

2-DOF Arm and Loader Robot for Jurassic Park

Grant Keefe, MREN 303 Student AND Tony Yang, MREN 303 Student
(20gak5@queensu.ca) (yang.tony@queensu.ca)

Queen's University, Kingston, Canada

Abstract - This paper presents the design and development of a specialized robot for a rescue and retrieval operation as a part of the MREN 303 design course. Drawing inspiration from historical disaster recovery robots, the design integrates an adaptable mechanical structure with autonomous navigation systems like line following and particle filter localization. The Robot showcased effective navigation in tests, though encountered challenges with power management and localization accuracy. These results highlight the potential of autonomous robots in emergency scenarios. Insights gained offer valuable lessons for future iterations, including enhancements to battery pack design and improvements in localization methodology for more effective autonomous navigation.

0 INTRODUCTION

For the MREN 303 design course, our team was confronted with a challenge reminiscent of the chaotic scenario depicted in Jurassic Park: rescuing an unconscious individual, Ken, from a pit atop a hill within the confines of a park teeming with dinosaurs. The goal was to devise a method to safely deliver Ken to the helipad while simultaneously corralling the dinosaurs back into their enclosures. To tackle this task, we embarked on the design, assembly, and programming of a specialized robot.

Taking cues from past robotics endeavours such as the STR-1 robot and Mobot, which played pivotal roles in the cleanup operations following the Chernobyl nuclear disaster of 1986, we drew inspiration for our robotic solution. The Mobot's ability to traverse hazardous environments, akin to our current predicament, served as an inspiration. Furthermore, its loader and excavator design, reminiscent of a backhoe loader, made it versatile and agile—qualities valuable in maneuvering around the area, picking up dinosaurs, and grabbing Ken.

For autonomous navigation, our strategy revolved around two primary methodologies: either by line-following or using encoder and ultrasonic sensor data for offline navigation and localization. Our manual mode goals were geared towards prioritizing Ken's rescue while corralling as many dinosaurs as feasible within the allotted time frame.

With careful planning and execution, we anticipated achieving autonomy in navigation, effecting Ken's rescue with minimal harm, and corralling a significant number of dinosaurs, aiming for a top-four placement in the competition.

1 METHODS

Figure 1 shows the final robot during competition as it aims to corral dinosaurs. In the proposed design, gear ratios and selections were optimized for efficiency, allowing the lifting capabilities of the actuated bucket and arm system to behave predictably. The selection of spur gears in some parts of the assembly was made out of space and material restrictions. When able helical gears were utilized for the desirable loading properties. A parallel linkage mechanism was incorporated to maintain the stability of the bucket and level operation. Control algorithms, such as the simple line follower and more sophisticated particle filter localization algorithm, were developed to allow for autonomous mobility. This approach ensures redundancy in controls and strong mechanical support for the execution of tasks.

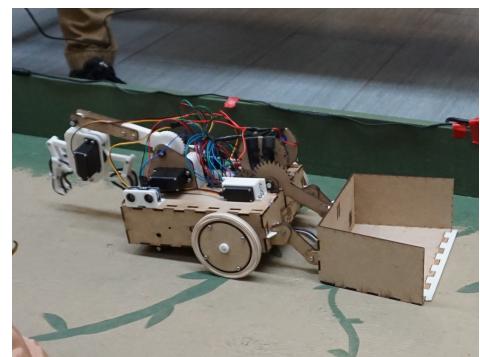


Fig. 1. Final design of the robot during competition

1.1 Mechanical

The goal of the mechanical design was to be able to lift the dinosaur with an actuated arm, weighing 0.174 kg and multiple dinosaurs weighing 0.044kg with a bucket. The methods of mechanical design and mechanisms utilized are described below.

1.1.1 Gear Ratios

Front Loader and Arm lifting Gearbox:

$$\text{Gear Ratio} = \frac{\text{Number of Teeth on Gear}}{\text{Number of teeth on pinion}} = \frac{30}{15} = 2 \quad (1)$$

Torque calculations:

$$\tau_{\text{out}} = \tau_{\text{in}} \times \text{Gear Ratio} = (4.1) \times 2 = 8.2 \text{ kg} \cdot \text{cm} \quad (2)$$

Lifting capabilities of bucket:

$$\text{Max weight (bucket)} = \frac{\tau_{\text{out}}}{\text{dist.}} = \frac{8.2}{24} = 0.34 \text{ Kg} \quad (3)$$

Lifting capabilities of arm:

$$\text{Max weight (arm)} = \frac{\tau_{\text{out}}}{\text{dist.}} = \frac{8.2}{13} = 0.63 \text{ Kg} \quad (4)$$

With the addition of the geared motors the bucket and arm should comfortably handle the loads applied from both dinosaurs and Ken with a margin of safety to account for losses due to friction and unaccounted for weights[1].

1.1.2 Gear Selection

Helical gears exhibit less bending moment at the tooth root when compared to spur gears because of the slanting line of contact. This reduced stress is advantageous for 3D printed parts as PLA is prone to failure along layer lines. Spur gears were chosen, were used as a method of saving space, while utilizing the MDF provided and not wasting needed print volume[2].

1.1.3 Parallel Linkage

Parallel Linkages, as shown in Figure 2 are a mechanism applicable when the needed task requires the mechanism to remain level throughout actuation. This is applicable to lifting and dropping dinosaurs at the desired time. The mechanism also has the benefit of distributing the joint loading allowing for more weight to be stabilised[3].

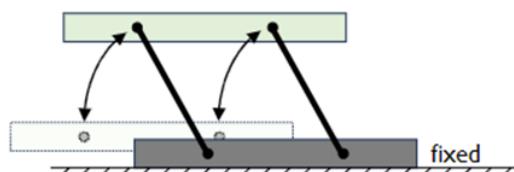


Fig. 2. Parallel linkage diagram with one fixed frame showing the motion and DOF of moving body[4]

1.1.4 Final Robot Design

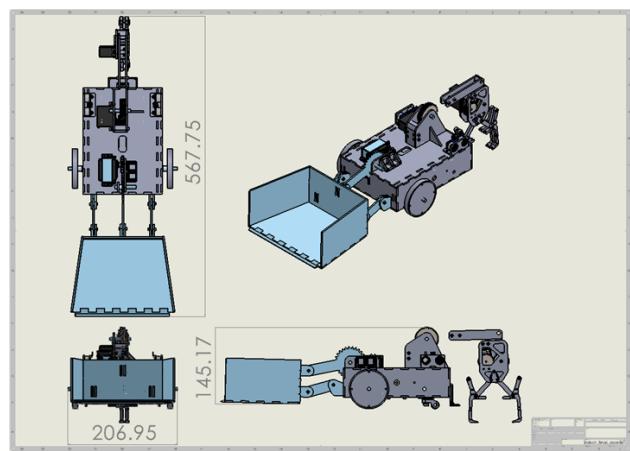


Fig. 3. Solid works drawing of final robot assembly with dimensions

Figure 3 demonstrated the scale of the robot. A key design constraint being its width can be at maximum 250mm. The design is well below that allowing there to be a margin of error when completing tasks.

1.1.5 Materials Used

Table 1. Component Volumes and Materials

Part	Volume (cubic inches)	Material
Wheels	2.07	TPU
Gripper	2.32	PLA
Spin Wheel	1.23	PLA
Rod Spacers	0.51	PLA
Motor Mounts	1.22	PLA
Servo Mount	0.32	PLA
Total:	7.67	

Utilizing less filament than the allotted amount, seen in Table 1 Total, lead to an over usage of the MDF, with the percentage surface area used being roughly 80-90 percent.

1.2 Electrical

The robot featured two DC motors for its driving actuation, complemented by three servos dedicated to the operation of the bucket, arm, and gripper. Autonomous navigation capabilities were possible by the inclusion of three ultrasonic sensors, encoders, and a line-following sensor. A raspberry Pi Pico was used as the microcontroller for the robot. Figure 4 shows the wiring schematic of the robot.

1.3 Manual Controls

To facilitate manual control of the robot, a Logitech controller was employed. Connected to a laptop running a Python script, the controller transmitted states as TCP packets via a local area network (LAN), which the Raspberry Pi Pico received to execute commands. One joystick governed the robot's locomotion, while the other managed its arm movements. The left and right trigger buttons were

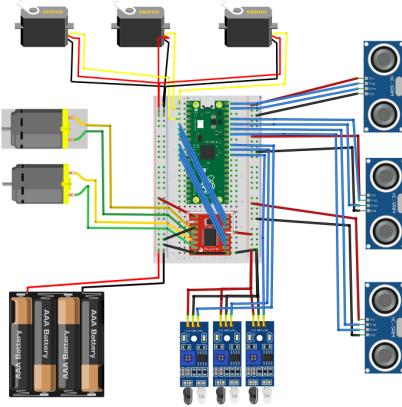


Fig. 4. Wiring diagram of the robot

designated for bucket and gripper control, respectively. Additional functionalities, including battery status retrieval, mode adjustments, and driving direction toggling, were assigned to buttons A, B, and X, respectively.

1.4 Autonomous Algorithms

Line following are the two areas of navigation the team researched, the more simplistic approach being line following. For this reason, we used line following as a backup plan given the localization algorithm did not behave as expected.

1.4.1 Line Following

A simple proportional controller, shown in Figure 5, was used to follow the line. The control loop adjusts the right and left wheel Pulse Width Modulation PWM values by a proportional term K_p , depending on the robot's state on the line.

1.4.2 Particle Filter Localization

A basic overview of the localization approach is shown in Figure 6, consisting of 4 crucial steps.

Initialization: Upon start of the autonomous mode, the estimate position of the robot can be determined inside of the environment. This start position estimate is assumed to be accurate through each test. The initialization distributes particles around this start position, indicating possible true robot states[5].

Prediction: Using dead reckoning, the motion of the robot is estimated with encoder over each sample time. The particles generated in the previous sample time are shifted utilizing the state transition (dead reckoning) of the robot.

$$x_t^i = f(x_{t-1}^i, u_t) + \varepsilon \quad (5)$$

Where f is the state transition, u_t are the encoder ticks, and ε accounts for motion noise[5].

Update: Upon measuring the three sonar sensors oriented as $[-\frac{\pi}{2}, 0, \frac{\pi}{2}]$ wrt. the robot frame, all the particle weights can now be adjusted. This adjustment is completed by simulating the three sonar readings at each particle and

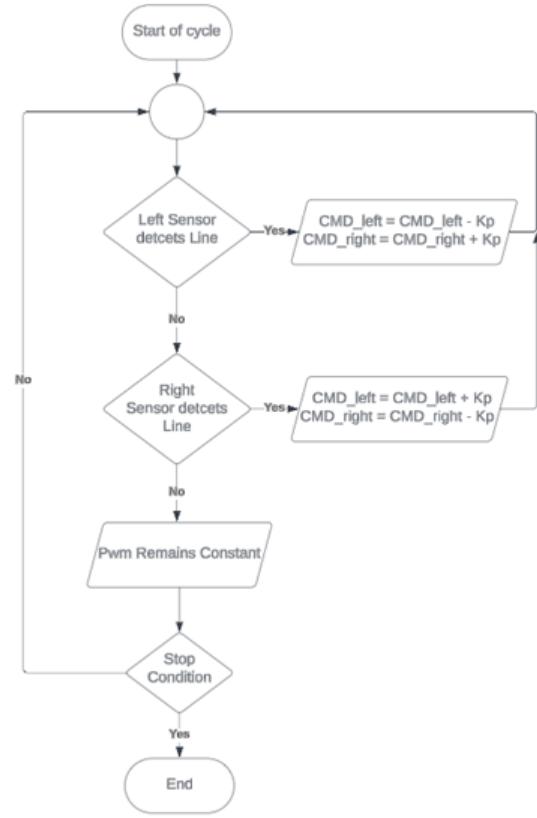


Fig. 5. Autonomous loop for line following with proportional control

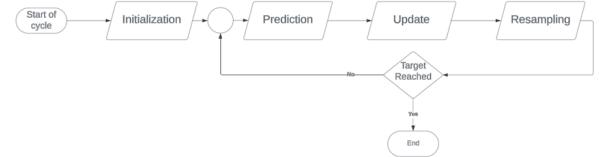


Fig. 6. Simplified Block Diagram of Particle filtration loop

comparing the simulated ranges to the true ranges. The error in these values determines the weight of the particle.

$$w_t^i = p(z_t | x_t^i) \quad (6)$$

Where p is the likelihood of observing the true sonar reading z_t at the particle(state) x_t^i [5].

Resampling: Particles with higher weights are kept, while lower weighted particles are discarded. This is set by a w_{thresh} . A new set of particles are distributed around the higher weighted state estimates, and the highest weighted state estimate is passed on as the estimated robot position[5].

The Simulink model in Figure 7 shows the implementation of the simplified block diagram in Figure 6, the key features being the Occupancy grid generated on startup with 1cm x 1cm resolution, goal way-points, Adjust robot pose which handle the particle filtration, and the robot path

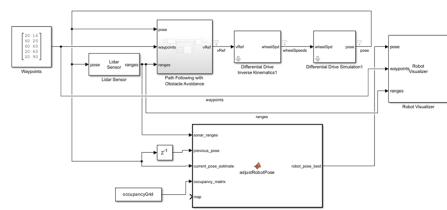


Fig. 7. Overview of particle filter controls in Simulink.

following and inverse kinematics to simulate dead reckoning.

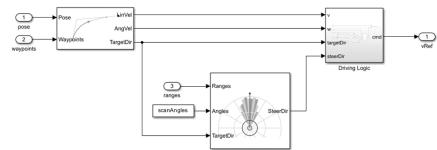


Fig. 8. Pure Pursuit subsystem in Simulink model

The Path following with obstacle avoidance subsystem in Figure 8 determines left and right wheel velocities using pure pursuit between current and desired position as well as avoidance implemented with the simulated scan distances from sonars.

2 RESULTS

2.1 Line Following

When the value of K_p was set 100 the line following program showed repeated success. Pre-competition testing showed a success rate of 100 percent excluding proportional term tuning. The in competition results followed the same trend with no failures occurring.

2.2 Path Following

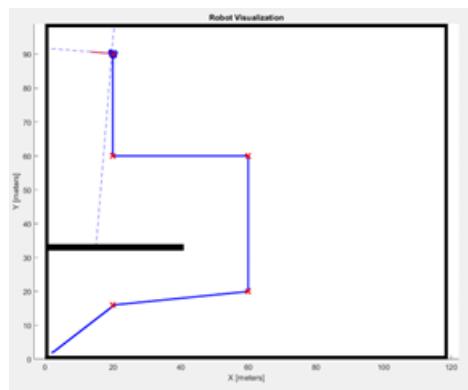


Fig. 9. Performance of path following pure pursuit algorithm in simulated environment

As seen in Figure 9 the path planning simulation performed as expected with the robot navigating to each goal pose accurately. The model assumed ideal conditions with infinite friction, allowing the inverse kinematics to achieve

little error. This result was positive for the progression of the localization algorithm as it was designed to account for the addition of un-ideal conditions.

2.3 Localization

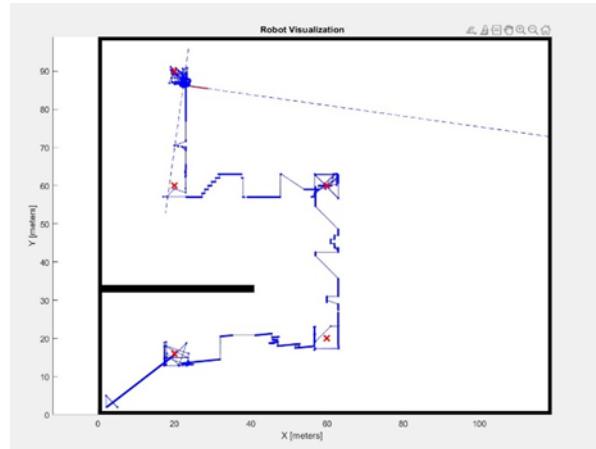


Fig. 10. Performance of particle filter localization in simulated environment

As seen in Figure 10 localization algorithm was noisy and ineffective. The result is largely due to the lack of sensor data. Particle filtration operates more effectively when z_t contains a larger set of data points. The larger sample size makes the robot state probability weight more accurate and reduces false robot states.

2.4 Performance in Competition

Qualifying Round 1: During the initial qualifying round, our robot encountered setbacks. It failed to follow the line in autonomous mode, resulting in a collision with a wall. Later, when control was regained, it became lodged in the gate, necessitating manual intervention to pass through, deducting us a single point. Additionally, it couldn't corral any dinosaurs or rescue Ken, leading to a final score of -1.

Testing Practice: Before the second qualifying round, rigorous testing and practicing was conducted. The robot executed line following ten times consecutively and successfully retrieved Ken and one dinosaur 9 times out of 10.

Qualifying Round 2: In the subsequent qualifying round, the robot successfully followed the line, retrieved Ken, and placed him on the helipad. However, it faced difficulty in grasping the dinosaurs, yielding a total of 9 points.

Competition Round 1: Despite successful line following, a mishap occurred when Ken slipped from the gripper as the robot traveled down the ramp, remaining on the ramp, which posed challenges for the robot to grab Ken again. Consequently, only one dinosaur was retrieved, resulting in a total of 5 points, sufficient for advancement to the next round. The next round was hosted on the next day but unfortunately, the robot's batteries were not recharged ultimately leading to the team's resignation from the competition. Figure 11 shows the robot grabbing Ken from the competition day.



Fig. 11. Robot grabbing Ken during competition.

3 DISCUSSION AND CONCLUSION

Our hypothesis was incorrect, the team did not place top 4. Power requirements was a limitation for the final robot and ultimately caused a forfeit during competition. The design of the secondary battery pack could have been significantly improved with the utilization of 3D printed parts. This addition could allow for easier recharging of the batteries as well as swapping them when dead.

The final robot design performance was also limited in its ability to localize. This limitation forced the method of autonomous movement to the fallback of line following. An improvement to the robot design to allow for localization to be more effective would be the actuation of one or more of the sonars in order to collect more data points. This improvement would limit the remaining actuators for the arm,

gripper and bucket mechanism but would have drastically improved the particle filters ability to select the most accurate robot states.

4 REFERENCES

- [1] S. M, *et al.*, “What is gear ratio? [how to calculate gear ratio with formula],” <https://www.theengineerspost.com/gear-ratio/> (2024), accessed: April 19, 2024.
- [2] British Standards Institution, “Calculation of load capacity of spur and helical gears – Part 3: Calculation of tooth bending strength,” BS ISO 6336-3:2006, Incorporating corrigendum June 2008 (2006 October).
- [3] C. Rasmussen, “How to use Parallel link mechanism and a warning,” <https://mentoredengineer.com/parallel-links/> (2024), accessed: 2024-04-19.
- [4] A. Wu, “MREN 303: Mechatronics Robotics Design III,” (2024), unpublished course notes, Department of Mechanical and Materials Engineering, Queen’s University, Kingston, ON.
- [5] J. Elfring, E. Torta, R. van de Molengraft, “Particle Filters: A Hands-On Tutorial,” *Sensors (Basel)*, vol. 21, no. 2, p. 438 (2021 jan).

5 AUTHOR CONTRIBUTIONS

Grant Keefe: Methodology, Formal Analysis, Software, Writing textbf– original draft

Tony Yang: Conceptualization, Software, Writing textbf– review and editing

THE AUTHORS



Grant Keefe

Grant Keefe is a Mechatronics and Robotics student. He has interest in the areas of aerial robotics and aerospace engineering



Tony Yang

Tony Yang is a Mechatronics and Robotics Engineering Student. He is interested in mobile robotics, autonomous vehicle navigation, and mapping