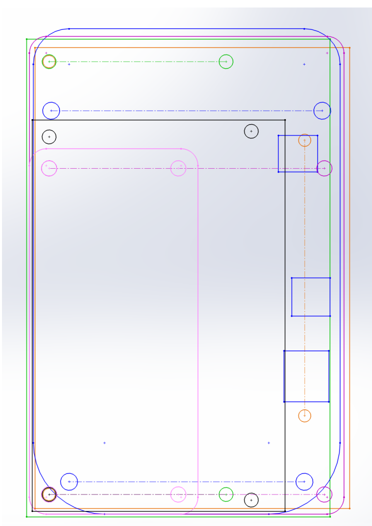


(a) Corner sub-assembly in Solidworks.

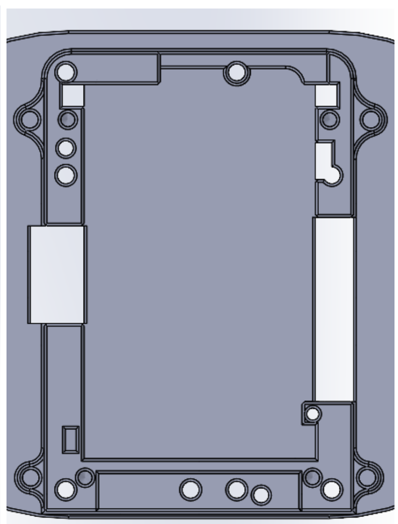


(b) Kinetics diagram of the steering mechanism [18].

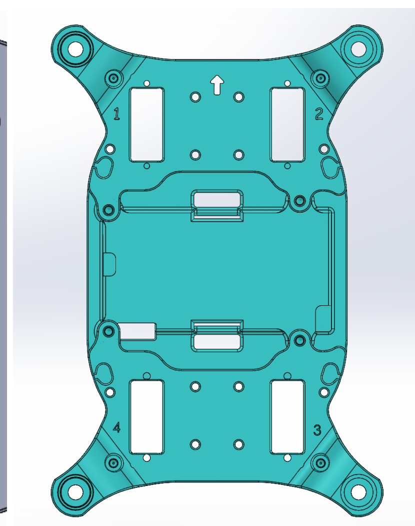
Fig. 3 Corner sub-assembly and its steering mechanism.



(a) Mounting holes sketch.



(b) Mounting holes on chassis.



(c) Top view of the whole chassis.

Fig. 4 Chassis Designs

In addition, we design 6 sets of mounting holes on the chassis bottom, as illustrated in Fig. 4. We implement this design in order to adapt to a number of System-on-a-Chip (SoC) boards for users to carry out their own projects on top of the uses of BeagleRover in this study. These SoC systems include but not limited to Raspberry Pi, Raspberry Pi Zero, Jetson Nano, and BeagleBone Blue.

E. Additive Manufacturing

As discussed above, the majority of the BeagleRover components are produced using additive manufacturing. In fact, excluding cables and hardware, 37 out of 54 parts are manufactured in PETG filament by a hobbyist-level Ultimaker 3 desktop printer and they weigh only 267.36 grams in total, accounting for 44% of the total weight. Printing these parts takes 44.3 hours in total. An assembled BeagleRover prototype with camera is displayed in Fig. 5.

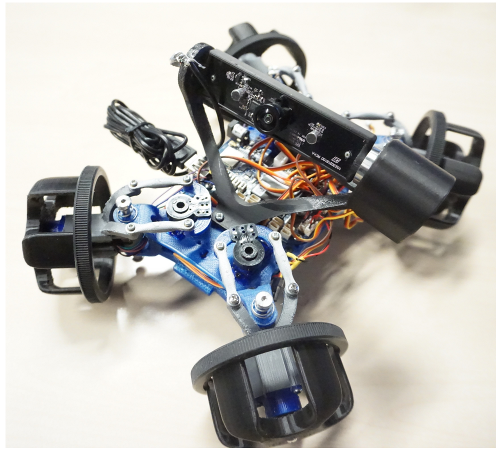
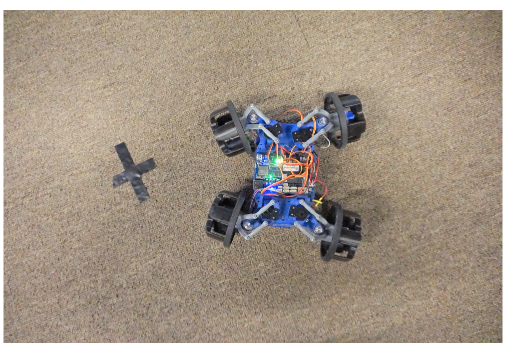


Fig. 5 A 3D-printed BeagleRover prototype

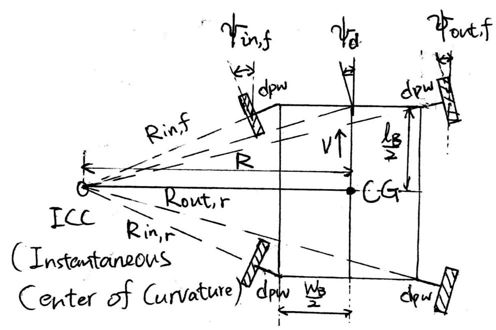
III. Kinematics and Control

A. Four-Wheel Steering

The BeagleRover mobile platform is a non-holonimic or over-actuated car-like vehicle with four-wheel steering (4WS). This type of steering mechanism can also be categorized as double Ackermann steering [19] if both front and rear wheels are actuated but steered to different angles. In general, 4WS improves the maneuverability of the vehicle by allowing sharper turns due to requiring a smaller admissible turning radius than does 2-wheel steering. In addition, since the BeagleRover does not have linkage mechanism for steering but utilize servo motors to steer each wheel individually, we can produce a versatile set of steering modes as shown in Fig. 9 providing the BeagleRover with higher agility.



(a) BeagleRover steering.



(b) Schematics of steering.

Fig. 6 Double Ackermann steering of BeagleRover

Next, we perform the inverse kinematics to compute the desired steering angle ψ_i^d and motor rotary speed n_i^d of i-th wheel. Our implementation of double Ackermann steering is design for a symmetric 4WS configuration, meaning the steer angles of front and rear wheel pairs are mirrored as shown in Fig. 6b [20]. Therefore, We first approximate this kinematics using a two-bicycle model[21], where both front and rear pairs of wheels are reduced to single imaginary front middle wheel and rear middle wheel, respectively. Next, according to the path it is following, we compute the turn radius R from the length of the vehicle body and desired steer angle δ_d if imaginary middle wheel by

$$R = \frac{l_B}{2\tan\psi_d} \tag{1}$$

and find its instant center of rotation (ICR). Next, from the desired velocity, we compute the instant angular velocity ω

of imaginary front and rear middle wheels with respect to ICR by

$$\omega = \frac{V_d}{R}. \tag{2}$$

Next, we compute the turn radii of inner wheels R_{in} and outer wheels R_{out}. Since front and rear wheels are mirrored to each other by the vertical bisector of the body, we have

$$\begin{aligned} R_{in,f} = R_{in,r} &= \sqrt{(R - w_B/2)^2 + (l_B/2)^2} \\ R_{out,f} = R_{out,r} &= \sqrt{(R + w_B/2)^2 + (l_B/2)^2} \end{aligned} \tag{3}$$

to compute the turn radius for each wheel. Due to the symmetry, we now only need to compute the wheel velocities and steer angles of the front 2 wheels. The equations are

$$V_{in,f} = \omega \cdot (R_{in,f} - d_{pw}),\ V_{out,f} = \omega \cdot (R_{out,f} - d_{pw}). \tag{4}$$

where $V_{in,f}, V_{out,f}$ are the wheel velocities of inner and outer front wheels, respectively. And the front steering angles are

$$\psi_{in,f} = \tan^{-1}\Big(\frac{\frac{l_B}{2}}{R - \frac{w_B}{2}}\Big),\ \psi_{out,f} = \tan^{-1}\Big(\frac{\frac{l_B}{2}}{R + \frac{w_B}{2}}\Big). \tag{5}$$

for inner and outer front wheels, respectively. In addition of the steering angles, we can also compute the rotary speed of the motor shaft from the wheel speed knowing that we have a gearbox to transfer the motor speed to gearbox shaft speed (wheel speed) by

$$n_{w,i} = \frac{2 * V_i / D_w}{2\pi},\ n_{m,i} = K_g n_{w,i} \tag{6}$$

where $n_{w,i}$ is the wheel rotary speed driven by i-th motor, V_i is linear speed of this wheel, and $n_{m,i}$ is the motor shaft speed.

B. Inverted Pendulum and 2-Wheel Driving

In order to protect the BeagleRover from damaging in the event of rollover, we design the mobile inverted pendulum mode where BeagleRover can be driven on two rear wheels. In addition, the balancing task for a mobile inverted pendulum requires detailed studies of its equations of motion, which is often considered as a good project or assignment for students learning dynamic systems and controls. Indeed, we assigned this task as the final project of a course as mentioned in Section 5.

For a BeagleRover in its inverted pendulum mode, we derive its equations of motion based on Lagrangian methods as follows:

$$\begin{aligned} ((2m + M)R^2 + 2J_w + 2n^2 J_m)\ddot{\theta} + (MLR\cos\psi - 2n^2 J_m)\ddot{\psi} - MLR\sin\psi\dot{\psi}^2 \\ -((2m + M)R^2\theta + MLR\sin\psi)\dot{\phi}^2 = F_\theta, \end{aligned} \tag{7a}$$

$$\begin{aligned} \Big(\frac{mW^2}{2} + \frac{W^2}{2R^2}(J_w + n^2 J_m) + J_\phi + 2MLR\theta\sin\psi + (2m + M)R^2\theta^2\Big)\ddot{\phi} + 2(2m + M)R^2\theta\dot{\theta}\dot{\phi} \\ +2ML^2\sin\psi\cos\psi\dot{\psi}\dot{\phi} + 2MLR\dot{\phi}(\dot{\theta}\sin\psi + \cos\psi\theta\dot{\psi} = F_\phi, \end{aligned} \tag{7b}$$

$$(mL^2 + J_\psi + 2n^2 J_m)\ddot{\psi} + (-2n^2 J_m + MLR\cos\psi)\ddot{\theta} - MgL\sin\psi - (MLR\theta + ML^2\sin\psi)\dot{\phi}^2\cos\psi = F_\psi, \tag{7c}$$

where $R = D_w/2$ is the wheel radius, $m = m_w$ is the mass of the wheel, θ is the average rotational angle of left and right wheel, ψ is the body pitch angle, and ϕ is the body yaw angle. The equations of motion are then linearized around the equilibrium point, when the body angle ψ changes at about ±5°. We linearize Eqn. 7 as follows, with ψ approximately being 0, meaning that $\sin\psi \simeq 0$, $\cos\psi \simeq 1$, and $\dot{\psi}^2 \simeq 0$.

$$((2m + M)R^2 + 2J_w + 2n^2 J_m)\ddot{\theta} + (MLR\cos\psi - 2n^2 J_m)\ddot{\psi} - MLR\sin\psi\dot{\psi}^2 = F_\theta, \tag{8a}$$

$$(\frac{mW^2}{2} + \frac{W^2}{2R^2}(J_w + n^2 J_m) + J_\phi)\ddot{\phi} = F_\phi, \tag{8b}$$

$$(mL^2 + J_\psi + 2n^2 J_m)\ddot{\psi} + (-2n^2 J_m + MLR)\ddot{\theta} - MgL\psi = F_\psi. \tag{8c}$$

Then we design the state vector $x_1 = [\theta \;\; \psi \;\; \dot{\theta} \;\; \dot{\psi}]^T$, $x_1 = [\phi \;\; \dot{\phi}]^T$, and $u = [v_l \;\; v_r]^T$ as control input, where $v_l, \; v_r$ are the input voltage to motors of the rear left and right wheels, respectively. Thus, we can re-write Eqn. 8 as state space equations form:

$$\begin{aligned} \dot{x}_1 &= A_1 x_1 + B_1 u, \\ \dot{x}_2 &= A_2 x_2 + B_2 u, \end{aligned} \tag{9}$$

Then, an optimal controller is designed to stabilize the BeagleRover angles around equilibrium points using the linear-quadratic-regulator (LQR) method [22]. With the 2-wheel balancing realize, we perform the steering and driving with a simple differential driving control based on Fig. 7.

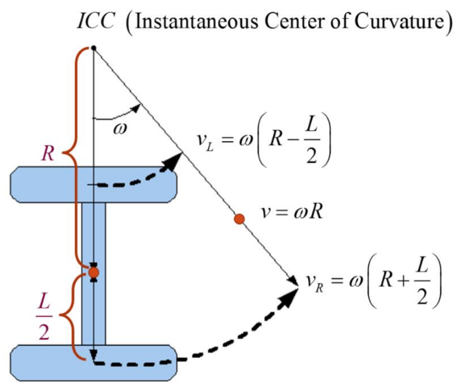


Fig. 7 2-wheel differential driving.

IV. Software Approaches

The software implementation for BeagleRover is designed according to the needs for its application for educational and research purposes. The two main approaches at this moment are code-based and Simulink based, and both code and Simulink interfaces allow users to develop purely on-board applications or control from a personal computer.

A. Code-Based Implementation

Table 3 lists the current available software libraries which are known to the authors. BoneScript is the JavaScript library developed and maintained by the beagleboard organization [6]. In addition, we know that the implementation of Robot Operating System (ROS) has been successful by researchers at the John Hopkins University for teaching a class using EduMIP [23]. The Robot Control Library by StrawsonDesign [5] is a C library with the most comprehensive APIs and great documentations for the users wishing to writing high-performance programs then compile directly it into the machine code. And the rcpy [24] is the python library developed by Dr. Mauricio C. de Oliveira at UC San Diego and is currently the most populer BeagleBone APIs. The BlackLib is a C++ library mainly design for BeagleBone Blue's previous version BeagleBone Black, but some C++ users still prefer to build software based on this library.

With basic programming skills and some knowledge of the BeagleBone embedded systems, users can choose their own code-based implementation for BeagleRover. From the authors' perspectives, the most reliable libraries are Robot Control Library and rcpy, with which the authors developed most of the applications in this paper.

Table 3 BeagleBone Blue Libraries

Name	Language	Community Support	Being Updated
BoneScript	JavaScript	Yes	Unknown
ROS	python or C++	Yes	No
Robot Control Library	C	Yes	No
rcpy	python	No	No
BlackLib	C++	No	No

B. Simulink Implementation

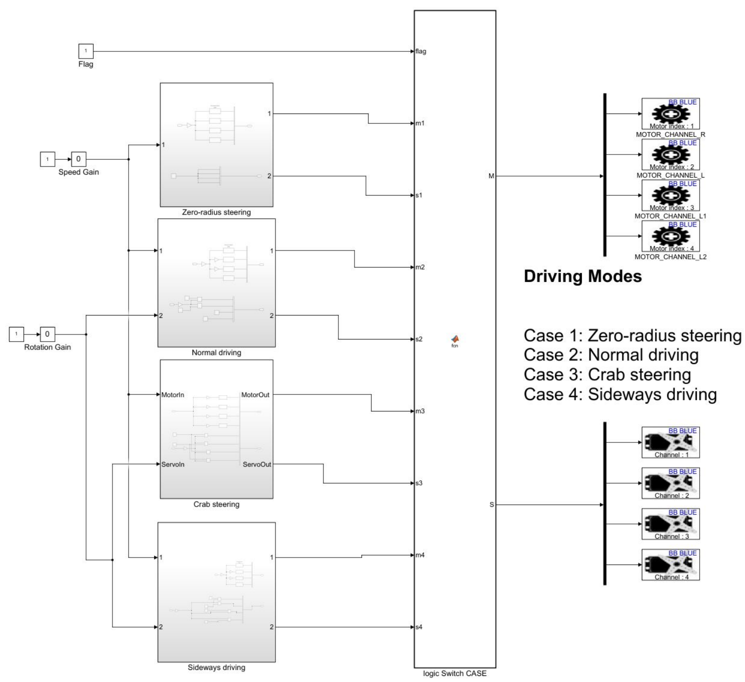


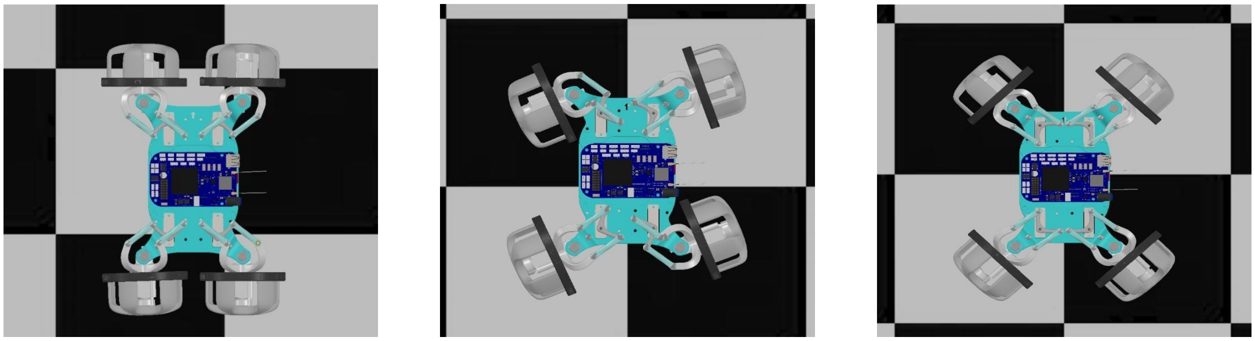
Fig. 8 Driving control interface for BeagleRover via Simulink

The authors also worked on the control interface software via MATLAB Simulink. These software implementations are made possible by the Simulink support package for BeagleBone Blue. As illustrated in Fig. 8, The user can select from 4 driving modes which are spinning mode (for the BeagleRover to start self-spinning around its CoM), quarter-turn turn, sidling mode, and normal car (driving) mode. These commands generated from Simulink are sent to BeagleBone Blue controller board to drive the servo motors and wheel motors using the pre-programmed blocks from Simulink Coder Support Package for BeagleBone Blue Hardware. Note that the authors have only achieved open-loop control for both servo motors and wheel motors at this moment.

Due to the existence of 4 servo motors, BeagleRover is able to perform at least 3 more driving modes in addition to the normal forward driving and Ackermann steering modes, which are spinning mode, sideways mode, and sidling mode, as illustrated in Fig. 9.

V. Educational Deployment

In this section, we introduce two course-based case studies with the students in the Department of Mechanical and Aerospace Engineering (MAE) at UC San Diego where we use the BeagleRover as practical learning tool. These two



(a) **Sideways mode** (b) **Crab steering mode** (c) **Zero-radius steering mode**

Fig. 9 3 additional driving modes due to over-actuation

courses are MAE 40: Linear Circuits and MAE 144: Embedded Control & Robotics. Next, we discuss the approaches and results of both courses in detail.

A. MAE 40: Linear Circuits

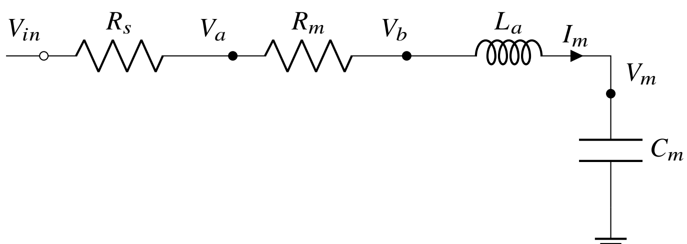


Fig. 10 A simplified linear circuit schematic of BeagleRover motor.

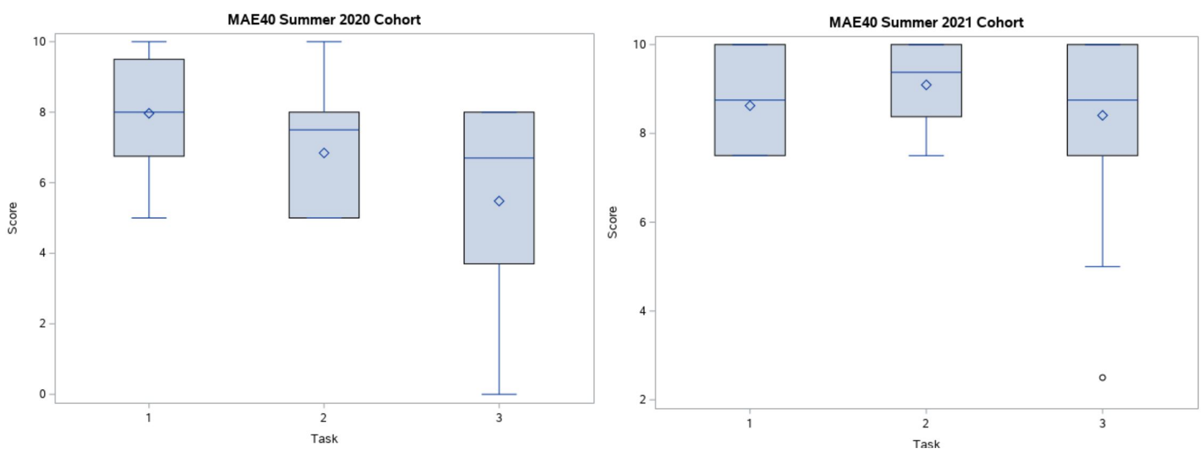
MAE 40 Linear Circuits is a lower-division undergraduate course offered by the Department of Mechanical and Aerospace Engineering (MAE) at UC San Diego. It aims at teaching the fundamentals of linear electrical circuits to MAE undergraduate students. The contents of this course include but do not limit to steady-state and dynamic behavior of linear, lumped-parameter electrical circuits, Kirchhoff's laws, RLC circuits, node and mesh analysis, operational amplifiers, signal acquisition and conditioning, electric motors, design applications in engineering and data sheet reading. During the last assignment (Homework 5) for both Summer 2020 and 2021 cohorts, we required students to work on 3 major tasks which involve transfer function modeling of a DC motor and analysis of a circuit according to Fig. 10 and the datesheets. The details of this assignment are enlisted in Table 4.

For the 2021 cohort, we organized and delivered a lecture using BeagleRover as an instrument to validate the results computed from example problems similar to the problems in Homework 5, and exemplified the uses of these computations in the design of robots and electromechanical systems, and then assigned this homework. While for students in Summer 2020 cohort, we had just given them instructions on how to solve similar linear circuit problems without using a practical instrument or giving examples of real-world systems following the old syllabus. In addition, we attempted to balance the difficulty level between the assignments from previous years and this year.

Table 4 Details of MAE 40 Homework 5 for Summer 20&21.

Task	Contents	Goal of the Task
T1	Find the general solution of equivalent motor capacitor voltage $V_m(t)$.	Recap the use of Kirchhoff's Voltage Law and general solution of ODEs.
T2	Find the transfer function from motor input voltage to current shaft position $\Theta_m(s)/V_{in}(s)$.	Understand Laplace Transform, transfer functions, and system damping.
T3	Compute the bootstrap (BST) pin voltage, NTC pin voltage, and currents from the schematic of battery management circuit.	Learn to read IC datasheet, and recap Kirchhoff's Laws.

After grading all the assignments, we can compare the Summer 20& 21 students' performance on each of the tasks. In both years the course was taught in a remote format so we do not need to consider the gap between physical and web-based lectures. We present the students scores on all three tasks as in Fig. 11. Each score of the tasks is scaled to 10 points in maximum.



(a) Scores of Summer 2020 Cohort. (b) Scores of Summer 2021 Cohort.

Fig. 11 Scores distribution of Summer 2020&2021.

As can be seen from the grade distribution, the mean of Task 1 scores increases from 7.96 in 2020 to 8.62 in 2021, while Task 2 and Task 3 means increase by 2.25 and 2.59, respectively. The improvement of the students' performance on these tasks can be considered significant especially for Task 2&3 after the introduction of BeagleRover as teaching instrument.

B. MAE 144: Embedded Control & Robotics

Table 5 MAE 144 Assignments during Fall 20.

Assignments	Main Tasks	Goal of the Homework
Homework 1	Assemble the BeagleRover, flash the BeagleBone Blue, and write the first program to drive motors and servos.	Getting started with embedded system setup and programming
Homework 2	Program the microprocessor tasked with counting a wheel encoder. Write the pseudo-code that increments the global variable ‘position’ up or down depending upon the direction of rotation.	Understand the logic and mechanism of rotary incremental encoders.
Homework 3	Rotate the BeagleRover about each of these axis and observe the gyroscope measurements, and write a simple program printing the accelerometer data to the console at 10Hz.	Learn to read and log IMU readings.
Homework 4	Combine the accelerometer and gyroscope readings for better body angles and linear position estimates. Design a high-pass filter for gyroscope and a low-pass filter for acceleromter to provide smooth and precise enough state estimates.	Learn to filter the noisy IMU readings to provide reliable state estimates of the robot.
Final Project	Balance the robot on 2 wheels. Write a program where BeagleRover climbs up against a wall and balance itself with upright pitch angle when standing on 2 rear wheels. Do the feedback control design in MatLab and implement the controller at 100Hz.	Practice the control design workflow using classical design tools to stabilize the pose of an unstable system.

MAE 144 Embedded Control & Robotics is a upper-division undergraduate course offered by the MAE department at UC San Diego. During this class, the students are taught about classical control theories, digital circuits, fundamental dynamics, and how this knowledge set is applied in robotics. This class is commonly known as a practical learning course since students are expected to work on hand-on projects with embedded systems e.g. the EduMIP robot [25] from Renaissance Robotics.

However, during the fall quarter of year 2020, due to the supply chain problems amid the COVID-19 pandemic, the shipping of EduMIP kits from overseas were expected to have a long delay. In light of the situation, we decided to mail the 4 full sets of BeagleRover components in to students who were close to campus (it was during remote-teaching period). Each of these kits can be assembled as one BeagleRover and has 37 3D-printed parts out of the total 54 as discussed in Section II.E. Based on BeagleRover, we assigned 5 assignments for students to work on off-line as listed in Table 5.

On top of the requirement of the final project, one team of the students became the overachievers who managed to balance the robot on both rear wheels pair and left wheels pair as in Fig. 12. They were also able to steer and drive around on 2 wheels such that we gave them extra credits on their final project. Moreover, we collect the students' comments from the course evaluations at the end of the quarter, where the words "fun" and "interesting" were mentioned 11 times in total, and the words "hard", "difficult", and "challenging" appeared 8 times. This actually meets quite well with us instructors' predictions on students' feeling about hands-on classes. However, an unexpected merit of learning with BeagleRover was that it actually helped students stay focused on course even as off-line learning tool, and with adequate community support students could always taught themselves and eventually acquired strong problem solving skills.

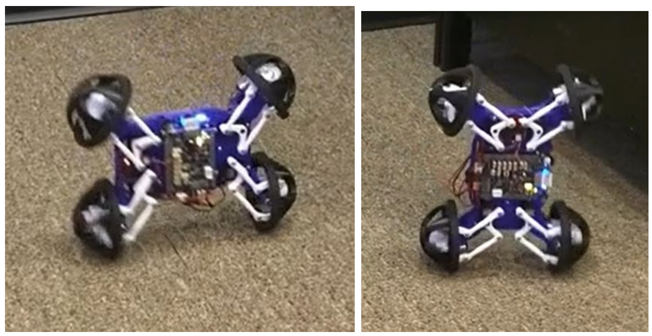


Fig. 12 Demo video screenshots of the MAE 144 final project.

VI. Research Deployment

A. Indoor Navigation and SLAM

Indoor navigation of robotic platforms have played a key role in the history of robotics research. Specifically, understanding how to navigate a previously unseen environment, where the robot needs to understand what is around it, and how to navigate it. This problem is most commonly referred to as Simultaneous Localization and Mapping (SLAM). The BeagleRover's hyper-articulation and 4-wheeled operation capability, places this design at an interesting advantage for indoor exploration of constrained environments. It can tightly navigate areas where traditional 2-wheeled steering systems could not manoeuvre within.

The addition of a camera onto the platform enables the expansion into Visual SLAM. A set of sparse image features can be tracked over time, in image space. With the use of camera models which are either previously calibrated, or auto-calibrated during operation, such image features correspondences can be triangulated into 3D space over time to provide global pose and motion estimates. During one experiment in a home living room, we set up 12 QR codes as landmarks and set up a path for the BeagleRover to perform room coverage while gives localizations of itself simultaneously. With the image data collected, we are able to reconstruct the 3D-map of the room in the form of point cloud data sets as illustrated in Fig 13.

The indoor navigation project has been conducted by the authors on the current BeagleRover platform. Waypoints tracking algorithms are still being developed and increased to obtain the minimum cross-track errors and power consumption. The Vicon mocap readings are used as ground truth, and the fusion of camera, IMU, and encoder data gives the state estimates throughout the experiments.

B. Object Detection

Apart from indoor navigation, we also attempt on realizing the navigation and control of BeagleRover based on object detection in outdoor or fieldwork scenario. For this, we design 2 object detection based experiments for avoiding obstacles and structure from motion (SfM), respectively. To train the correponding weights for different objects, we implement YOLOv5 [26] which is the object detection algorithm based on grid scanning of the image frame. Implementing YOLOv5-based software package on Jetson Nano, the BeagleRover is able to avoid the obstacle, a rock, in the first experiment as illustrated in Fig. 14. And in the second experiment (Fig. 15) we perform SfM around the target object, a cup of orange juice, by driving the BeagleRover around the object and collects image data automatically.

Fig. 13 Point cloud-based 3D map generated using SLAM on BeagleRover.

(a) Detected obstacle. **(b) Steer to avoid obstacle.**

Fig. 14 Obstacle avoidance experiment

Fig. 15 Structure from motion experiment.

VII. Conclusions

This paper introduces the design and implementation of the robotic platform BeagleRover developed for multiple educational and research purposes. We can draw the conclusion that low-cost 3D-printable mobile robots like BeagleRover can serve various purposes such as practical learning tool as well as research platform for robotic control and vision. Besides, with proper kinematics and control design, BeagleRover can realize complicated motions which are usually not possible for traditional 2-wheel or 4-wheel low-cost mobile robots, which allows for better maneuverability.

However, there are also some unexpected challenges occurring from both teaching-and-learning side and robotic research side. During MAE 144 we found that code-based approach to start with robotic programming could still be difficult even for students in engineering disciplines. In some of the "traditional" engineering departments like aerospace engineering, mechanical engineering, and material science, teaching robotics is a difficult task since students can feel unfamiliar with basic understanding of embedded systems and programming logic. On the other hand, the limited on-board computational resources prevent many tasks to be fully run within the BeagleRover, making outdoor SLAM and navigation deployment currently non-achievable.

The authors' future work with BeagleRover will feature design of better graphical or modular programming interface to lower the learning barrier with BeagleRover. In addition, the authors would like to thoroughly study the correlation between teaching instrument robots and students' performance. Last but not least, the authors would like to develop robust SfM algorithms which can run on the fly for field deployment of BeagleRover.

Acknowledgement

This work was supported by the National Institute for Standards and Technology (NIST) under award #70NANB17H21, "Improving Disaster Resilience Through Scientific Data Collection with UAV Swarms," and the National Science Foundation (NSF) under award #DGE-0966375, "Training, Research and Education in Engineering for Cultural Heritage Diagnostics." Additional support was provided by the NSF under award #CNS-1338192, "MRI: Development of Advanced Visualization Instrumentation for the Collaborative Exploration of Big Data," as well as the Kinsella Heritage Engineering Fund and the Qualcomm Institute at UC San Diego. Opinions, findings, and conclusions from this study are those of the authors and do not necessarily reflect the opinions of the research sponsors . The authors would also like to thank the students of MAE 40 and MAE 144 in year 2020 and 2021 at UC San Diego for their participation in course-related research.

References

[1] Resnick, M., Ocko, S., et al., *LEGO/logo–learning through and about design*, Epistemology and Learning Group, MIT Media Laboratory Cambridge, 1990.

[2] Rollins, M., and Lowe, M., *Beginning Lego Mindstorms Ev3*, Vol. 253, Springer, 2014.

[3] Seo, K., and Robotics, A., "Using nao: introduction to interactive humanoid robots," *AldeBaran Robotics*, 2013.

[4] makeblock, "Mbot," , 2018. URL `https://www.makeblock.com/mbot-3`.

[5] Strawson, J. R., *Feedback Control Driven Mechanical Design Optimization*, University of California, San Diego, 2018.

[6] "BeagleBone® Blue," , 2021. URL `https://beagleboard.org/blue`.

[7] de Souza, T. L., and Elisiario, L. S., "Educational robotics teaching with Arduino and 3d print based on stem projects," *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*, IEEE, 2019, pp. 407–410.

[8] Gonzalez-Gomez, J., Valero-Gomez, A., Prieto-Moreno, A., and Abderrahim, M., "A new open source 3d-printable mobile robotic platform for education," *Advances in autonomous mini robots*, Springer, 2012, pp. 49–62.

[9] Lapeyre, M., Rouanet, P., Grizou, J., Nguyen, S., Depraetre, F., Le Falher, A., and Oudeyer, P.-Y., "Poppy project: open-source fabrication of 3D printed humanoid robot for science, education and art," *Digital Intelligence 2014*, 2014, p. 6.

[10] Gul, J. Z., Yang, Y. J., Su, K. Y., and Choi, K. H., "Omni directional multimaterial soft cylindrical actuator and its application as a steerable catheter," *Soft robotics*, Vol. 4, No. 3, 2017, pp. 224–240.

[11] Strawson, J., Cao, P., Tran, D., Bewley, T., and Kuester, F., "Monocoque Multirotor Airframe Design with Rotor Orientations Optimized for Direct 6-DoF UAV Flight Control," *AIAA AVIATION 2021 FORUM*, 2021, p. 2431.

[12] Strawson, J., Cao, P., Bewley, T., and Kuester, F., “Rotor orientation optimization for direct 6 degree of freedom control of multirotors,” *2021 IEEE Aerospace Conference (50100)*, IEEE, 2021, pp. 1–12.

[13] De Vivo, L., Tran, D., and Kuester, F., “Towards Design of a 3D Printable Prandtl Box-Wing Unmanned Aerial Vehicle,” *2019 IEEE Aerospace Conference*, IEEE, 2019, pp. 1–17.

[14] Muhammad, A., Abbas, S., Manzoor, T., Munawar, A., Abbas, S., Hayat, M., Abbas, A., and Awais, M., “Marwa: A rough terrain landmine detection robot for low budgets,” *Proceedings of the 43rd International Symposium on Robotics*, 2012.

[15] Colaprete, A., Andrews, D., Bluethmann, W., Elphic, R. C., Bussey, B., Trimble, J., Zacny, K., and Captain, J. E., “An overview of the volatiles investigating polar exploration rover (viper) mission,” *AGU Fall Meeting Abstracts*, Vol. 2019, 2019, pp. P34B–03.

[16] “AOPOY RC Stunt Car Toys for Kids 5-7,” , 2021. URL `https://www.walmart.com/ip/AOPOY-RC-Stunt-Car-Toys-Kids-5-7-Years-Remote-Control-Gyro-Upright-Rollover-Toy-2-4Ghz-High-Speed-Off-road-Vehicle-Inflatable-Tires-LED-Lights-Blue/`.

[17] Barthelmes, S., and Zehnter, S., “An all-terrain-controller for over-actuated wheeled mobile robots with feedforward and optimization-based control allocation,” *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 5215–5222.

[18] , 2020. URL `https://ocw.metu.edu.tr/pluginfile.php/6885/mod_resource/content/1/ch7/7-1.htm`.

[19] Hulttinen, L., and Mattila, J., “Flow-limited path-following control of a double Ackermann steered hydraulic mobile manipulator,” *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, 2020, pp. 625–630.

[20] Choi, M. W., Park, J. S., Lee, B. S., and Lee, M. H., “The performance of independent wheels steering vehicle (4WS) applied Ackerman geometry,” *2008 International Conference on Control, Automation and Systems*, IEEE, 2008, pp. 197–202.

[21] Thuilot, B., Cariou, C., Martinet, P., and Berducat, M., “Automatic guidance of a farm tractor relying on a single CP-DGPS,” *Autonomous robots*, Vol. 13, No. 1, 2002, pp. 53–71.

[22] Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E. N., “The explicit linear quadratic regulator for constrained systems,” *Automatica*, Vol. 38, No. 1, 2002, pp. 3–20.

[23] Systems, D., and Laboratory, C., “EduMIP ROS,” , 2017. URL `https://dscl.lcsr.jhu.edu/home/courses/edumip_ros/`.

[24] de Oliveira, M. C., “pyctrl: a Python Suite for Systems and Control,” 2017.

[25] Bewley, T. e. T., “EduMIP,” , Feb 2018. URL `https://beagleboard.org/p/edumip/edumip-13a29c`.

[26] Jocher, G., Nishimura, K., Mineeva, T., and Vilariño, R., “Yolov5,” *Code repository https://github. com/ultralytics/yolov5*, 2020.