# Novel Weighted A* Footstep Planner for Quadruped Robot Over Rough Terrain

Katherine Greatwood  Supervisor: Dr. Ioannis Havoutis

DYNAMIC ROBOT SYSTEMS GROUP
OXFORD ROBOTICS INSTITUTE

UNIVERSITY OF OXFORD

## Motivation

For legged robots to be valuable in real-world scenarios, they must be able to cross complex terrain efficiently. Quadruped robots require four foothold locations per pose (as opposed to two for humanoid robots). Therefore, in environments which have a limited number of suitable footholds, quadruped footstep planners are often either impracticably **slow** or often **fail**.

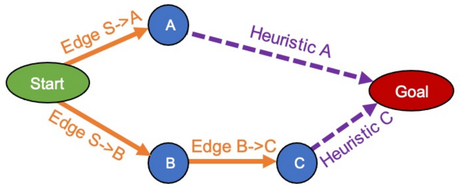
## Introduction

The goal of this project was to design a footstep planner for a four-legged robot which could **efficiently** and **successfully** (no falls) find a path across complex terrain.

Two versions of a new weighted A* footstep planner are outlined: an offline version, assuming prior knowledge of the terrain, and an online (no prior knowledge) version. The planner uses a planar representation of the terrain to quickly identify the areas of the environment containing viable footholds.

Both versions are successful despite having no information about the motion between poses. Additionally, this new quadruped footstep planner is **as fast** as existing biped (two leg) planners [1], despite requiring a number of footsteps which is an order of magnitude larger to cross a similar distance.

## Intro To Weighted A* Graph Search

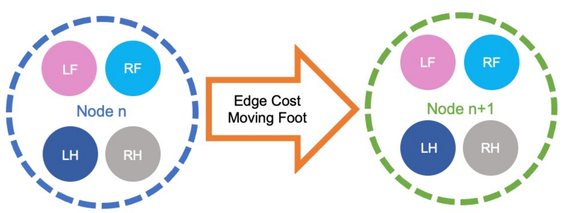



The total cost of each node is:

$$\text{Weight} \times \text{Cost-To-Go Heuristic (an Estimate)} + \sum \text{Edge costs from start to node}$$

A* graph search is based on a priority queue, ordering the not-yet-expanded noes by cost. The lowest cost node is expanded, meaning new nodes are created and connected to the graph. This repeats until a path to the goal node is found.
The weight biases the planner towards nodes which are closer to the goal, usually increasing efficiency.

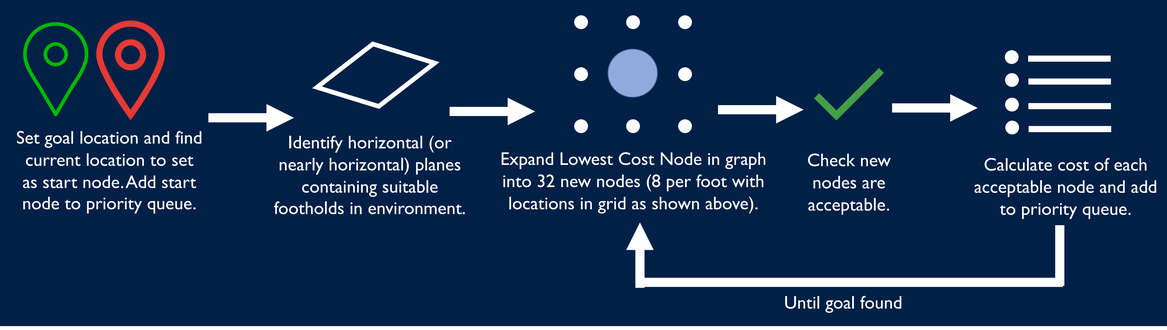
## Graph Setup




The nodes in the graph contain positions for all 4 feet. Each edge connects two nodes and has two properties:
1. The cost associated with the move from Node n to Node n+1.
2. The one foot which moves.

This planner is for a crawling gait (one foot moving at a time) but could be easily adapted for a trotting gait (two feet moving).

## Offline Planner




### Acceptability

Each node is deemed acceptable if it is within some defined constraints. This step eliminates nodes which are **infeasible** (the robot cannot physically reach the four foothold locations) as well as poses with are likely to be **unstable**. If a node is unacceptable is is deleted, removed from the graph, and cannot be included in the plan.

Checks on all 4 feet separately:
1. Close to collision.
   Check surrounding area for higher elevation than foot.
2. Close to edge.
   Check surrounding area for lower elevation than foot.
3. On plane (identified in environment).
   Efficient geometric solution for checking point in polygon.
4. Height change of moving foot from previous node.

Checks on relative locations of feet:
1. Feet too far apart.
2. Feet too close together.
3. Feet forming narrow parallelogram (unstable).
4. Pitch angle too high.
   Use foot locations to estimate angle of inclination from horizontal of line from front to back of robot body.
5. RMS of heights of feet.

### Cost of Edge

The terms in the cost edge mirror the acceptability criteria. The **quality** of the footholds of the child node (node n+1 as shown on right) of the edge is included in the edge cost.

$$\frac{1}{5} \times \text{Distance moved in XY plane} + 2 \times \text{Change in elevation} + \frac{1}{15} \times \text{RMS feet height} + \frac{1}{5} \times \text{Relative foot location}$$

$$+ \frac{1}{6} \times \text{How close to edge of plane} + \frac{1}{5} \times \text{Height diff between left and right feet} + \frac{1}{15} \times \text{Pitch angle} + \text{Static cost per step}$$

### Cost-To-Go Heuristic

The heuristic is an estimate of the cost from the current node to the goal and in this planner the weight (in the total cost calculation) is built-in.

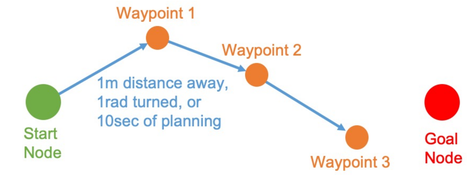
$$60 \times \text{Distance to goal in XY plane} + d \times \text{Diff in yaw angle from goal (turn on spot)} + \frac{15}{2} \times \text{Change in elevation} + \text{Estimated cost for number of steps}$$

$d$ is inversely proportional to the distance to the goal (in 3D) which encourages the robot to move to the goal before turning to match the goal yaw angle.

## Online Planner

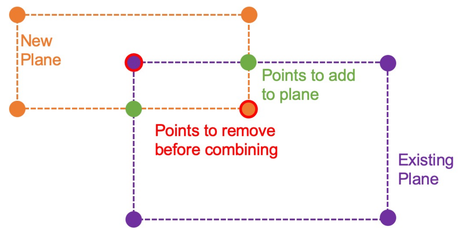
### Waypoints

The offline planner is adapted into an online planner, which has no prior information about the environment. It works out one segment of the plan to the goal at a time. It finds a plan to a waypoint (defined below), then the robot starts moving. When the robot is close to the waypoint, the waypoint is set as the new start node and the planner begins again to find a new waypoint.




### Planes

As the robot moves around the environment, it "discovers" new planes and expands existing planes. It is critical to combine overlapping planes to increase the efficiency of the planner when checking if a foothold is on a plane (the fewer planes to check the better).

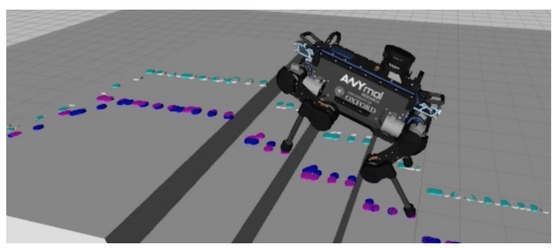



### Change in Environment

The online planner is a real-time planner so should be able to react to a change in the environment (such as a moving obstacle). All nodes which have been planned (but have not been reached yet) are checked continuously while the robot moves. If the elevation of any of their foot locations has changed, the robot stops and replans from its current location.

## Results

The planner is validated in simulated environments with an ANYmal quadruped robot.

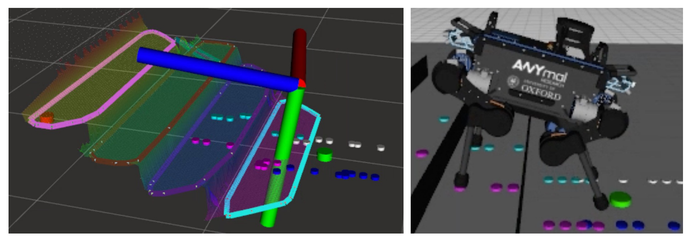


The offline planner is shown on the left successfully traversing a set of stairs (up and down). The robot remains stable and identifies high quality footholds throughout the entire plan.

It is **very efficient**, taking a similar amount of time to plan with many more steps in the plan required to cover the same distance across terrain with a limited number of suitable footholds.

| *Quadruped Planner* | | *Biped Planner [1]* |
|---|---|---|
| 0.19 sec to plan per meter | < | 0.214 sec to plan per meter |
| 0.004 sec to plan per step | < | 0.078 sec to plan per step |

In a variety of environments, the quadruped footstep planner is more efficient (both in terms of time to plan per meter distance and time to plan per step) than similar biped footstep planners [1].



The online planner is also successful, identifying planes as the robot moves and planning and moving seamlessly (no pause while planning next step). Again, the planner was tested on a variety of environments and successfully found a plan to the goal.

[1] Griffin, R. J., et al. (19 July 2019). Footstep Planning for Autonomous Walking Over Rough Terrain. *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. DOI:10.1109/humanoids43949.2019.9035046.