Development of Motion Planning Algorithms for Inchworm Robots

A WPI RBE 550 Project Spring 2021

Alex Tacescu
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts
actacescu@wpi.edu

Nicole Kuberka
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts
nlkuberka@wpi.edu

Meha Mohapatra
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts
mmohapatra@wpi.edu

Sri Kalimani
Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts
skalimani@wpi.edu

***Abstract*—In this paper, we focus on developing the inchworm's path and motion algorithms. A path planning system will be developed to consider variables such as space and collision avoidance. To do this, several algorithms will be considered, such as A*, RRT*, or RAGS (Risk Averse Graph Section). To produce the inchworm motion, we look into various footstep planners and analyze the kinematics and dynamics to achieve a walking gait. These algorithms will be tested in our simulated environments within Gazebo and RVIZ.**

I. Introduction

We will plan the motion of a 5 DOF inchworm robot from a start to end node. As we progress in the project we plan on making the creation of the path more complex by either introducing more obstacles or adding more robots to the system. If we have time we will attempt to ferry blocks from one location to another.

II. Related Work

A. Navigation Planning Robotics

Chestnut, 2007, designed the Adaptive Action Model that uses a search algorithm based on its suitability to the current environment. On obstacle-free, flat ground, the planner uses action set directly as the set of actions for node expansion. In the presence of rough terrain or obstacles, the algorithm performs a local search around $x\prime$ to find a new state, $x\prime\prime$, and compute a new action to reach that state, $a\prime$. This approach provides multiple local search mechanisms to drive footstep planning. This is important in the case of an inchworm, especially when we navigate around obstacles. While this approach uses multiple options, it does not guarantee optimality. Optimality is an important consideration in the light of swarm construction as this involves multiple robots. In order to allow each robot to behave efficiently, we need to provide close to optimal paths for navigation. [4]

B. Swarm Scaffolding M.A.R.I.A Legacy Project

The current iteration of the project, M.A.R.I.A, uses inchworm robots to build structures. To do this, they focused on making smart blocks and developing a construction algorithm

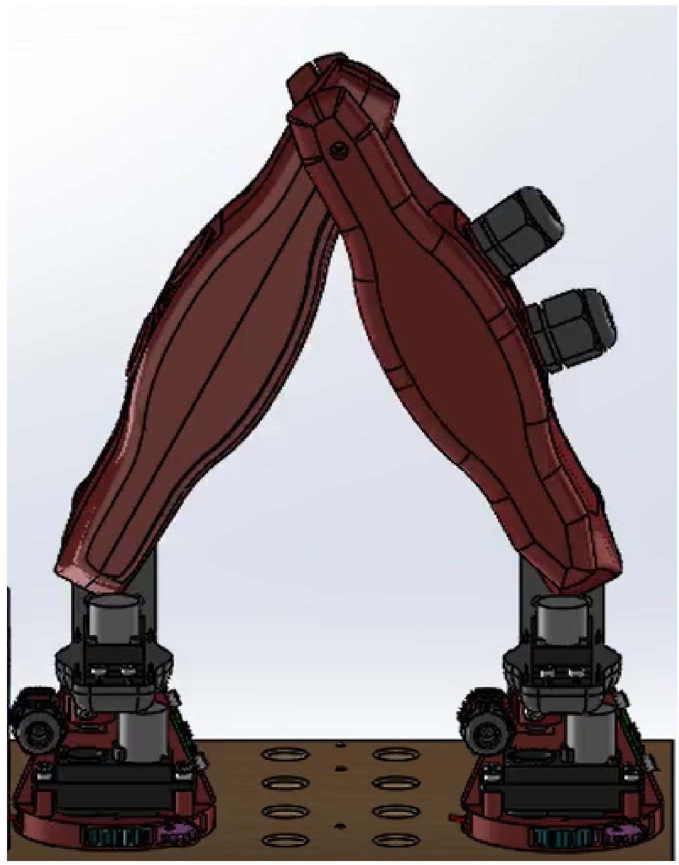

Fig. 1. SolidWorks model of the inchworm robot used in this project [2]

that creates the blueprint for the desired structure. These blocks take inspiration from the decentralized manner of stigmergic building by introducing intelligent building material with no centralized planner. By doing so, the blocks will be able to guide the inchworms with simple instructions to create complex structures. This allows for the inchworm robot to only concern itself with creating a path to place the block and collision detection in the structure according to the block algorithm.

In the previous iteration of this project (Swarm Scaffolding) developed the inchworm robot design and rudimentary motion planning to navigate the workspace area (see Figure 1). The robot's dimensions and gait are based on this design. The team developed a novel Face* algorithm to go from block A to block B within the structure. This is a 3D adaptation of A* and in addition, accounts for reachability of the end effector. The algorithm models each step as two types: (1) a step that results in different orientations between the end effectors; (2) a step that results in the same orientations between the end effectors. The algorithm filters out faces (nodes) based on whether they match the base orientation. Although this algorithm is complete, it does not guarantee optimality. [2]

C. BILL-E

BBILL-E is a 2019 MIT project that attempts to solve a similar project statement as ours - create a multi-robot construction system. To reach this goal, they use voxels (basic blocks) and inchworm-inspired robots. Their success in efficient building is due to their simple motion-planning algorithm which is based on counting the steps the robot takes around the structure. While this helps the robot know where it is with relation to the structure, it poses a drawback - the robot can not leave the structure and a human must add the next block to the structure. Therefore if the robot is detached from the structure, it will not know where it is or how to continue construction. In our project, we propose that the inchworm will ferry the building material from a quarry to the structure, allowing for a better replica of a construction site layout. [10]

III. Goals

We have created minimum, expected, and reach goals for our project. Our minimum goal is creating a fully setup simulation playground which includes a robot able to follow the gait. This simulation is necessary to test and complete the path and motion planning for our project. We will be able to visualize what the algorithm is doing and debug. The expected goals are creating a path for the robot and footstep planning for just the robot. We aim to visualize the path created and see the robot taking the path by the end of our project. The reach goals we have considered are moving with a block, creating a 3D walking gait, 3D path planning of going up walls, and implementing multiple robots into the system. These reach goals break into two distinct routes of expansion: introducing blocks or multiple robots. To introduce blocks, we will develop the gait for the inchworm to move with a block.Then, we will work on the 3D walking gait for going up walls and traversing corners . With a complete gait, we can develop a 3D path planning algorithm which allows for multi-level structures. To expand on the swarm route, we will add multiple robots to the system and work on collision detection.

Minimum Deliverables

- Fully setup simulation playground
- Inchworm robot able follow the gait

Expected Deliverables

- Functional Path planning algorithm visualized in the simulation
- Foot step planning for inchworm robot
- Inchworm robot followed layout plan in simulation

Reach Deliverables

- Motion planning of robot with block
- Creation of a 3D walking gait
- 3D path planning
- Creating a multi-robot system

IV. Proposed Methods

In order to complete the goals we have outlined, we have split the project into a few major sections. First and foremost is the simulation environments, which will help us debug and develop our algorithms. Second major goal is to plan the path of an inchworm from one place to another, a path planning algorithm will be developed and implemented. Finally, a motion planning algorithm will be given the path created and simulate the robot walking the path.

A. Simulation Environment

Since the Covid-19 pandemic is preventing testing with physical hardware, a full physics simulator will be used to test the motion planning software that is designed and implemented. Simple simulators will also be developed to aid in the development of complicated path planning algorithms.

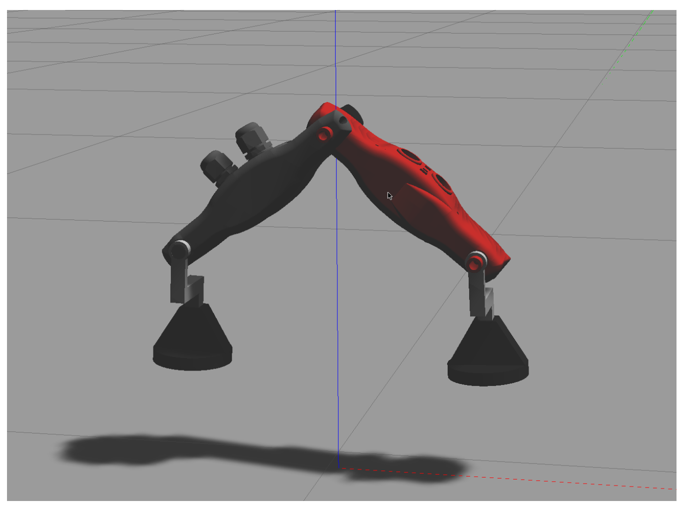

Fig. 2. Inchworm modeled in Gazebo (simulator) [2]

RVIZ will be the tool of choice to display and loosely simulate the path planning algorithms developed. Since physics is not considered, many of the variables that are not necessary for initial development of the path planning and footstep algorithms can be isolated and temporarily removed to simplify the software. These sudo-simulators can also be turned into visualizers to better understand what the system is doing when all of the algorithms are implemented and running together.

To test the robot in a virtual playground, a Gazebo simulator will be used with a playground and inchworm model. Since

Gazebo is a full physics simulator, the inchworm model will need to include mass, inertia, and a semi-accurate model of the end effector. This includes a method to approximate the magnetic end effector of the inchworm. A Gazebo Vacuum Gripper Plugin will be used to achieve this [1][14], since the desired effect of a magnetic gripper is similar to a vacuum gripper, and developing a new plugin for Gazebo is out of the scope of this project. If we reach our expected goals, a model of the cube will be developed, which will include its own controller to simulate the connection between different cubes.

B. Path Planning

A path planning algorithm will be implemented to generate a path between the current position of the robot and the goal node, as determined by a higher-level system. The system should consider several robot conditions, including (but not limited to) inchworm step limitations, movement constraints, and collision avoidance [13] (if the project progresses to a point where the inchworm is being tested in a cluttered environment). Several path planning algorithms are currently being considered, including A*, RTT*[11], and RAGS (Risk Averse Graph Section). A relatively simple algorithm (A* or RTT*) will likely be implemented first, since initially the inchworm will be tested on a perfectly level and empty playground. If time allows, RAGS [9] may be implemented and compared to other algorithms, especially if the project leads to more advanced 3D graphs and maps.

The map is very important to a successful path planning algorithm. This project will initially use a simple 2 dimensional map (stored as a graph) to simplify the initial implementation of the path planning algorithm developed. If the climbing reach goal [insert ref to goal] is attempted, the map will need to consider a 3 dimensional environment. The first solution to be tested will be a “2.5” dimensional map, where the height will be added as a 3rd parameter. This may work for situations where the inchworm will only use the upward-facing face of the blocks in a structure. For situations where more advanced climbing is required, a fully 3-dimensional map will be required.

A map also needs to be generated given a playground. Perception is beyond the scope of this project, so the map will be generated based on a list of building blocks and their positions maintained by Gazebo. Initially for the 2 dimensional map, occupancy grids will be used as the map data structure, and blocks will be considered obstacles to be avoided by the robot. If the project leads to a 3 dimensional map, octrees are planned to be used as the map’s underlying data structure. One limitation that may cause issues during development is definition of what is graspable and what is not. To solve this, the occupancy grid may also contain information on the graspability of the position. This information will then be relayed back to the main path planner to be considered as a heuristic.

C. Motion Planning

The second major part of this project is the motion planning of the inchworm itself. While the path planning algorithm is in charge of determining an achievable path for the inchworm, the motion planning algorithms implemented will be in charge of actually executing that path. This major task can further be split into three major sections: inchworm kinematics, movement gaits, and footstep planning.

The 5 degree-of-freedom inchworm kinematics will be kept relatively simple, and dynamics will be automatically considered by a smart joint controller built into ROS and Gazebo [5]. This is meant to simplify the development of the project, since dynamics of the robot is beyond the scope of this project. The overall kinematics of the inchworm will be similar to a robot arm, since the robot will have a stationary base during the movement gaits. However, two different kinematic controllers will need to be developed, depending what side of the inchworm is grounded [15].

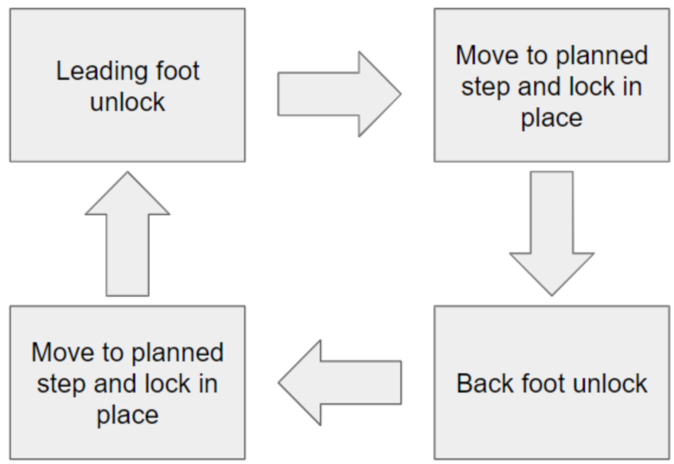

Fig. 3. Our movement gait state machine flowchart

The movement gaits include the general walking gait and the climbing gait. Ideally, both gaits will be combined into one main gait which can climb and walk around a playground. The gait itself will essentially be a large state machine (see Figure 3). Initially, the robot is assumed to be in a position where both end effectors are locked into the playground. The leading foot will then be unlocked and moved to the next footstep position (as generated by the footstep planner). Then the leading foot will be placed in the correct position and locked down. Finally the back foot will unlock and move to its designated footstep position, and lock down. The cycle will loop back to the beginning until the inchworm reaches its final footstep position [7] [3] [4].

The final part of the motion planner will be the footstep planner. Inspiration will be taken from bipeds and quadrupeds [4], as many footstep planners exist for those types of robots. There are two options for footstep planners for the inchworm. A preemptive footstep planner generates all possible footsteps from any one location recursively, and places them in a graph.

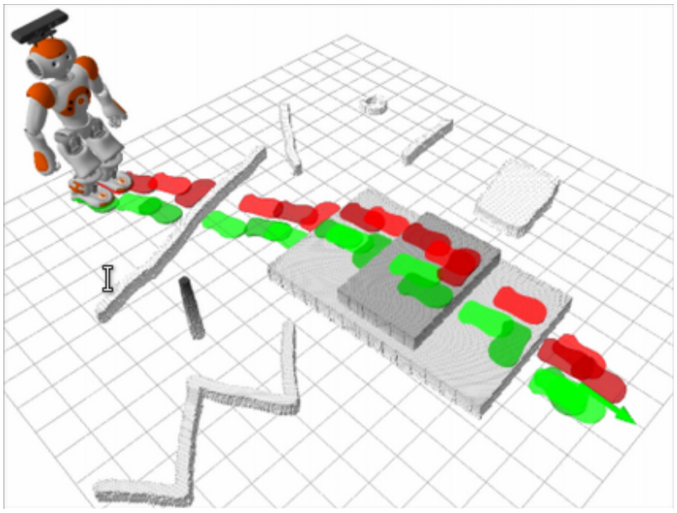

Fig. 4. An graphical example of a search-based footstep planning using a 2-legged humanoid robot [8]

Then it can use a graph search (like A*, WA*, ARA*, RAGS, etc) to find the optimal footsteps throughout the path [8]. This option is the simplest and will likely be the one implemented for this project. However, a more comprehensive and complete system is a dynamically generating real-time footstep planner. Such an algorithm can choose the best footsteps as the inchworm is moving, given real-time map data and information, by implementing modified versions of existing model predictive control algorithms (such as SLQ - Sequential Linear Quadratic - controllers) with different heuristics [12] [6]. While this algorithm can lead to a much more comprehensive and complete footstep planner in dynamic environments, it is also relatively over-complicated for the static playground that we are assuming to test the inchworm in simulation.

V. Metrics for Evaluation of Success

The project's baseline success metric will be the successful path planning and traversal of an inchworm robot in a playground simulated in Gazebo. This will require a working path planning algorithm as well as a fully developed motion planning system. This project is focused on gait work and simulating the robot following the created path. We will create a path planning algorithm that can create an optimized path to a determined location. We have larger hopes to possibly get to a bigger picture with multiple robots maybe with collision detection but this is past our evaluation of success.

VI. Schedule

The project has been split into three different sections: development environment setup, motion planning of the inchworm robot, and path planning of the inchworm. The development environment setup effort will be spearheaded by all 4 team members. After the environment is set up, two developers will be tasked with motion planning, while the other two will develop the path planning algorithms required. If all expected deliverables are completed, the team will re-evaluate the task

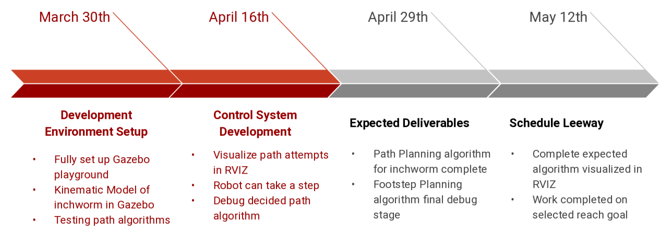

Fig. 5. Proposed schedule for the project

division for reach goals. It is important to note that 2 weeks is dedicated to schedule leeway to deal with unexpected delays and/or development problems with previous tasks.

References

[1] [Online]. Available: http://docs.ros.org/en/melodic/api/gazebo_plugins/html/group__GazeboRosVacuumGripper.html.

[2] Cameron Collins, Josue Contreras, Neel Dhanaraj, Hannan Liang, Trevor Rizzo, Caleb Wagner, "Swarm construction: A method in multi-agent robotic assembly," 2020. [Online]. Available: https://web.wpi.edu/Pubs/E-project/Available/E-project-051620-143223/unrestricted/3DSwarmConstruction_FinalReport.pdf.

[3] I.-M. Chen, S. H. Yeo, and Y. Gao, "Locomotive gait generation for inchworm-like robots using finite state approach," *Robotica*, vol. 19, no. 5, pp. 535–542, 2001. DOI: 10.1017/S0263574700003271.

[4] J. Chestnutt, "Navigation planning for legged robots," 2007. [Online]. Available: http://www.cs.cmu.edu/~cga/papers/chestnutt-thesis.pdf.

[5] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. Rodríguez Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. Fernández Perdomo, "Ros_control: A generic and simple control framework for ros," *The Journal of Open Source Software*, 2017. DOI: 10.21105/joss.00456. [Online]. Available: http://www.theoj.org/joss-papers/joss.00456/10.21105.joss.00456.pdf.

[6] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 577–584. DOI: 10.1109/HUMANOIDS.2017.8246930.

[7] Ghanbari A., Rostami A., Noorani S.M.R.S., Fakhrabadi M.M.S, "Modeling and simulation of inchworm mode locomotion," 2008. DOI: 10.1007/978-3-540-88513-9_66. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-88513-9_66#citeas.

[8] A. Hornung, A. Dornbush, M. Likhachev, and M. Bennewitz, "Anytime search-based footstep planning with suboptimality bounds," in *2012 12th IEEE-RAS

International Conference on Humanoid Robots (Humanoids 2012), 2012, pp. 674–679. DOI: 10.1109/HUMANOIDS.2012.6651592.

[9] Jen Jen Chung, Ander J. Smith, Ray Skeele, Geoffrey A. Hollinger, "Risk-aware graph search with dynamic edge cost discovery," *The International Journal of Robotics Research*, 2018. DOI: 10.1177/0278364918781009. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0278364918781009.

[10] B. Jenett and K. Cheung, "Bill-e: Robotic platform for locomotion and manipulation of lightweight space structures," in *25th AIAA/AHS Adaptive Structures Conference*. DOI: 10.2514/6.2017-1876. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2017-1876. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2017-1876.

[11] P. H. Kourosh Nader Joose Rajamaki, "Rt-rrt*: A real-time path planning algorithm based on rrt*," 2015. DOI: 10.1145/2822013.2822036. [Online]. Available: https://users.aalto.fi/~hamalap5/FutureGameAnimation/p113-naderi.pdf.

[12] D. Pagano, "Uncertainty modelling and motion planning of an inchworm robot navigating in complex structural environments," 2018.

[13] R. K. Utkarsh Kumar Adrish Banerjee, "Collision avoiding decentralized sorting of robotic swarm," 2020. DOI: 10.1007/s10489-019-01602-5. [Online]. Available: https://link.springer.com/article/10.1007%2Fs10489-019-01602-5.

[14] *Vacuum/magnetic feet for a walking robot edit*. [Online]. Available: https://answers.gazebosim.org//question/14462/vacuummagnetic-feet-for-a-walking-robot/.

[15] S.-H. Yeo, I.-M. Chen, R. S. Senanayake, and P. S. Wong, "Design and development of a planar inchworm robot," in *Proceedings of the 17th IAARC/CIB/IEEE/IFAC/IFR International Symposium on Automation and Robotics in Construction*, M.-T. Wang, Ed., Taipei, Taiwan: International Association for Automation and Robotics in Construction (IAARC), Sep. 2000, pp. 1–6, ISBN: 9789570266986. DOI: 10.22260/ISARC2000/0075.