

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—This document is a model and instructions for L^AT_EX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.

I. INTRODUCTION

II. RELATED WORK

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 hope to be able to visualize the path created and see the robot taking the path by the end of our project. The reach goals we have thought about are moving with a block, creating a 3D walking gait, 3D path planning of going up walls, and implementing multiple robots into the system. Since these robots are building structures in the larger MQP our next step once we have finished our expected goals we would be moving with blocks which leads to the building of structures. The 3D walking gait would be next and have to take into account going up walls and moving around the corners of the blocks. If this gait is completed a 3D path planning algorithm could be attempted meaning adding much more complexity and destinations that could be up multiple levels of blocks. A different route we could take is adding multiple robots to the system. This would create a swarm situation where we have to account for collision detection with all robots.

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.

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 [ref both sim stuff], 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 (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*, 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 [insert RAGS ref] 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. 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.

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[ref long paper]. 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. [ref inchworm footstep paper (if we can find it)]

The final part of the motion planner will be the footstep planner. Inspiration will be taken from bipeds and quadrupeds [ref long paper], 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. Then it can use a graph search (like A*, WA*, ARA*, RAGS, etc) to find the optimal footsteps throughout the path [ref graph search footstep planner]. 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 [ref model predictive control]. 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. While the MQP this project is extending has done some gait work, it has yet to be implemented or tested

on a real or simulated robot. 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. Additionally, if time allows, we would like to test building a pyramid to compare our system to the previous MQP results. However, this will require us to achieve a significant portion of our reach goals, and will likely not be achieved in this project.

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 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.

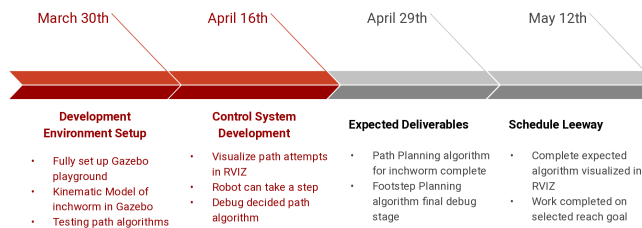


Fig. 1. Proposed schedule for the project