

**INFORMATIVE FOOTFALL PLANNING:
TERRAIN CHARACTERISATION THROUGH
TORQUE FEEDBACK IN LEGGED ROBOTS**

ELANGO PRAVEEN

**INNOVATION AND DESIGN PROGRAMME
NATIONAL UNIVERSITY OF SINGAPORE**

2023

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in this thesis.



ELANGO PRAVEEN

29 Oct 2023

Table of Contents

Summary.....	4
1 Acknowledgements	6
2 Introduction and Related Work	7
3 Project Objective	9
4 Proposed Methodology.....	10
5 Perception Stack.....	12
5.2 End Effector State Observer	14
5.3 End Effector Finite State Machine (EFSM).....	15
5.3.1 Ground Reaction Forces (GRF).....	15
5.3.2 Alternating Tripod Gait	15
5.3.3 Static Contact Experiments.....	16
5.3.4 Dynamic Contact Experiments.....	19
5.3.5 Decomposition of the Stance Phase.....	21
5.3.6 Terrain Contact Model.....	22
5.3.7 EFSM Structure	23
5.3.8 Contact State Estimation – 3D Vision	25
5.3.9 Contact State Estimation – GRF	27
5.3.10 Contact State Estimation – Binary Contact Sensor	27
5.3.11 Check, Transition, and Action.....	29
5.4 Stiffness Extractor	31
5.4.1 Stiffness Extraction Experiment Setup.....	31
5.4.2 Stiffness Extraction Acceptance Checks.....	32
5.4.3 Regression for Stiffness Gradient Calculation	35
5.4.4 Stiffness Scale Construction	37
5.4.5 Stiffness Visualisation Experiment Setup	38

6	Mapping Stack	41
6.1	Foothold Measurements Representation	41
6.2	Gaussian Process Preliminaries	42
6.3	GP Terrain Stiffness Map Experiment Setup	44
7	Planning Stack.....	48
7.1	Informative Path Planning.....	48
7.2	Adaptive CMA-ES Planner Preliminaries	48
7.3	Encourage Exploration with Enforced Budget.....	49
7.4	Shortfalls of the CMA-ES Planner	52
7.5	Tripod Gait Controller	53
8	Conclusion and Future Work.....	56
8.1	Research Design Matrix	56
8.2	Future Work	57
9	References	59
	Appendix.....	60

Summary

Autonomous legged robots are often tasked with navigating challenging terrains, such as forest areas, rocky areas, and underwater sandy surfaces. In vision-denied scenarios and cases where the robot is locomoting on such challenging and monochromatic terrain, there is a need for the robot to physically interact with the environment to obtain important and useful information to navigate successfully and effectively. In particular, this work focuses on applications where a legged robot is tasked with locomoting over a malleable ground of variable stiffness, where areas of higher stiffnesses may indicate the presence of objects of interest buried underground. We propose an approach for a legged robot system to estimate terrain stiffnesses from the proprioceptive feedback of legged robots, and to use it to build a stiffness belief map represented by a Gaussian Process, and finally how the map can be used for the robot to perform informative path planning to explore the potential areas of interest.

1 Acknowledgements

Firstly, I would like to thank my supervisor Dr Guillaume Sartoretti for providing me with the opportunity to work on this very interesting problem, and also for providing us with the appropriate guidance for this project. I would also like to thank my project mate Mukund whom I worked with on this project extensively, and Tarasvin for his mechanical design contributions to the project. I am also grateful for having worked with members of the MARMoT Lab, namely Sun Ge and Yaswanth, for making this project possible.

2 Introduction and Related Work

Humans and animals are adept at using haptic information from their feet to characterise their environments through contact to perform various useful tasks, including gait adaptation and navigation on rough terrain. Legged robots, aiming to achieve human-level proprioception for such tasks, need the ability to understand and model their interactions with the environment. This could include specific terrain information such as surface friction and stiffness, which experience significant variation from terrain to terrain in unstructured environments. While vision-based strategies such as point cloud and object detection can identify the geometric properties of such terrains, the robot would need to directly interact with the terrain to reason about its physical properties.

Recent work has focused on equipping legged robots with sensorised feet to gather such physical information from the terrain, allowing for its classification using such unique characteristics. Legged robots find suitability in applications such as search and rescue missions across unstructured terrains, owing to their exceptional maneuverability and inherent stability [1]. These robots leverage their extensive degrees of freedom to directly engage with the environment while simultaneously maintaining a firm and secure posture. As evidenced in [2], legged robots have effectively employed direct contact techniques, including a scratching motion of the leg against the floor, to ascertain the frictional properties of surfaces like cement, which is a task challenging to accomplish solely relying on visual methods. The authors in [3] have further laid the groundwork for understanding the correlation between the force experienced at the leg's endpoint and the resulting endpoint displacement. This foundational knowledge has been harnessed in [4] for unsupervised learning applications, where geometric terrain features and haptic feedback are combined to assess the resilience of footholds.

The incorporation of probing capabilities proves particularly advantageous when seamlessly integrated with the fundamental tasks that robots must undertake, including locomotion, navigation, mapping, and manipulation. Additionally, tactile localization methods, such as Sequential Monte Carlo approaches, have been successfully utilized, relying exclusively on foot contact states and joint kinematics, to

precisely localize legged robots in challenging terrains, building upon prior geometric environment maps [5]. Legged robots have played a pivotal role in this context by constructing comprehensive terrain maps that encompass both geometric attributes and physical characteristics, including terrain collapsibility [6]. These maps facilitate the computation of traversability scores, aiding in the determination of whether the robot should proceed across these regions or consider alternative routes.

In particular, we are interested in applications where a legged robot has haptic sensors attached at its end effectors, allowing for it to effectively traverse deformable terrains such as loose sand to locate buried objects with higher stiffnesses. Such applications are useful in augmenting or automating currently human-only operations in dangerous search and rescue missions, demining, and underwater seabed exploration. In this work, we propose an approach for a legged robot system to estimate terrain stiffnesses from the proprioceptive feedback of legged robots, and to use it to build a stiffness belief map represented by a Gaussian Process, and finally how the map can be used for the robot to perform informative path planning to explore the potential areas of interest.

3 Project Objective

The key objective of this work is to effectively locate potential objects of interest such as explosive ordnances that may be hidden underneath challenging terrain environments. This task-level objective can be decomposed into the following three sub-objectives:

1. **Terrain Stiffness Estimation:** How can we estimate terrain stiffnesses using proprioceptive information from the robot's feet?
2. **Environment Belief Representation:** How can we use this stiffness information to build and update a stiffness belief map of the robot's locomoted environment?
3. **Informative Path Planning:** How can we use this map to inform the robot's path planning to explore potential areas of interest?

In the successive sections, we outline our proposed approach to achieve these 3 sub-objectives in conjunction with each other to realise the overall key objective of the project.

4 Proposed Methodology

In this section, I detail our approach to achieve our task-level project objective. To this end, we propose our robotic system pipeline - TerCAP: Terrain Characterisation Pipeline, which was deployed on the 6-legged HEBI Daisy Robot with 18 joint actuators. The software stack of the pipeline was developed using the Robot Operating System (ROS) middleware suite in Python and C++ given its modularity for robot applications and inter-process communication and coordination across different components of the robotic system.

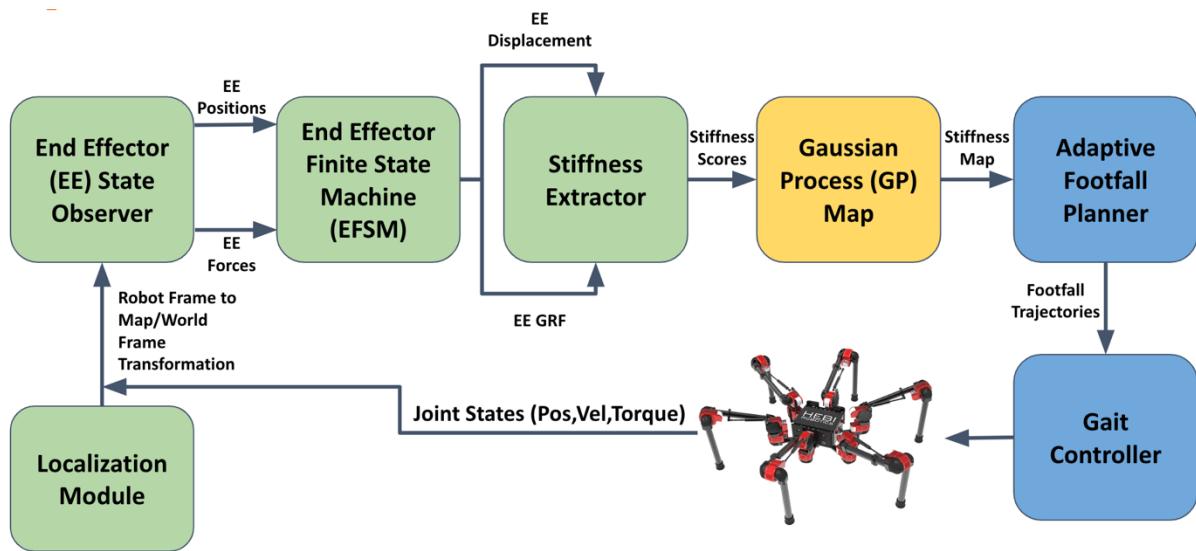


Figure 1: The Terrain Characterisation Pipeline (TerCAP)

The TerCAP pipeline can be deconstructed into the following 3 subsystems as seen in Figure 1:

- 1. Perception (Green):** The role of the perception subsystem is to sense and interpret the robot's observations of its interactions with the surrounding environment. This subsystem consists of four components – the localisation module, the end effector state observer, the end effector finite state machine (EFSM), and the stiffness extractor. The purpose of the localisation module is to localise the robot in the environment. The end-effector state observer correctly processes the feet proprioceptive observations (or footholds) from the joint space to the cartesian space for successive contact and stiffness estimation in the pipeline. The EFSM is used to track the phases of the hexapod robot's tripod gait cycle to determine the time of feet contact with the environment surface, and the magnitudes of the ground reaction forces and displacement at the point of contact. Finally, the stiffness extractor serves the purpose of estimating the terrain stiffness at the point of contact. We explain the methodology, implementation, experiments, and results involved in developing the perception stack in Chapter 5.
- 2. Mapping (Yellow):** The role of the mapping subsystem is to process the foothold measurements to represent and regularly update the belief of the robot's environment in the form of a Gaussian Process terrain stiffness map. We explain the methodology, implementation, experiments, and results involved in developing the mapping stack in Chapter 6.
- 3. Planning (Blue):** The role of the planning subsystem is to plan foothold trajectories for the robot to execute in its environment. The purpose of the adaptive footfall planner is to use the stiffness belief map to adaptively plan the robot's path in the environment to encourage its exploration of more uncertain regions. The planner outputs footfall trajectories, which are sent to the tripod gait controller of the robot to execute them. We explain the methodology, implementation, experiments, and results involved in developing the mapping stack in Chapter 7.

5 Perception Stack

5.1 Localisation Module

The purpose of the localisation module is to enable the robot to determine its own position in a global reference frame to navigate accurately and to create an accurate map of its environment. The robot's observations including its base orientation, joint positions, velocities, and force observations that are sensed by its joint encoders and force-torque sensors are obtained with respect to its body frame, which are in its own coordinate system. The transformation of position observations to the world frame is necessary for mapping as the collected sensor data needs to be represented in a global context to build a consistent and coherent map of the robot's surroundings.

To construct the global map frame, the Intel RealSense T265 Pose Tracking Camera was used. As seen in Figure 2, the camera frame is represented by $\{P\}$, the global map frame is represented by $\{M\}$, the robot base-link frame is represented by $\{B\}$ and the odometry frame, which is the specific coordinate frame that represents the estimated pose of the robot over time, is represented by $\{O\}$.

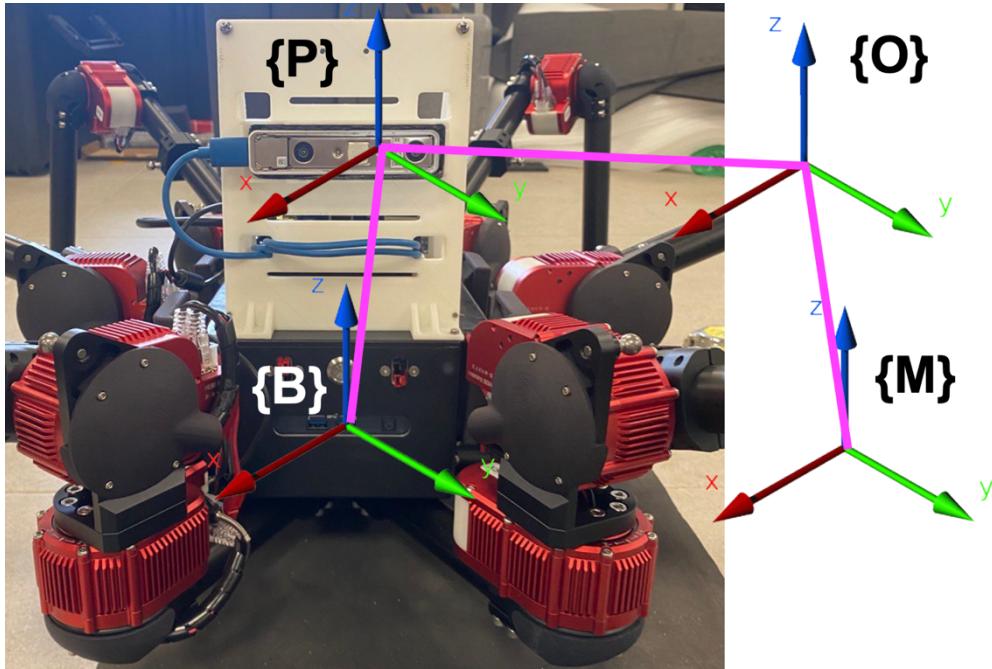


Figure 2: Transformation Coordinate System

In the same Figure 2, the transforms between the frames are represented by the pink solid lines. The transform of $\{B\}$ expressed in $\{P\}$, ${}^P T_B$, is a static transform that represents the relative pose of the camera with respect to the base link of the robot, since the camera is fixed on the robot. The transform of $\{O\}$ expressed in $\{M\}$, ${}^M T_O$, is also a static transform that represents the relative pose of the odometry frame expressed in the global map frame. The transform of $\{P\}$ expressed in $\{O\}$, ${}^O T_P$, is a dynamic transform that is the tracked pose of the camera in the odometry frame. The matrix multiplication of these transforms yields the 4×4 SE(3) homogenous transformation matrix ${}^M T_B$, which represents the pose of the robot expressed in the global map frame, is as follows:

$${}^M T_B = {}^M T_O {}^O T_P {}^P T_B \quad (1)$$

This transformation matrix can be applied to positions in the base-link frame to express them in the global map frame.

5.2 End Effector State Observer

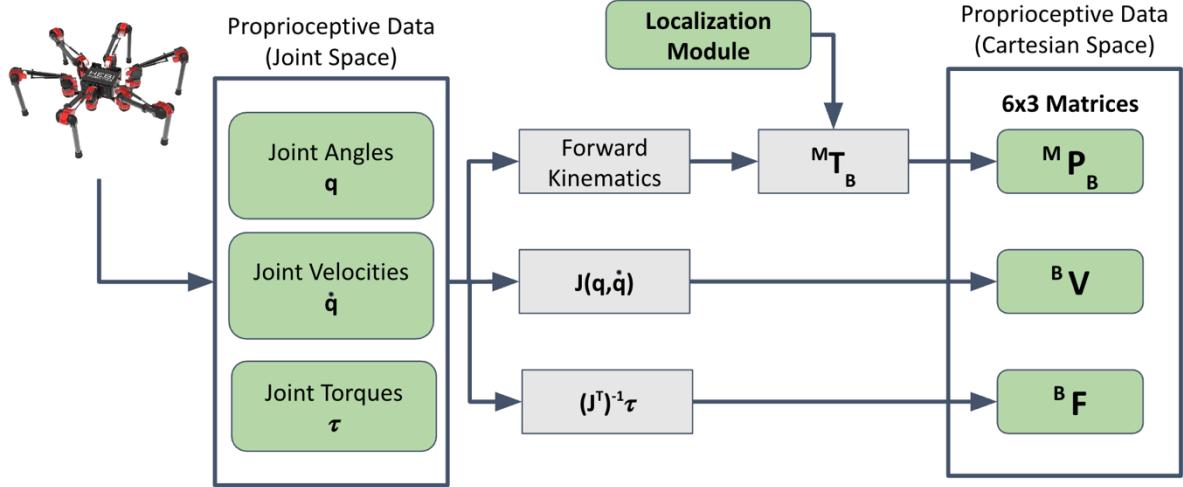


Figure 3: Architecture of the End Effector State Observer

The role of the end effector state observer is to poll for proprioceptive information in the joint space from the robot through the HEBI API, namely joint angles q , velocities \dot{q} , and torques τ , where $q, \dot{q}, \tau \in \mathbb{R}^{18 \times 1}$, and convert them to the Cartesian space. Using forward kinematics, we can convert the 18×1 joint angles q to 6×3 end-effector Cartesian positions in the base frame ${}^B P$. We can then apply the transform ${}^M T_B$ that we obtain from the localisation module to ${}^B P$ to obtain ${}^M P_B$, which are base-link end-effector positions expressed in the map frame. This information is useful when we want to obtain the displacement of the end-effector inside deformable terrain upon contact. We also obtain the end-effector ground reaction forces (GRFs) in the base frame ${}^B F$ as per the following equation:

$${}^B F = -(J^T)^{-1} \tau \quad (2)$$

where $(J^T)^{-1}$ is the inverse of the transpose of the Jacobian matrix, which is a mapping between forces applied between forces applied at the end effector and the torques applied at the robot's joints. Both ${}^B F$ and ${}^B P$ are published out to the EFSM using the ROS Message Publisher protocol.

5.3 End Effector Finite State Machine (EFSM)

The purpose of the end effector state machine (EFSM) is to obtain the time of contact between the end effector and the terrain to extract the normal ground reaction forces that are used for stiffness estimation. In this section, I explain the relevance of Ground Reaction Forces (GRF), and the experiments conducted to analyse and interpret GRFs, the structure of the EFSM, and how GRFs are extracted as accurately as possible using the EFSM.

5.3.1 Ground Reaction Forces (GRF)

When the end effector of the robot contacts a particular terrain, the feet exerts a normal force on the terrain plane which exerts an opposing but equal force that is experienced by the end effector. This force is referred to as GRF and is experienced by all the end effectors of the robot during locomotion. The force profile of the GRF experienced during the window of contact is expected to be vary for terrains with differing stiffnesses, since varying terrain stiffnesses directly influence the weight acceptance of the robot under certain assumptions as explained in Section 5.3.6.

5.3.2 Alternating Tripod Gait

In this work, we use the alternating tripod gait for the hexapod robot for locomotion and navigation tasks. The alternating tripod gait is a walking pattern commonly used by hexapod robots as it provides stability and efficient locomotion. In a hexapod robot, the legs are typically arranged in two sets of three legs each. One set of three legs is referred to as a tripod. In Figure 3, we observe that legs 0, 3, and 4 form a single tripod group as reflected by the green triangle. We shall refer to this group as tripod group 0. We also observe that legs 1, 2, and 5 form another tripod group as reflected by the green triangle. We shall refer to this group as tripod group 1.

The alternating tripod gait involves two phases: the swing phase and the stance phase. During the swing phase, the legs in tripod group 0 will be lifted off the ground and moved forward or backward. At this point of time, the legs in tripod group 1 will be in

the stance phase, where they would have contacted the ground and are supporting the robot's weight.

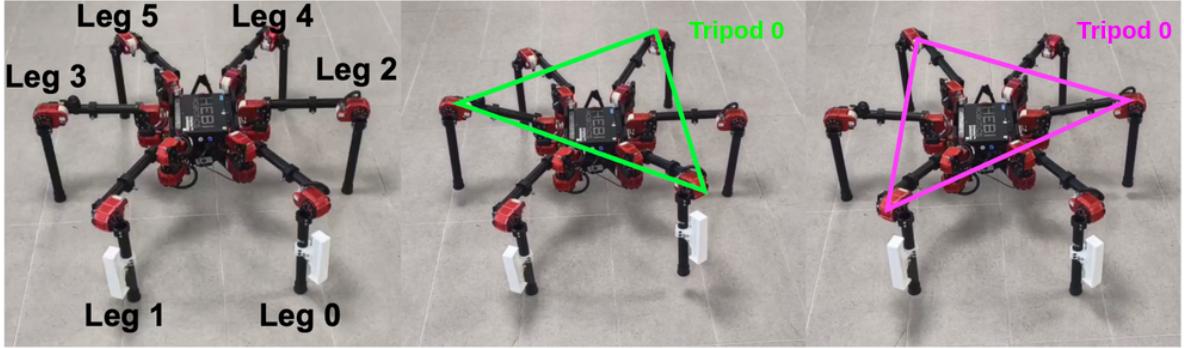


Figure 4: The alternating tripod gait – (Left) The order of the legs, (Middle) Tripod Group 0, (Right) Tripod Group 1

5.3.3 Static Contact Experiments

Our objective was to enhance our comprehension of the relation between the GRFs encountered by the end effectors and the contact dynamics involving the terrain. Specifically, we wanted to test the following hypothesis:

Hypothesis 1: *The sum of the GRFs experienced by all legs that are contacting the ground should equal the total weight of the robot.*

To this end, we devised a set of three static contact experiments where the end effectors would experience contact with the ground while in a state of rest.

Experiment 1: All 6 legs on the ground

In the first experiment, all 6 of the robot legs were contacting with the floor. The GRF magnitude values in the z direction for each leg were recorded for three iterations and were averaged. The sum of the mean end effector GRFs equated to 143.28 N as seen in Table 1.

Table 1: End Effector GRFs when all 6 legs are on the ground

End Effector GRF / N when all 6 legs are on the floor (at rest)					
Conducted on 20-07-23					
Experiment	0	1	2	MEAN	VAR
Leg 0	12.54	12.29	12.59	12.47	0.03
Leg 1	31.68	31.54	32.73	31.98	0.42
Leg 2	31.63	32.11	30.3	31.35	0.88
Leg 3	14.1	15.44	14.64	14.73	0.45
Leg 4	21.79	22.41	23.16	22.45	0.47
Leg 5	30.77	29.82	30.29	30.29	0.23
ALL LEGS	142.51	143.61	143.71	143.28	0.44
TRIPOD 0	48.43	50.14	50.39	49.65	1.14
TRIPOD 1	94.08	93.47	93.32	93.62	0.16

Experiment 2: Tripod 0 in Swing Phase and Tripod 1 in Stance Phase

In the second experiment, the 3 legs in the Tripod 0 group were raised and were in the swing phase while the 3 legs in the Tripod 1 group were in stance phase and were supporting the whole weight of the robot. The sum of the mean GRFs of the legs in the z-direction in Tripod 1 group were 231.38 N as seen in Table 2.

Table 2: End Effector GRFs of Tripod 1 group in stance phase

End Effector GRF / N when Tripod 0 is raised. Look at Tripod 1 GRFs					
Conducted on 20-07-23					
Experiment	0	1	2	MEAN	VAR
Leg 0	-9.08	-9.33	-9.42	-9.28	0.03
Leg 1	56.23	57.08	57.7	57	0.54
Leg 2	108.83	111.46	107.27	109.19	4.48
Leg 3	-7.3	-7.45	-7.36	-7.37	0.01
Leg 4	-9.45	-9.6	-9.66	-9.57	0.01
Leg 5	64.07	65.43	66.07	65.19	1.04
Tripod 1	229.13	233.97	231.04	231.38	5.94

Experiment 3: Tripod 1 in Swing Phase and Tripod 0 in Stance Phase

In the second experiment, the 3 legs in the Tripod 1 group were raised and were in the swing phase while the 3 legs in the Tripod 0 group were in stance phase and were supporting the whole weight of the robot. The sum of the mean GRFs of the legs in the z-direction in Tripod 0 group were 202.72 N as seen in Table 3.

Table 3: End Effector GRFs of Tripod 0 group in stance phase

End Effector GRF / N when Tripod 1 is raised. Look at Tripod 0 GRFs					
Conducted on 20-07-23					
Experiment	0	1	2	MEAN	VAR
Leg 0	56.8	57.34	57.01	57.05	0.07
Leg 1	-6.67	-6.66	-6.72	-6.68	0
Leg 2	-9.91	-10.15	-10.2	-10.09	0.02
Leg 3	69.81	72	71.98	71.26	1.58
Leg 4	74.25	74.53	74.44	74.41	0.02
Leg 5	-8.35	-8.42	-8.45	-8.41	0
Tripod 0	200.86	203.87	203.43	202.72	2.64

Conclusion of the Experiments

The total weight of the robot is 21 kg, which can be approximated to 206.01 N as per $W = mg$ where $g = 9.81 \text{ N kg}^{-1}$. While the sum of the GRFs of the grounded legs in experiment 3 were the closest to this value, the sum of the GRFs of the grounded legs in experiment 1 and 2 are not close to this value. Moreover, the evidence of the variations in the sum of the GRFs of all the grounded legs suggests that it is not accurate to consider the sum of the GRFs of the whole system when evaluating terrain contact. However, we can observe that the GRFs experienced by the end effector of a single leg varies across all the experiments. For example, when considering Leg 1 (which is in Tripod 1 group), its encountered mean GRF is 31.98 N, 57 N, and -6.68 N in experiments 1, 2, and 3 respectively. The observations are consistent with the expectations: the GRF is highest when only Tripod 1 is in stance phase, lowest when only Tripod 0 is in stance phase, and the magnitude is in the middle of these two values when all 6 legs are grounded and are bearing the weight of the robot.

A possible explanation for this difference in inter-leg and intra-leg GRF measurements could be that series elastic actuators are used in the joints of the robot, and their elastic elements could experience less displacement when more legs are supporting the weight of the robot, resulting in a lower magnitude of joint torques being measured. Hence, it would be reasonable to analyse end effector GRFs only with respect to each leg individually.

5.3.4 Dynamic Contact Experiments

The terrain contact dynamics between the end effectors and the terrain may differ when the robot body is in motion. Adopting the method of intra-leg GRF measurements as outlined in Section 5.3.3, we devised a set of 2 experiments to measure normal GRFs when the robot body was in locomotion with an alternating tripod gait. In experiment 1, the robot was walking on the floor terrain while in experiment 2, the robot was walking on more deformable floor terrain. The number of steps walked by the robot in both experiments is the same. The objective of this set of experiments is to observe the change in feet GRFs of the robot when it is in motion and to test the following hypothesis:

Hypothesis 2: *There could be differences in the force profiles measured when the robot is locomoting on terrain with different stiffnesses.*

Experiment 1: Robot locomotion on floor terrain

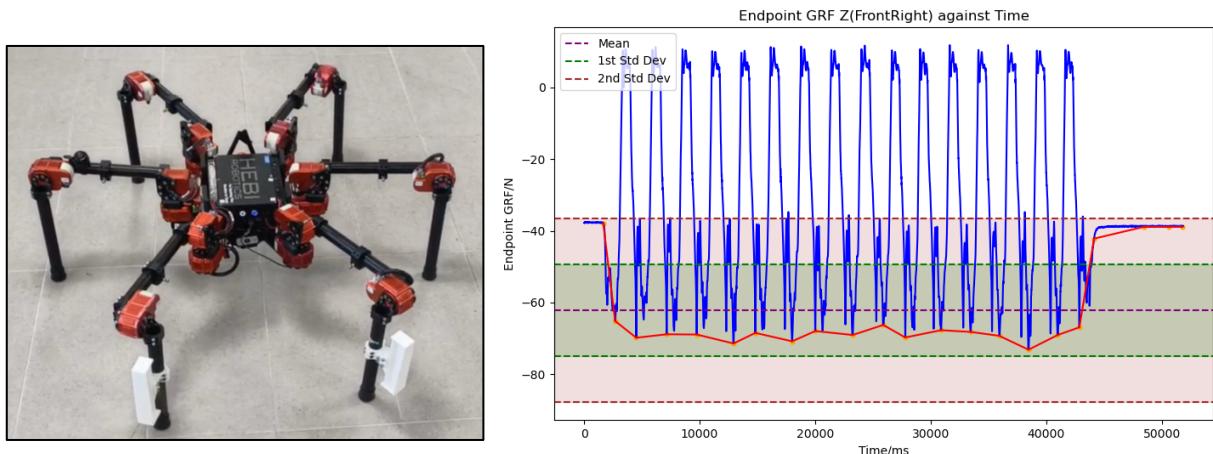


Figure 5: (Left) Robot locomotion on stiff floor terrain, (Right) End effector GRF of the front right leg recorded over time during the locomotion experiment.

In Figure 5, we can observe that the endpoint GRF changes periodically and that the peaks represent the swing phase and the troughs represent the stance phase. It can also be observed that the change in GRF when transitioning from the swing to stance phase is erratic, suggesting that the end effector could be bouncing off the floor surface before completely transitioning to the stance phase.

Experiment 2: Robot locomotion on foam terrain

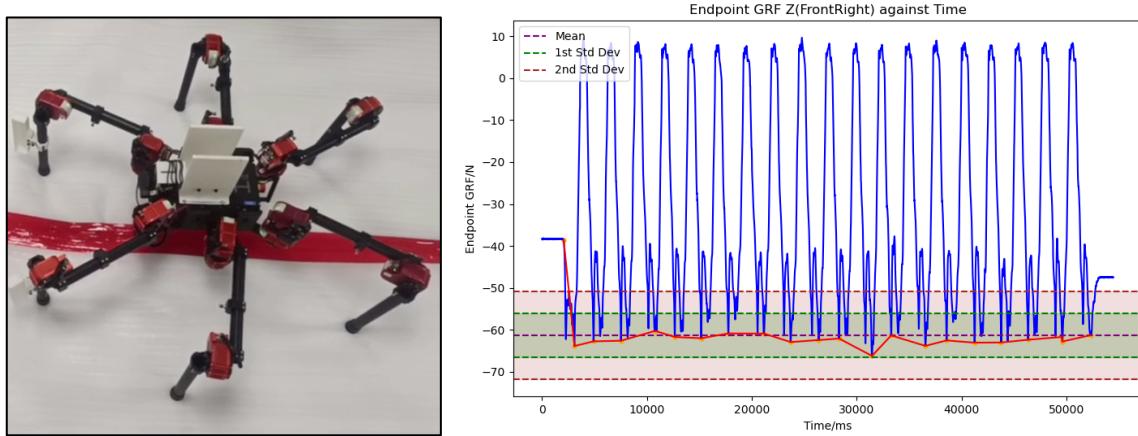


Figure 6: (Left) Robot locomotion on deformable foam terrain, (Right) End effector GRF of the front right leg recorded over time during the locomotion experiment.

In Figure 6, it can be observed that the periodic shape of the graph and the peak and trough values are similar to those observed in experiment 1. This is not unexpected as the magnitude of the weight of the robot that is accepted by the surfaces is the same. However, comparing both the graphs does not yield any meaningful conclusions since the transition period between the swing and stance phases in both experiments is transient. Specific information such as the time of first contact, the phase of weight acceptance, and the point at which the end effector leaves the ground to swing phase needs to be extracted in this transient phase to truly analyse the change in GRF values. Hence, there is a need to decompose and expand the stance phase further as outlined in Section 5.3.5.

5.3.5 Decomposition of the Stance Phase

During the entirety of the stance phase, there is considerable variation in the GRF experienced by the end effector. Related work in [7] shows that the stance phase can be decomposed into the Land (LND), Weight Acceptance State (WAS), and the Takeoff (TKO) state. This decomposition is essential as we need to find the GRF experienced by the robot when the terrain experiences the maximum displacement. The decomposition and the sequence of the states can be visualised in Figure 7 below.

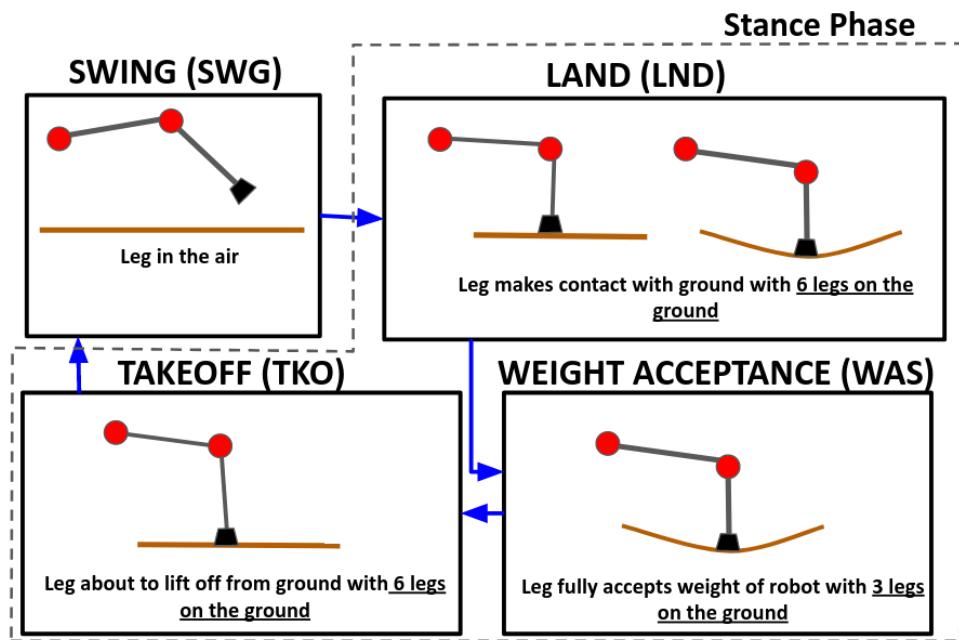


Figure 7: The Decomposition of the Stance Phase

The LND phase is when the end effector first contacts the ground from the SWG phase. In this phase, all the 6 legs of the robot are also grounded. The end effector then progresses to the WAS phase where the end effector along with the other end effectors in the same tripod group fully support the weight of the robot. In this phase, the legs in the other tripod group are in the SWG phase. Finally, the leg then transitions to the TKO phase where it is about to lift off from the ground. In this phase, all the 6 legs are on the ground. The leg then exits the stance phase and transitions into the SWG phase. These four states will form the sequence for the EFSM for each leg, which will be further outlined in Section 5.3.7. The choice of the transition functions, and how the decomposed states are used to estimate the terrain stiffness will be outlined in Section 5.4.

5.3.6 Terrain Contact Model

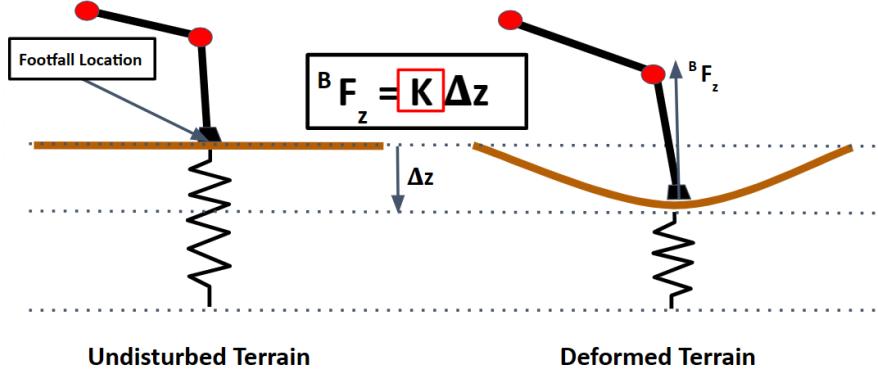


Figure 8: Terrain Contact Model for Undisturbed and Deformed Terrain

When the end effector transitions from the LND to the WAS phase during contact, it exerts a normal force on the surface of the terrain. To achieve the sub-objective of stiffness estimation, we would need to model this terrain contact interaction. For undisturbed terrains that typically have higher stiffnesses, the force impact by ${}^B F_z$, will cause zero to minimal change in terrain displacement. For deformed terrains, we model the terrain as a linear spring system that will experience displacement Δz at the point of contact upon experiencing the normal contact force ${}^B F_z$ exerted by the leg. This allows us to estimate the stiffness parameter k as per the equation ${}^B F_z = k\Delta z$.

5.3.7 EFSM Structure

Each end effector has an assigned state at a particular point of time in the state machine. When the end effector first contacts the terrain surface, it is in LND, and the state machine then transitions states for the leg to WAS, TKO, SWG, and then back to LND. These transitions are performed for all 6 legs of the robot, depending on the current state of the leg end effector. The structure and flow of the EFSM states are shown in Figure 9.

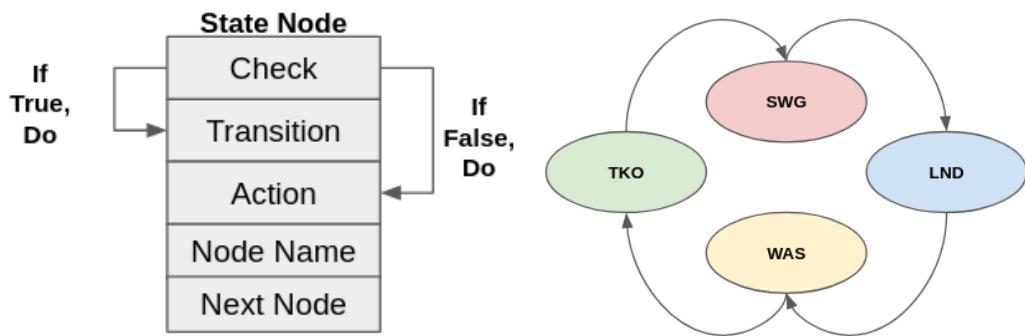


Figure 9: State Machine Logic and Sequence

A check assesses the transition condition during every iteration and performs the transition to the next state if the transition condition is met. Else, the state performs a particular action that is assigned to that state. The EFSM operates at a high frequency of 250 Hz to ensure that the transitions between the states happen precisely to prevent any possible missed transitions. However, even if such transitions are missed in real-life, the EFSM is sufficiently robust to self-recover at the next cycle given that the state transitions are periodic.

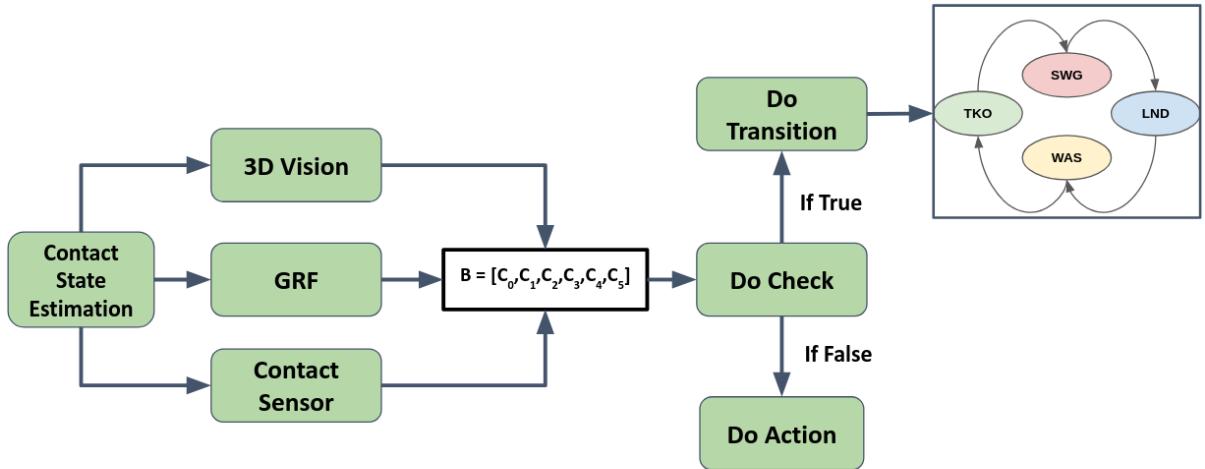


Figure 10: EFSM Structure

The EFSM depends on accurate contact state estimation of the end effector with the terrain surface for the formulation of contact Boolean C_i for each leg i , which is used to form the binary array B as shown in the following equations:

$$C_i = \begin{cases} 1 & \text{if contact with terrain}, i = \{0,1,2,3,4,5\} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$B = [C_0, C_1, C_2, C_3, C_4, C_5] \quad (4)$$

The binary array B is computed at the start of each EFSM step to inform the Check module if transition conditions have been satisfied. There are 3 proposed approaches for the contact state estimation in this work: 3D Vision, GRF, and Contact Sensor.

5.3.8 Contact State Estimation – 3D Vision

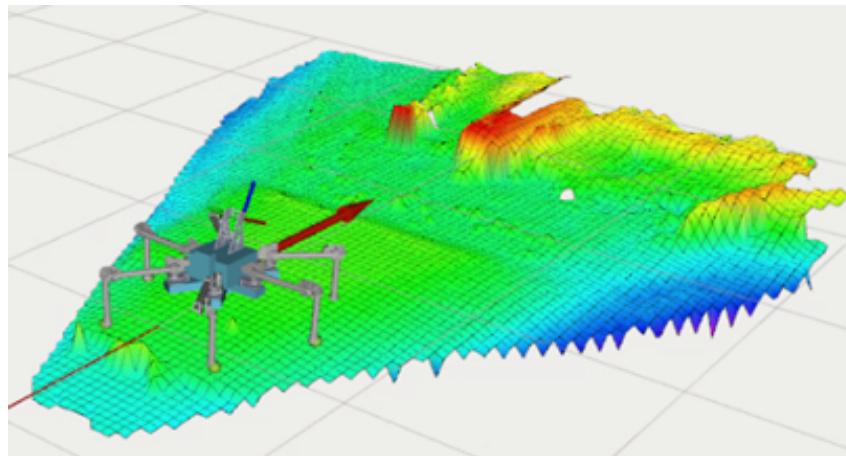


Figure 11: 3D Elevation Map

3D vision depth cameras enable the creation of 3D elevation maps in the robot's environment. Here, we utilize the Robot-Centric Elevation Mapping software suite [8] with Intel RealSense D435 Depth Camera data to build a 4x4 elevation map. When providing 2D base frame positions (x, y) to the elevation map, it yields the elevation height in the map frame. Figure 12 illustrates the variation in end-effector elevations in undulating terrains.

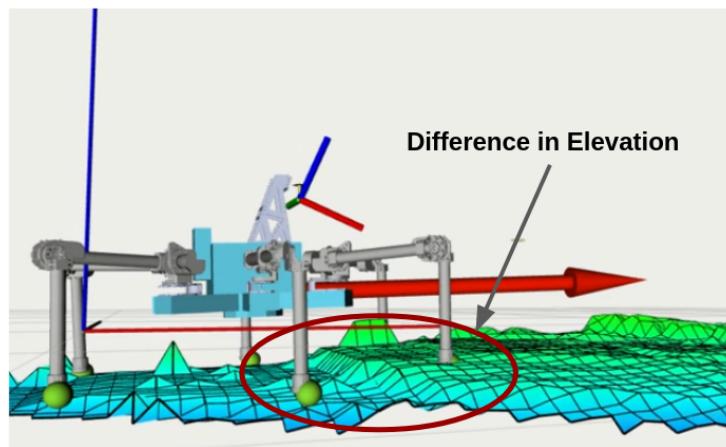


Figure 12: Elevation differences between end-effectors

The elevation map can be used to compute C_i as per:

$$C_i = \begin{cases} 1 & \text{if } ({}^M P_{z,i} - {}^M z_e) < d_z, i = \{0,1,2,3,4,5\} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where ${}^M P_{z,i}$ is the position of leg i in the z-axis and ${}^M z_e$ is the height of the elevation map at $({}^M P_{x,i}, {}^M P_{y,i})$, and d_z is an experimentally determined threshold. If the difference in the z-position of the leg and the height of the elevation map at that point is less than this threshold, then we consider the end effector to have contacted the terrain.

However, the use of 3D vision-based methods have significant shortfalls. Depth camera sensors that are fixed on the robot provide limited fields of view (FoV), which results in blind spots as seen in Figure 13. Therefore, omnidirectional locomotion of the hexapod is constrained as the robot is forced to walk in the direction where the camera is mounted. Replacing the camera with omnidirectional LiDARs also result in challenges which are outlined in the Appendix.

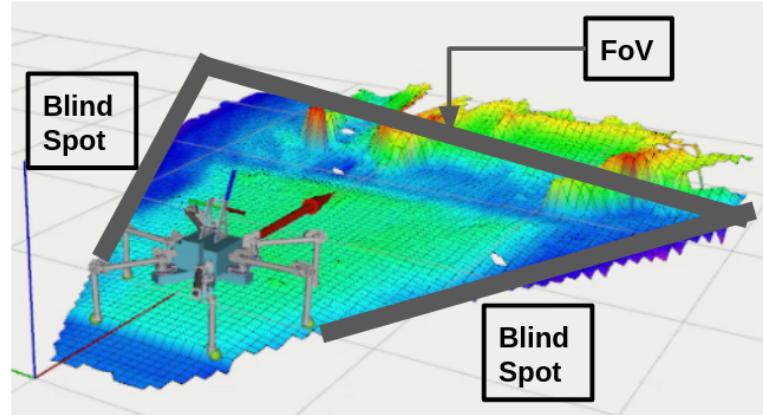


Figure 13: Blind spots arising from fixed field of view of depth sensor

5.3.9 Contact State Estimation – GRF

To sidestep the field-of-view and blind spot limitations caused by vision-based approaches, we propose to use proprioceptive feedback using GRFs for contact state estimation, specifically to compute C_i and B .

$$C_i = \begin{cases} 1 & \text{if } {}^B F_{z,i} > f_c \\ 0 & \text{otherwise} \end{cases}, i = \{0,1,2,3,4,5\} \quad (6)$$

${}^B F_{z,i}$ is the normal contact GRF experienced by the end-effector of leg i in the base frame and f_c was experimentally determined to be 0. This is because negative GRFs suggest that the end effector is not experiencing any contact yet and positive GRFs suggest that the end effector is already contacting the terrain.

5.3.10 Contact State Estimation – Binary Contact Sensor

A binary contact sensor was also designed and attached to each end effector of the 6 legs as seen in Figure 15, and C_i was computed as follows:

$$C_i = \begin{cases} 1 & \text{if } R_i > r_{c,i} \\ 0 & \text{otherwise} \end{cases}, i = \{0,1,2,3,4,5\} \quad (7)$$

where R_i represents the analog signal value output from the contact sensor, and $r_{c,i}$ is the experimentally determined threshold to determine contact.

The main drawback of pure contact-based methods such as GRF and binary contact sensor for contact state estimation is that it is challenging to estimate surface normals at footfall locations unlike vision-based methods as seen in Figure 15. This is useful in scenarios where local approximations to the elevation in the terrain require a resolution of the normal contact forces. However, this could be resolved through the fusion of data from proprioceptive force feedback and external vision-based sensors as seen in [9].

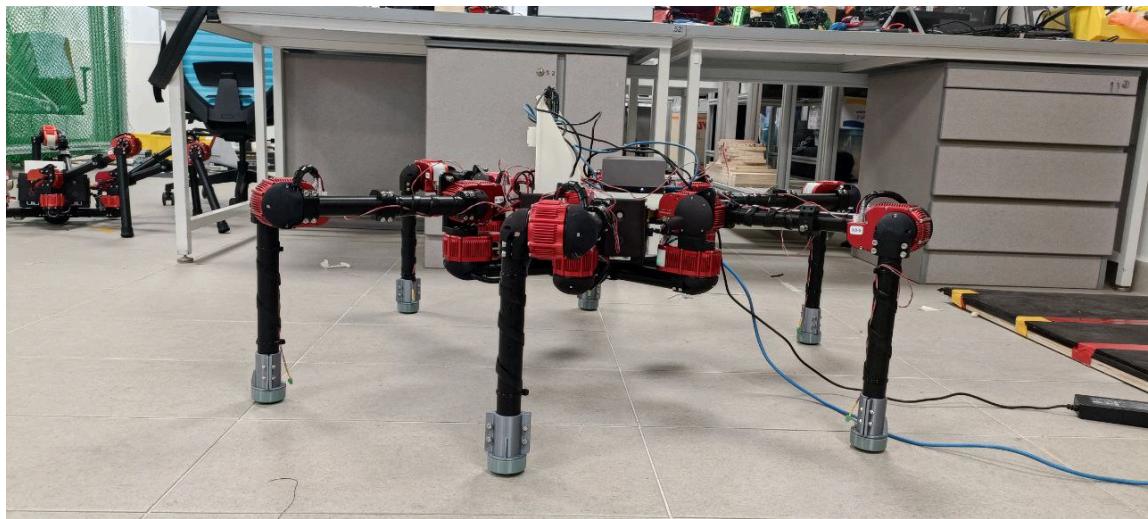


Figure 15: The binary contact sensors attached to the robot end effectors

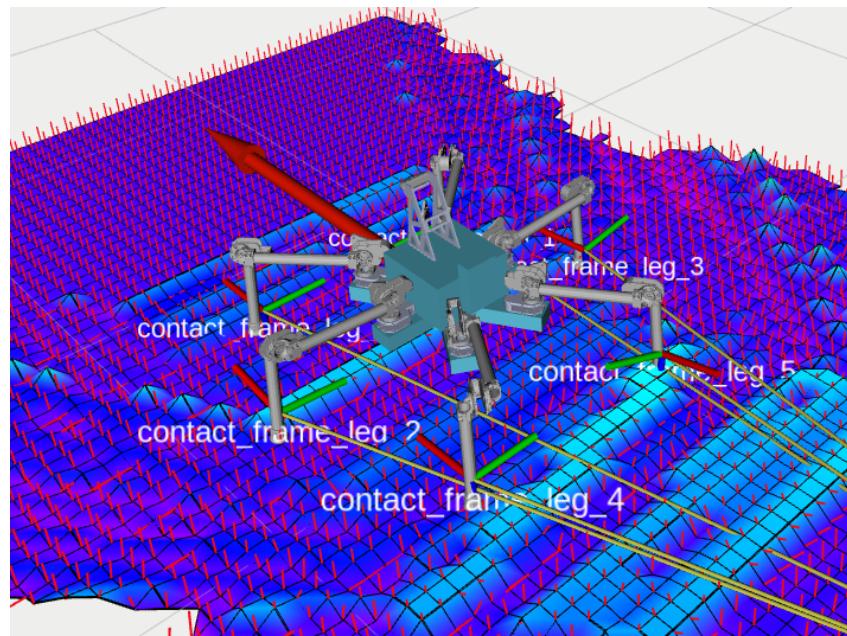


Figure 16: The red arrows represent the surface normal at the vertex of each elevation map cell and are used to build the contact frame for each leg end effector.

5.3.11 Check, Transition, and Action

The following table specifies the Check, Transitions, and Actions for each leg in the EFSM. The transitions are activated when the Checks evaluate to True, and the Transition Actions refer to additional actions besides state transition.

Table 4: Checks, Transitions, and Action for EFSM States

State	Check	Next State	State Action	Transition Action
LND	If the sum of the elements of the leg indices of the opposite tripod group in binary array B equals to 0	WAS	Record the following data: 1) Leg end effector positions in the map frame, ${}^m z_i$ 2) Normal contact GRF in the base frame, ${}^B F_{z,i}$ 3) Timestamp of each sample, t_i 4) Current state of the leg, e_i 5) Sequence of footfalls made by the end effector, S_i 6) Footfall location of leg end effector in the sequence S_i	-
WAS	$sum(B) == 6$	TKO	1) Record the following data: 2) Leg end effector positions in the map frame, ${}^m z_i$ 3) Normal contact GRF in the base frame, ${}^B F_{z,i}$ 4) Timestamp of each sample, t_i 5) Current state of the leg, e_i 6) Sequence of footfalls made by the end effector, S_i 7) Footfall location of leg end effector in the sequence S_i	Send out recorded data.
TKO	$B[i] == 0$	SWG	-	-
SWG	$B[i] == 1$	LND	-	$S_i \leftarrow S_{i+1} + 1$ ${}^M R_{S_i} \leftarrow {}^M R_{S_{i+1}}$

For the transition from SWG to LND, the end effector must be contacting the terrain. For the transition from LND to WAS, it is checked if the legs in the complementary tripod are all in SWG. This confirms that the 3 legs in the current tripod group are in the WAS phase. For the transition from WAS to TKO, it is checked if all the 6 legs are contacting the terrain and for the transition from TKO to SWG, it is checked if the leg end effector has left the ground. During the LND and WAS phase, the recorded quantities are stored for all the legs and are then logged out to the local system when the leg end effector exits the WAS state, i.e., the transition from WAS to TKO is triggered. This is because we want to analyse and extract the GRFs from the point of contact when first transitioning to LND to the maximum GRF in WAS and use the logged-out z-displacement values to the Stiffness Extractor to estimate the terrain stiffness.

In Figure 17, it can be observed that the EFSM is successful in correctly estimating the states and triggering the state transitions.

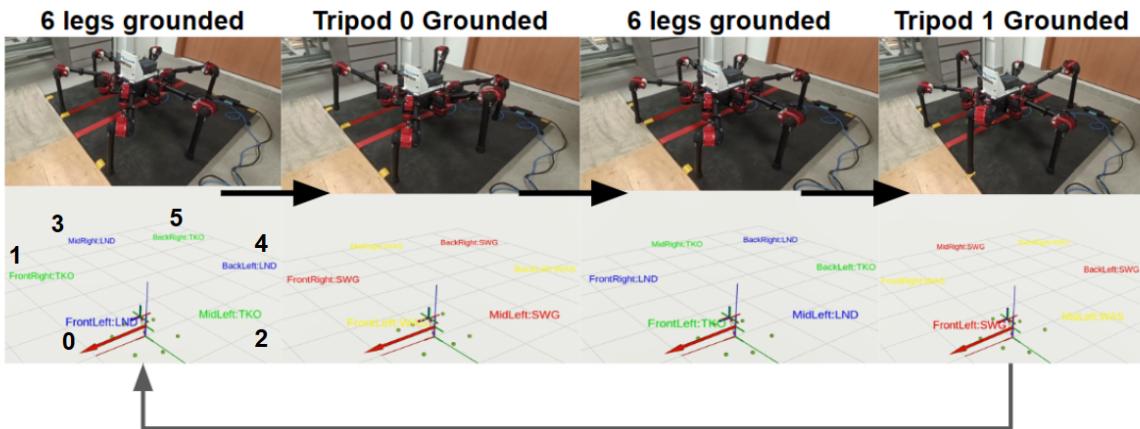


Figure 17: Successful State Transitions in a Gait Cycle

5.4 Stiffness Extractor

The role of the stiffness extractor is to use the GRF and displacement data obtained from the EFSM to estimate the stiffness at the point of contact between the end effector and the terrain. It achieves this by linearly regressing the GRF against the displacement, where the gradient K of the regressed line represents the estimated stiffness. This derivation is in line with modelling the terrain as a linear spring system, as discussed in Section 5.3.6. The extracted stiffnesses are then normalised to a scale between 0 and 1, where 0 is the reference stiffness for floor, an infinitely stiff surface, and 1 is the reference stiffness for a mattress, a highly deformable surface. Thus, the stiffness extractor outputs a stiffness score that is between 0 and 1 to quantify the terrain stiffness at a foothold location. The stiffness scores that are obtained are with respect to each leg, since the GRFs encountered by each leg end effector varies due to the difference in weight that is supported by each leg of the robot. The stiffness scores also do not necessarily reflect the actual stiffness of the terrain but is rather the perceived stiffness of the terrain. To this end, we designed an experiment setup to test the following hypothesis:

Hypothesis 3: *The contact dynamics of the interaction between the end effector and non-rigid surfaces can be reasonably modelled as a linear spring system.*

5.4.1 Stiffness Extraction Experiment Setup

To build the stiffness scale, it was necessary to estimate the stiffnesses of the reference terrains – the floor and the mattresses. To this end, the robot was made to walk on hard floor and a foam mattress separately where the EFSM was used to extract the GRF and displacement data for all 6 legs during terrain locomotion.

The end effector heights are converted to displacement relative to the footfall location height to reflect the extension ${}^M\Delta z_i$ into the terrain as follows:

$${}^M\Delta z_i = {}^M R_{z,S_i} - {}^M Z_i \quad (8)$$

Before extracting the GRF and displacement data to perform regression and extract the stiffness result, there is a need to perform a series of checks on all the collected footfall data sequences to reject inaccurate data that could reduce the accuracy of the computed stiffness scores.

5.4.2 Stiffness Extraction Acceptance Checks

Check 1: Incomplete Sequence Check

Sequence data that did not contain both LND and WAS phases were rejected as it indicates that the state transition was missed.

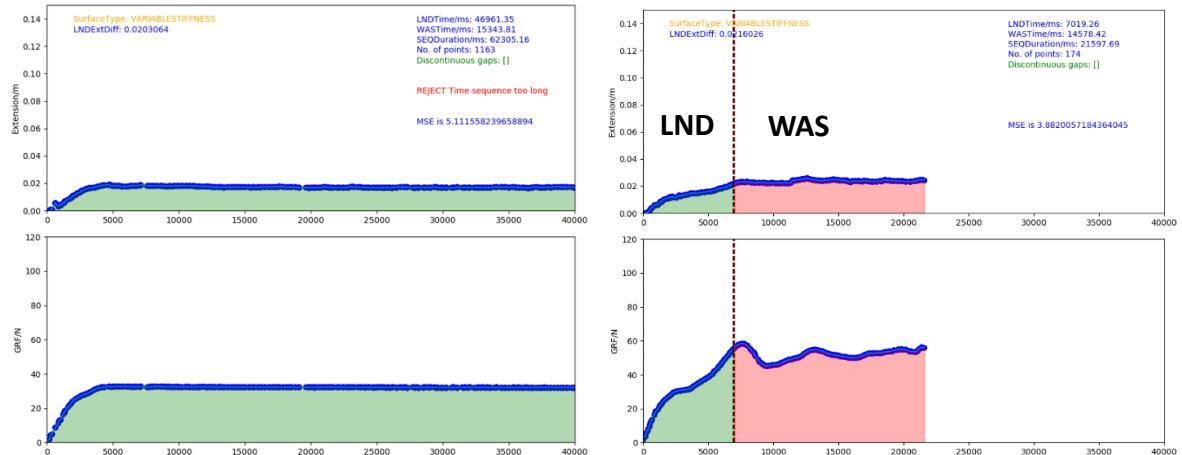


Figure 18: (Left) Incomplete Sequence Check, (Right) Complete Sequence Check

Check 2: Sequence Duration Check

In our experiments, the set period of half a gait cycle is 1.5 seconds. Sequences with excessively long durations indicate that it took far longer than expected to complete the state transition. Based on our observations during the experiments, this could be attributed to the motion of the leg being stopped during the gait. Since such overly long sequences are not representative of the transient contact dynamics, including such data in the stiffness calculations could reduce the accuracy of the computed stiffness scores. Hence, we reject such sequences of data.

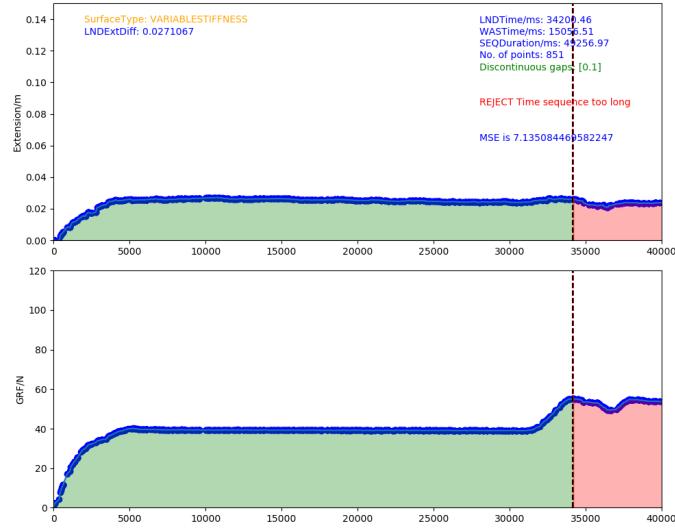


Figure 19: Sequences with excessive duration

Check 3: Rigid Surface Check

When the end effector contacts the floor (and other equally rigid surfaces), there is zero to minimal displacement experienced by the terrain. Compared to walking on mattresses, the change in GRFs are also more sudden and less continuous when walking on floor. Sudden changes in GRF coupled with minimal displacement result in large and spiking stiffness gradients. Therefore, rigid surfaces can be identified by observing the extremely minimal change in displacement as marked by the purple rectangle in Figure 20. Without having to calculate the stiffness gradients, we can determine if the contacted surface is rigid by checking if the extension (change in displacement) throughout the sequence is less than a pre-defined threshold.

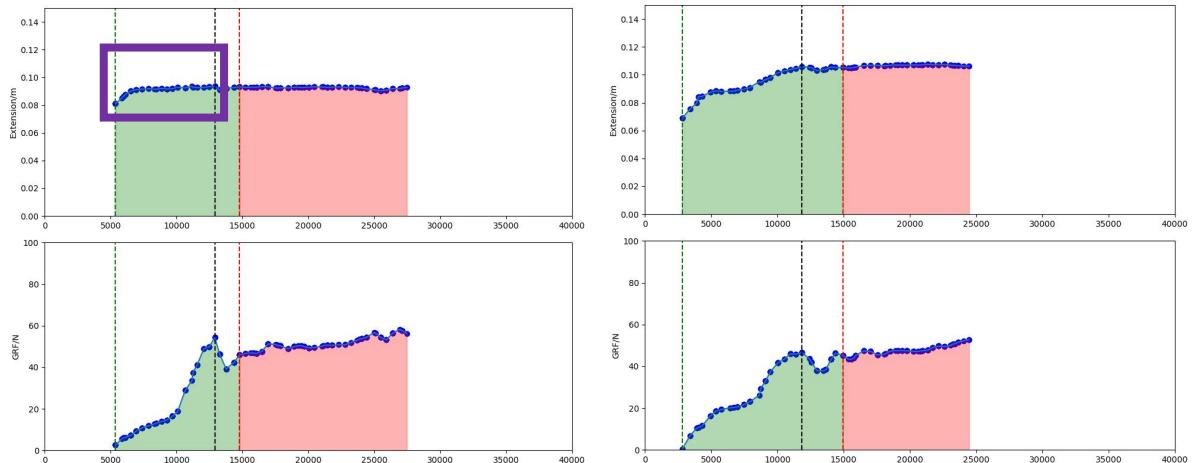


Figure 20: (Left) Displacement on floor, (Right) Displacement on foam

Check 4: Discontinuity Check

Sequences with discontinuous GRF and displacement do not provide accurate regression results as there are lesser points in the input data. Therefore, data with discontinuities as seen in Figure 21 were discounted from the stiffness gradient calculations. To minimise such discontinuous data, faster sampling rates were employed and discontinuity checks were introduced.

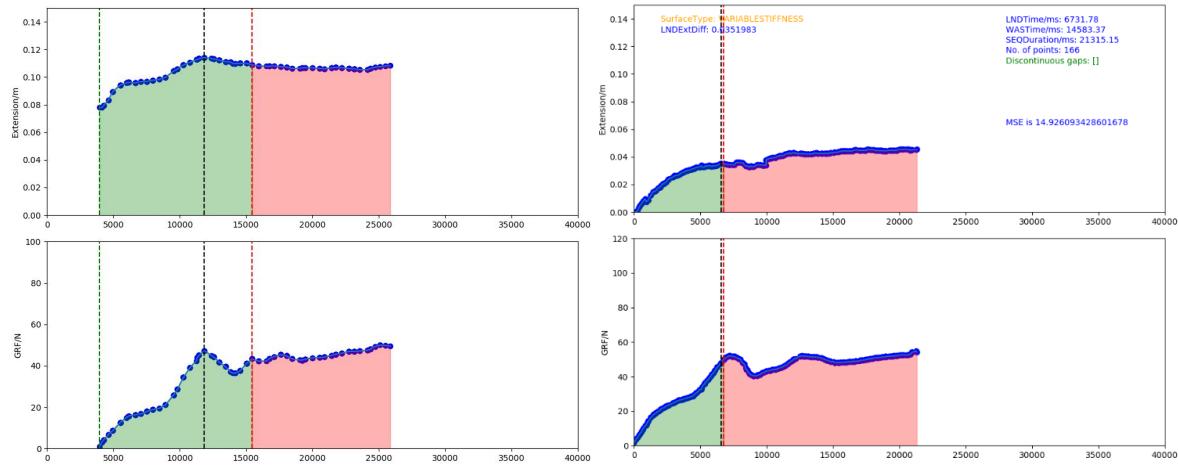


Figure 21: (Left) Discontinuous data, (Right) Smooth data

5.4.3 Regression for Stiffness Gradient Calculation

Once the aforementioned checks have been passed, the GRF and displacement values can be extracted from the LND phase of the data sequence as shown in Figure 22 to compute the estimated terrain stiffness. Linear regression was then performed on the data where a line was fit on the profile of GRF against displacement. The third plot shows the results of the regression, where the stiffness gradient of the foam terrain was estimated to be approximately 1833 N/m.

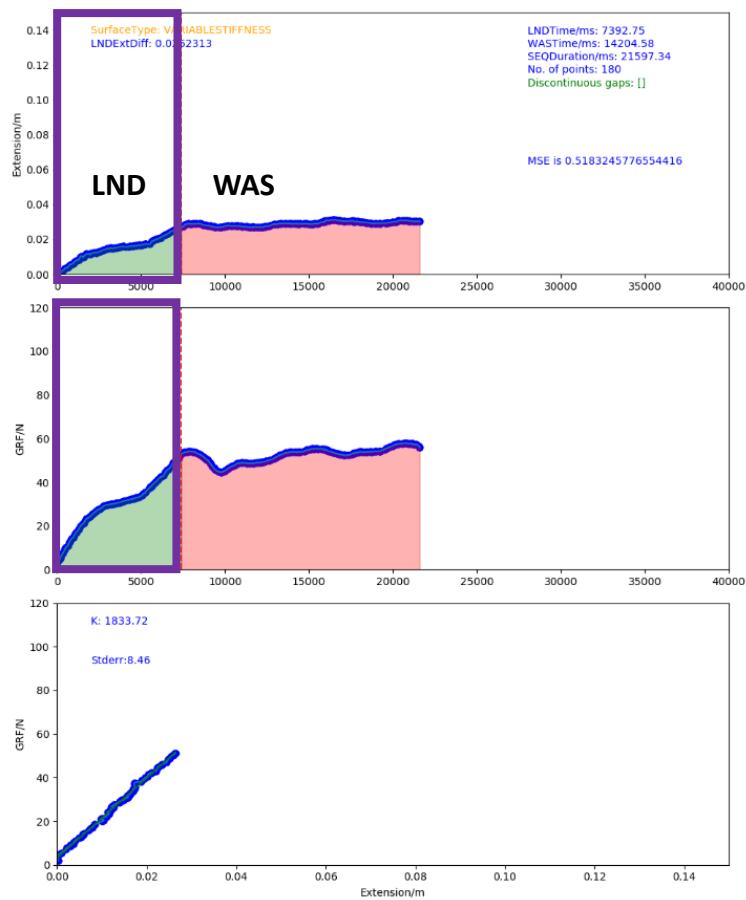


Figure 22: Extracting GRF and Displacement for Stiffness Regression

Despite the implemented checks to filter out inaccurate data, there could still be poorly regressed gradients. Hence, to assess the quality of the estimated gradient, the following 2 metrics were used:

1. **Residual Standard Error (RSE):** Statistical measure that quantifies the typical or average difference between observed data points and the values predicted by a statistical model, often in the context of regression analysis. It represents the unexplained variability or error in the model and is a measure of how well the model fits the data, with smaller RSE values indicating a better fit. In this work, we reject the estimated gradient if the RSE is greater than the experimentally determined threshold of $T_{RSE} = 35$.
2. **Mean Square Error (MSE):** Statistical measure used to measure the average squared difference between the values predicted by a model and the actual observed values in a dataset. It is commonly used to assess the quality of a predictive model, with lower MSE values indicating a better fit between the model and the data. In this work, we reject the estimated gradient if the MSE is greater than the experimentally determined threshold of $T_{MSE} = 85$.

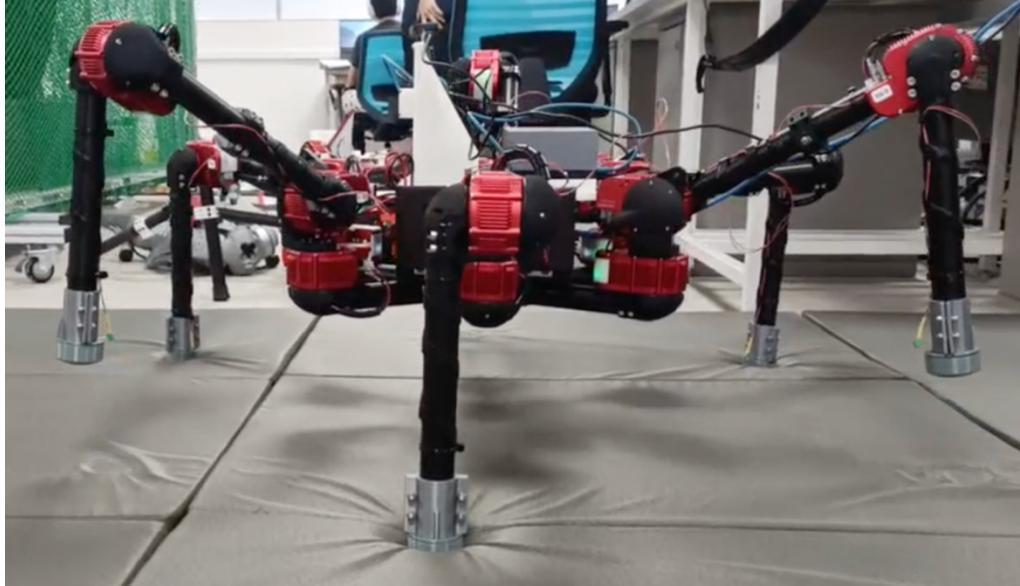


Figure 23: Robot walking on the Raholt mattress

5.4.4 Stiffness Scale Construction

The Raholt mattress as seen in Figure 23 is the reference material for the lowest stiffness in the stiffness scale. To determine the scale, the robot was made to walk for 225 sequences (or footfalls) over the Raholt mattress, of which some of them were filtered out by the pre-regression and post-regression checks. The statistics for the accepted measurements for each leg was computed and is reflected in Table 5.

Table 5: Stiffness Scale Statistics

Stiffness Scale Statistics					
Leg 0 (FrontRight)	Leg 1 (FrontLeft)	Leg 2 (MidLeft)	Leg 3 (MidRight)	Leg 4 (BackLeft)	Leg 5 (BackRight)
Mean	Mean	Mean	Mean	Mean	Mean
2075.27372	2322.301458	2457.126667	2646.994533	2272.829706	2303.885037
SD	SD	SD	SD	SD	SD
425.9700662	457.1075046	252.4638799	299.7308318	471.1848253	395.6137992
Min	Min	Min	Min	Min	Min
116	390.5247523	2008	2061	423.8542215	1454
Max	Max	Max	Max	Max	Max
2705	2839	2997.73	3377.53	3332	3209
Range	Range	Range	Range	Range	Range
2589	2448.475248	989.73	1316.53	2908.145778	1755
Sample Count	Sample Count	Sample Count	Sample Count	Sample Count	Sample Count
180	178	170	169	175	173

It can be observed that although inter-leg comparisons are not accurate, the mean values of all the 6 legs are similar to each other. Moreover, the standard deviation of the estimated stiffnesses for each leg was reasonably low, allowing for us to accurately estimate the stiffness of the contacted terrain. Finally, 77% of the 225 sequences passed both the pre-regression and post-regression steps, indicating that the measurements obtained are mostly acceptable. The stiffness thresholds for the mattress are computed using the mean values from Table 5 such that the stiffness thresholds $t_k = [2075, 2322, 2457, 2646, 2272, 2303]$, where each index corresponds to the leg number of the robot.

Given that K_i is the stiffness estimation at the footfall location of leg i , the continuous stiffness scale for each leg i is defined as follows:

$$\rho_i = \begin{cases} \min\left(\frac{t_k[i]}{K_i}, 1\right) & \text{if deformable terrain} \\ 0 & \text{if rigid terrain} \end{cases} \quad (9)$$

If the rigid ground check that compares displacement passes, then the contact terrain is classified as strictly rigid and is assigned a stiffness score $\rho_i = 0$. Footfalls that are stiffer than the mattress but less stiff than the floor will have $\rho_i \in (0, 1)$. Footfalls that are as deformable (or more) as the mattress will have $\rho_i = 1$.

5.4.5 Stiffness Visualisation Experiment Setup

To test the accuracy of the stiffness estimation, we designed an experiment setup of a 4.6 m x 4.6 m environment with the grey deformable mattresses, and wooden planks with high stiffnesses that have randomised locations throughout the environment as seen in Figure 24. The robot was made to follow this predetermined set path in an alternating tripod gait.



Figure 24: (Left) Experiment setup of the terrain environment, (Right) Predetermined path traversed by the robot in the environment.

The results of the footfall stiffness estimation visualisation are shown in Figure 25. Footfalls were visualised as red and blue markers. Light red or pink markers indicate a higher footfall stiffness with $\rho_i \in (0, 1)$ closer to 0. Darker red cubes indicate $\rho_i \in (0, 1]$ closer to 1 inclusive.

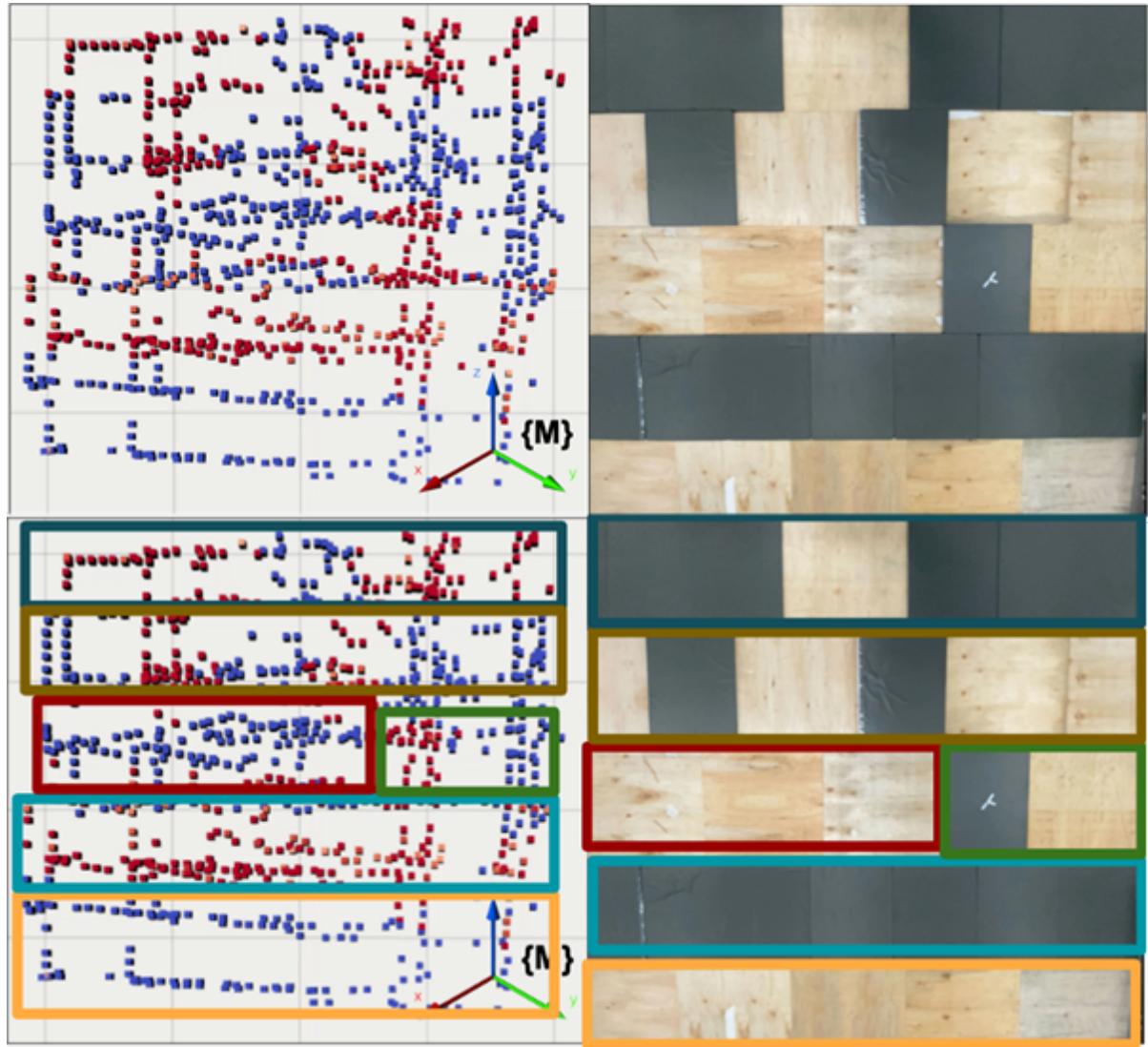


Figure 25: Randomised Terrain Prediction Outcomes

The planks in the regions marked by the orange and red locations have footfall predictions represented by blue cubes which have stiffness score $\rho = 0$, which satisfies our expectation. The mattresses in the light blue region also have correct footfall predictions as shown by the red cubes indicating stiffness score $\rho = 1$. In the remaining rectangular regions with mixed terrains, we can still see that the predicted footfall stiffnesses correspond correctly to the actual stiffness of the terrain. There are

also predictions marked by pink footfalls, which tend to occur around the boundaries between the plank and mattress terrains. This could be because these border regions may return stiffness scores that are a combination of 2 or more terrain types. Hence, we can conclude that it is indeed reasonable to use proprioceptive feedback with the appropriate techniques for terrain stiffness estimation.

6 Mapping Stack

6.1 Foothold Measurements Representation

The terrain stiffness measurements estimated by the robot feet at discrete sensed locations can be used to construct a global terrain stiffness map of the environment. The purpose of this spatial representation is to provide robots with valuable terrain information of its environment, informing the agent of how the stiffnesses change with location. This allows for the adjustment of its foot placement for path planning to explore areas of interest with different stiffness from that of the majority terrain. In Figure 26, the estimated stiffness footholds by the robot in its set environment can be represented with their global 2D Cartesian (x, y) coordinates and their associated stiffness values shown as either red if lower stiffnesses or blue if higher stiffnesses.

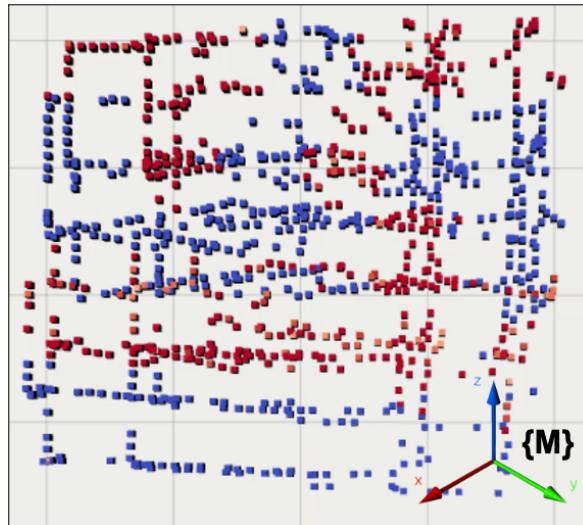


Figure 26: The estimated stiffness footholds of the terrain environment in the global frame

To generate a terrain stiffness map that enables us to predict the location of hidden objects of interest, a mapping from foothold global Cartesian positions to stiffnesses is essential. In unstructured environments such as underwater sandy surfaces with buried metal objects of interest, the spatial data reflecting the differing stiffnesses is expected to be nonlinear since the objects could not only be irregular but could also be buried in multiple varying locations in a manner like Figure. Hence, nonlinear

regression methods would be appropriate to model the spatial relationship between foothold positions and their associated stiffnesses. There is also a need to interpolate between the spatial measurements as it would be intractable for the robot to obtain foothold measurements at every point of its mapping and planning area. However, such interpolation will introduce uncertainty into the model and thus it is important to model the confidence of such uncertainties.

Gaussian Processes are suitable statistical models for the probabilistic modelling of such data with continuous distributions. They provide a flexible framework for modelling complex, non-linear relationships in data and provide uncertainty estimates for their predictions. These predictions are also extended to regions in the mapping space that do not have foothold measurements due to interpolation and incorporate the associated uncertainties into these predictions. Gaussian Process is an especially appropriate tool for our work as they are effective in modelling spatial and temporal data, allowing for the learning of an accurate mapping from foothold positions to stiffnesses. The temporal dependencies also allow for the update of the model's predictions of the environment over time with increased foothold measurements. The regular update of the robot's belief of its terrain stiffness environment is useful in building a terrain stiffness map for anomaly detection, allowing us to uncover such buried objects of interest.

6.2 Gaussian Process Preliminaries

Gaussian Processes can be represented as $\zeta \sim GP(\mu, P)$, where it provides a natural interpolation of the mean and variance between given discrete measurements. In this work, terrain stiffness maps contain terrain-specific information at discrete sensed locations over a 2D domain $\epsilon \subset \mathbb{R}^2$ as a continuous function $\zeta : \epsilon \rightarrow \mathbb{R}$. Given a set of observed locations $X \subset \epsilon$, and its corresponding set of measurements Y , used for regression to infer the set of new observations $X^* \subset \epsilon$, the mean and covariance of the GP are regressed as:

$$\mu = \mu(X^*) + K(X^*, X)[K(X, X) + \sigma_n^2 I]^{-1}(Y - \mu(X)) \quad (10)$$

$$P = K(X^*, X^*) - K(X^*, X)[K(X, X) + \sigma_n^2 I]^{-1} \times K(X^*, X)^T \quad (11)$$

where $K(\cdot)$ denotes a predetermined or chosen kernel function that is used to interpolate between observations, σ_n^2 is a hyperparameter that characterises the measurement noise, and I is an identity matrix of dimensions $n \times n$. The confidence interval of predictions is given by the square root of the diagonal elements of the covariance matrix. Therefore, by using a Gaussian Process, we can create a probabilistic model of the terrain stiffness map, accounting for uncertainties in measurements and providing the flexibility to adjust the model online as new measurements are obtained. In this work, we used the Matern kernel function as it is flexible and can capture a wide range of smoothness levels in the data to model functions with varying degrees of smoothness. It is also robust to outliers and noise in the data, providing stable and reliable predictions even in the presence of noisy observations. Moreover, an independent and identically distributed random Gaussian noise kernel was added to the Matern kernel to account for sensor noise. The length scales of the Matern and noise kernel, which represents how much the function represented by the process changes with respect to changes in the input observations, were set to 0.16 and 0.18 respectively.

To generate a terrain stiffness map that predicts the locations of buried objects in sand, we use the obtained footfall locations $X = (x, y)$ and the filtered normal ground reaction force measurements of each leg as separate inputs for the Gaussian Process. The predicted mean values from the Gaussian Process for each leg's data are normalised to a range between (0, 1). These normalised means are then combined using equal weights to create the final stiffness terrain map. To this end, we devised an experiment setup to build the terrain stiffness maps and test the following hypothesis:

Hypothesis 4: *Nonlinear regression with uncertainty-incorporated predictions can be reasonably used to build a terrain stiffness map based on the provided foothold stiffness measurements.*

6.3 GP Terrain Stiffness Map Experiment Setup

Experiment 1: Mixed Terrain

As seen in Figure, we designed an experiment setup of a 4.6 m x 4.6 m environment with the grey deformable mattresses, and wooden planks with high stiffnesses that have randomised locations throughout the environment. The resolution of the map was set to 0.05 m, which represents the diameter of the robot's foot. Hence, our Gaussian Process map space can be visualised as a 96 x 96 grid matrix, where each cell in the grid represents a foothold position and its value represents the predicted stiffness mean. We used a joystick controller that is programmed to process and convert joystick movements into inverse kinematics commands for robot's legs to follow a predetermined path as outlined in Figure 27.



Figure 27: (Left) Experiment setup of the terrain environment,
(Right) Predetermined path traversed by the robot in the environment.

While the robot was executing its predetermined path, the Gaussian Process model of the environment was updated and predicted online at a frequency of 250 Hz using the real-time foothold measurements of the robot. Each time the robot executes a half-cycle of its tripod gait, there are up to three foothold measurements that are obtained. These measurements are passed into time-ordered observation heaps that preserve

the temporal order of the incoming measurements and process them in that order. The foothold measurements are converted from the world frame to the stiffness map frame and are then fit into the Gaussian Process model and the predictions are made over the entire map space.

The Gaussian Process model that was obtained at the end of the path traversal and mapping is shown in the following Figure 28. The yellow regions represent areas with lower stiffness values $0.0 < k \leq 1.0$ whereas the blue regions represent areas with higher stiffness values $k = 0.0$. It can be observed that the spatial map on the right closely represents the ground truth of the environment.

Moreover, it can also be observed that there are footholds in the map that are pink in colour. These footholds represent stiffness values that are low but not lower than the mattresses. The location of these footholds evidently appears near the boundaries between the planks and the mattresses, suggesting that the end effector could be contacting both surfaces simultaneously. The faded colours of the region surrounding the pink footholds also highlight that the Gaussian Process is accounting for these uncertainties in its predictions, that is, it can only predict about the stiffness profile of that region with reduced certainty. The modelling of this uncertainty is also reflected outside the space where the measurements were taken, as there are no observations for interpolation.

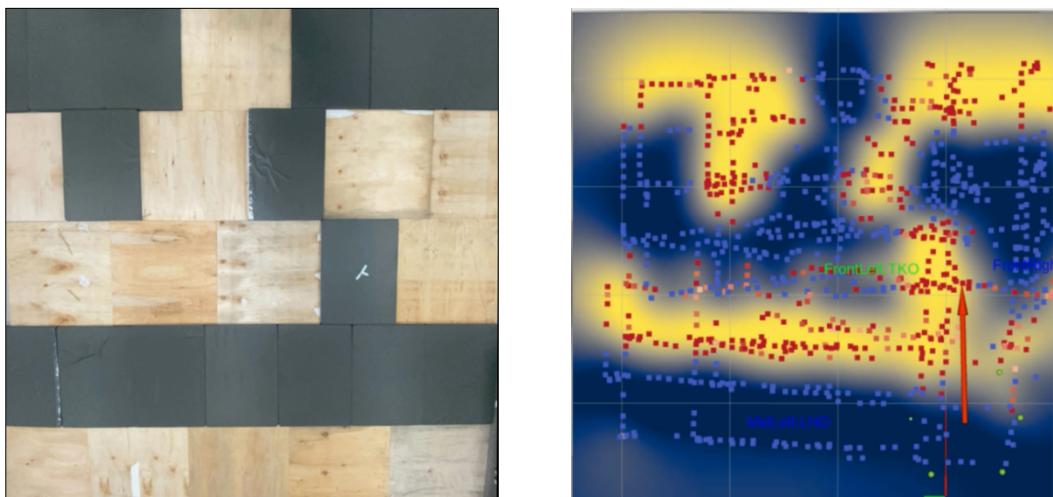


Figure 28: (Left) Ground truth of the terrain environment,
(Right) Gaussian Process stiffness prediction of the terrain environment

Experiment 2: Majority Mattress

As seen in Figure, we designed a similar experiment setup of an environment with the grey deformable mattresses, and wooden planks with high stiffnesses. However, in this setup, the ratio between the mattresses and wooden planks is increased as we mimic the majority sand environment with lower stiffness that has stiffer objects inside it. Figure 29 shows the experiment setup of the environment and the predetermined path taken by the robot during the experiment to collect foothold measurements. To mimic buried objects under the sand, we also hid metal plates under one of the mattresses, located by the green circle in Figure.



Figure 29: (Left) Experiment setup of the terrain environment,
(Right) Predetermined path traversed by the robot in the environment.

The Gaussian Process model that was obtained at the end of the path traversal and mapping is shown in Figure 30. It can be observed that the planks and the hidden plates were successfully located by the Gaussian Process as reflected by the blue higher stiffness regions in the terrain stiffness map.

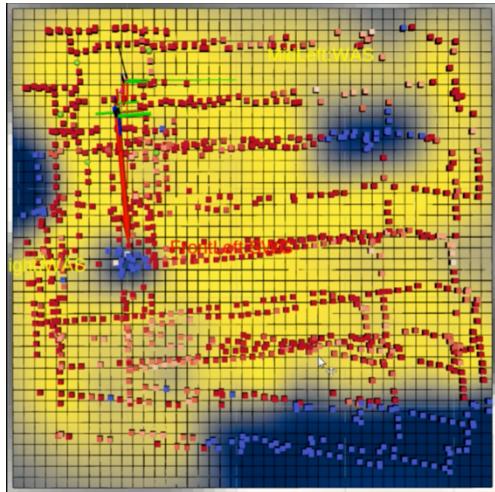


Figure 30: (Left) Ground truth of the terrain environment,
(Right) Gaussian Process stiffness prediction of the terrain environment

7 Planning Stack

7.1 Informative Path Planning

In Section 6.3, the robot followed a predetermined path that allowed it to cover the map space extensively. This enabled the robot to obtain sufficient foothold measurements to build a near-complete terrain stiffness map that located the anomalies successfully. However, the problem with extensive locomotion over the entire space is that it comes at a cost of energy expenditure, rendering such search operations infeasible in large-scale environments. Therefore, there is a need for the robot to achieve a balance between energy expenditure and maximising information gain while also exploring regions with greater uncertainty to uncover potential anomalies. This is formally known as the Informative Path Planning (IPP) problem, which aims at maximising the knowledge within a certain environment from collected measurements. This is achieved by planning an optimal local path for the robot, where the path chosen has the greatest informativeness in the space of all possible paths within the environment while also enforcing a constraint that the cost of path should not exceed its predefined budget.

7.2 Adaptive CMA-ES Planner Preliminaries

Adaptive Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is an optimisation algorithm that is appropriate for solving such IPP problems. It employs a covariance matrix to model the multivariate distribution of candidate solutions in the search space, which is adapted over time to control the shape and orientation of the search distribution. The evolutionary strategy of CMA-ES evolves the population over iterations where it uses the current distribution and covariance matrix to generate new candidate solutions, of which the most promising individuals from each generation are selected to form the basis of the next generation. A population of these candidate solutions are maintained, which are evaluated using the objective function to determine their fitness values to search for the optimal solution.

7.3 Encourage Exploration with Enforced Budget

In this work, the CMA-ES planner was used to plan local paths for the robot to explore regions with greater uncertainty, i.e., regions with lesser measurements, while also enforcing a local path budget in terms of distance. Provided with a global base-link trajectory P_g , planning horizon h , local path budget B , and the Gaussian Process prediction space, the CMA-ES planner will select and execute an optimal local path from its candidate solutions for every n iterations with a population size p until the budget B is exhausted. The generation of the candidate solutions is achieved through the objective function of the CMA-ES planner. We set $B = 1.5 \text{ m}$, $h = 0.8 \text{ m}$, $n = 100$, $p = 10$.

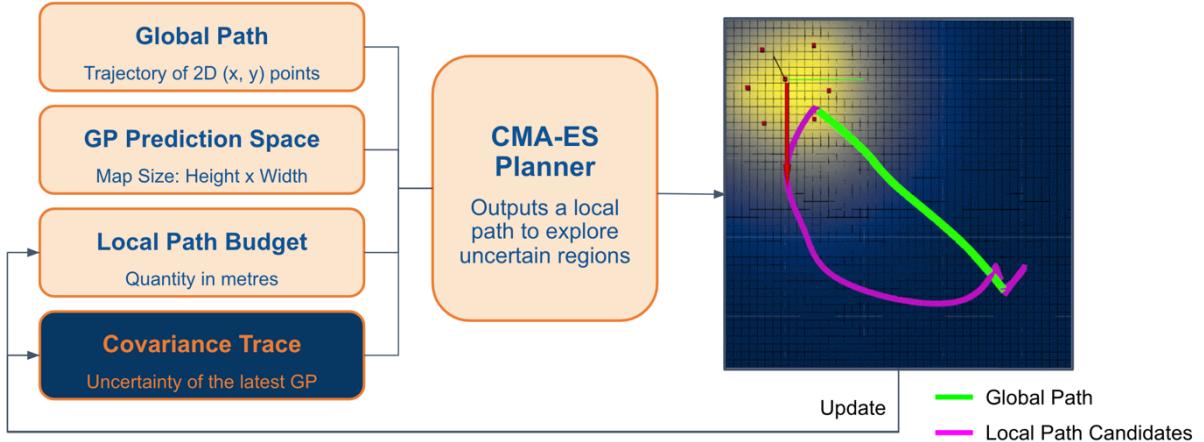


Figure 31: CMA-ES Planner Pipeline for Budget-Enforced Footfall Exploration

We designed the objective function to minimise the product of the covariance trace returned by the sampling function and the budget penalty incurred by travelling the path. Inside the sampling function, footholds are sampled based on successive nodes of an initial unoptimized local path. The position of these footholds and their mock observed values ($k = 1.0$) which represent the majority environment are fit in as observations to the Gaussian Process map space. Upon prediction over the Gaussian Process space, the returned standard deviation matrix is squared and multiplied to calculate the trace of the covariance matrix of the Gaussian process space with grid height h_c and grid width w_c as follows:

$$Tr(Cov) = \sum_{n=0}^{h_c * w_c} \sigma_{sample}^2 \quad (12)$$

The design of such an objective and sampling function that is centred on minimising the covariance trace of the Gaussian process space will encourage the robot to explore uncertain regions in the map. Initially, the chosen local path is likely to be random and dependent on the chosen local start and end points from the provided global path. However, as more foothold measurements are taken by the robot, the robot is likely to plan a path towards regions that have less measurements as they have greater uncertainty. We implemented a ground truth terrain environment in simulation and used this planning strategy for the robot to test the following hypothesis:

Hypothesis 5: *A planning strategy that optimises for the minimising of the environment's covariance trace will encourage the exploration of uncertain regions.*

We can see that the robot did plan towards uncertain regions with less measurements as seen in Figure 32. Hence, with an enforced budget, the robot is expected to explore uncertain regions and potentially uncover objects of interest while simultaneously expending less energy/distance in the process. The mapped outcome of the Adaptive CMA-ES Planner is shown in Figure 33, where all the objects of interest with high stiffnesses were successfully located and mapped.

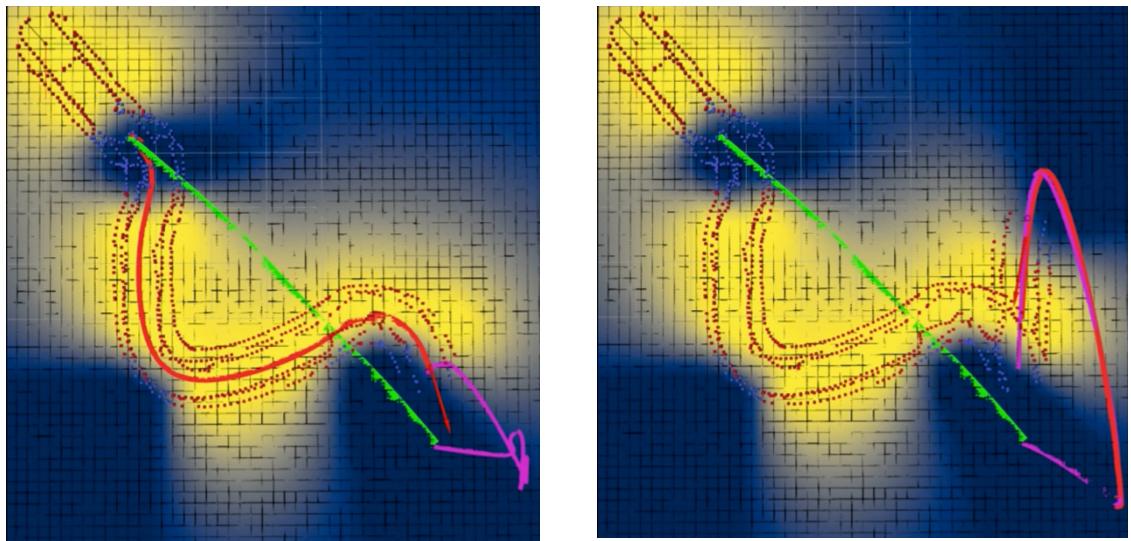


Figure 32: (Left) Sampled candidate local path (in purple) at iteration $n \sim 20$,
 (Right) Selected candidate local path after iteration $n = 100$.

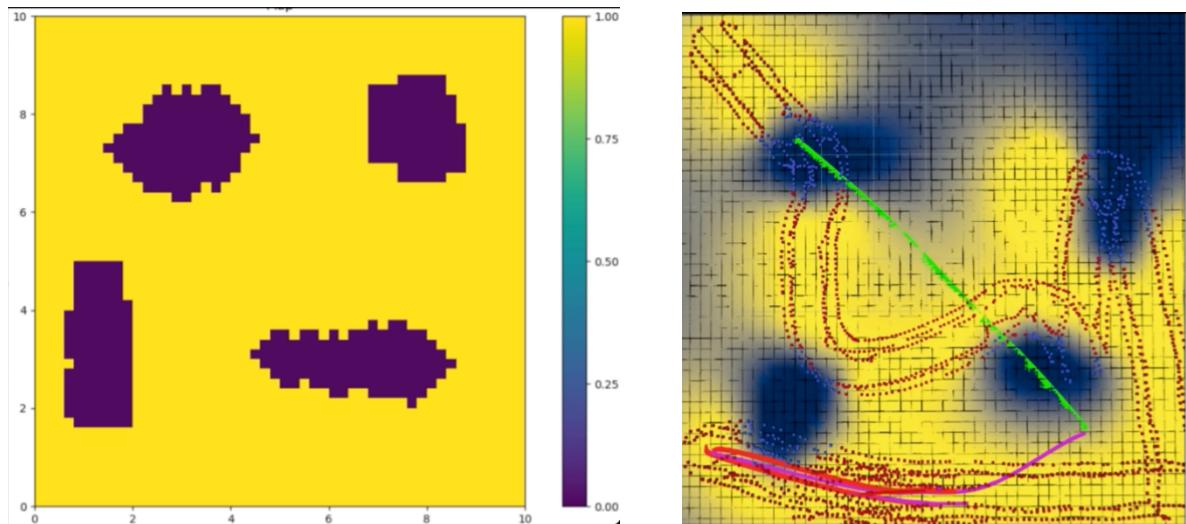


Figure 33: (Left) Ground truth of the simulated environment, (Right) Mapped outcome of the simulated environment upon termination of the CMA-ES planner.

7.4 Shortfalls of the CMA-ES Planner

While the CMA-ES planner is useful in encouraging the robot to explore regions with greater uncertainty, the planning of base-link trajectories results in the foothold measurements being constrained in a fixed configuration with respect to the robot base. This is evident in Figure 32, where the foothold measurements are almost parallel to the local path taken by the base of the robot. This constraint on the orientation of the footholds could be overcome if we planned trajectories over tripods in lieu of bases, which we outline as part of our future work in Section 8.2.

7.5 Tripod Gait Controller

The purpose of the tripod gait controller is to provide control for the robot to move between two successive nodes in a received path using the alternating tripod gait. This is achieved by calculating the translational and angular error between two successive nodes of the received path and scaling them to obtain the step length and rotation respectively. These values will be input to an inverse kinematics controller that will generate cycloid foothold trajectories for the robot end effectors based on an alternating tripod gait to help the robot reach the target node from the current node.

The step length L to traverse between two successive nodes in a path is defined as the Euclidean distance norm between the target position p_{tgt} and the current position p_{curr} that is scaled by the proportional gain k_{px} :

$$L = k_{px} |p_{tgt} - p_{curr}| \quad (13)$$

The rotation θ to turn between two successive nodes in a path is defined as the angular error between the target angular position θ_{tgt} and the current angular position θ_{curr} that is normalised within the $(0, 2\pi)$ range in the *limit_angle* function, where the output is then scaled by the proportional gain $k_{p\theta}$:

$$\theta = k_{p\theta} \cdot \text{limit_angle}(\theta_{tgt} - \theta_{curr}) \quad (14)$$

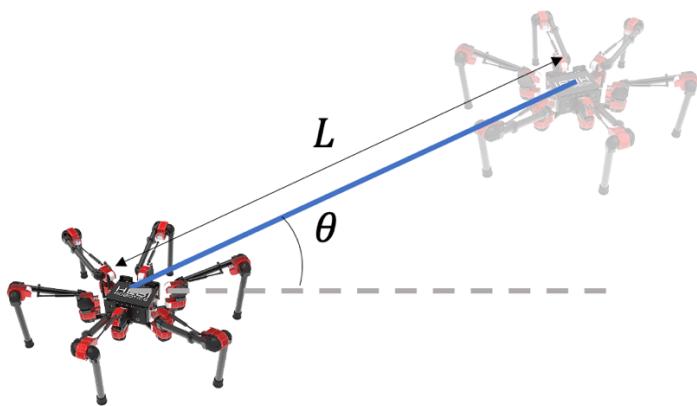


Figure 34: Step Length L and Rotation θ

The step length and rotation values are then passed into the tripod gait inverse kinematics controller, where leg trajectories for all six legs of the hexapod are planned within a stride. For the swing phase, there are n points that are generated over each leg cycloid trajectory of period T with time interval dt between each successive point for continuous control, where n is chosen as follows:

$$n = \frac{T}{2 * dt} \quad (15)$$

In this work, we set $T = 1.5, dt = \frac{1}{240}, n = 120$. For point $k \in (0, n)$ on the complete cycloid trajectory, a sub-trajectory is generated from each point k to $k + 1$ using the cycloid parametric equations. The trajectory of the point can be represented using the following set of cycloid parametric equations and also seen in Figure 35:

$$x = r * (t - \sin(t)) \quad (16)$$

$$y = r * (t - \cos(t)) \quad (17)$$

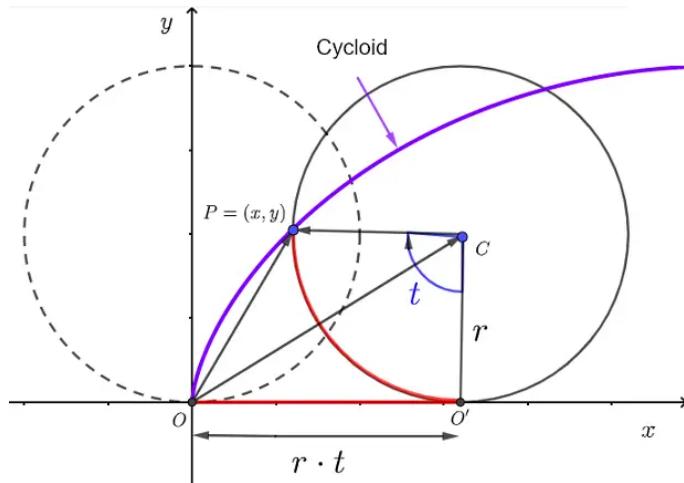


Figure 35: Cycloid Parametric Equations [10]

To use these equations for motion as a foot trajectory, we can resize the equations such that the change in x and y coordinates from point k to $k + 1$ can be expressed as follows:

$$\Delta x_{k \rightarrow k+1} = L \left(\frac{t - 2 \sin(t)}{2\pi} \right) \quad (18)$$

$$\Delta y_{k \rightarrow k+1} = c \left(\frac{1 - 2 \cos(t) + 1}{4} \right) \quad (19)$$

where L stands for step length and c stands for clearance. The sub-trajectory can thus be formed when the changes in these coordinates are applied to the current task-space Cartesian foothold pose to the desired foothold pose. The desired Cartesian pose of this sub-trajectory is then converted to desired joint positions of the legs using inverse kinematics, which are commanded to the joint actuators of the legs to execute the foothold sub-trajectory. When all the sub-trajectories in the subset space of the complete cycloid trajectory have been executed, the leg would have completed a full planned cycloid trajectory. The leg would then transition into the stance phase, allowing the robot to locomote using the alternating tripod gait.

8 Conclusion and Future Work

In this work, we have demonstrated the feasibility for a legged robot system to estimate terrain stiffnesses from the proprioceptive feedback of legged robots, and to use it to build a stiffness belief map and use that map to perform informative path planning to explore the potential areas of interest. In the **research design matrix** in Section 8.1, we outline our objectives, hypotheses, methodologies, and conclusions for this work.

8.1 Research Design Matrix

No.	Objective	Hypothesis	Methodology	Conclusion
1	To determine the plausibility of using end effector GRFs to estimate terrain stiffness.	The sum of the GRFs experienced by all legs that are contacting the ground should equal the total weight of the robot.	Measure and sum the GRFs of the grounded legs when: <u>Case 1</u> All 6 legs are grounded <u>Case 2</u> Only tripod 1 group legs are grounded <u>Case 3</u> Only tripod 0 group legs are grounded.	The sum of the GRFs of the grounded legs differs for the 3 cases, suggesting that inter-leg GRF comparisons for terrain stiffness estimation are inaccurate. However, the experiment uncovered that the use of intra-leg GRF comparisons is reasonable.
2	To determine how force information can be used to determine terrains with different stiffnesses.	There could be differences in the force profiles measured when the robot is locomoting on terrain with different stiffnesses.	Make the robot walk on hard ground and deformable foam separately to measure and analyse the GRF and displacement information.	The contact dynamics between the end effector and the terrain is transient, suggesting that there is a need to decompose the stance phase.
3	To estimate the stiffness of the contacted terrain.	The contact dynamics of the interaction between the end effector and non-rigid surfaces can be reasonably modelled as a linear spring system.	Stiffness extraction tool with rigid surface check and stiffness scale for more deformable surfaces.	It is possible to reasonably estimate the perceived stiffnesses by each leg of the contacted terrain.
4	To build a terrain stiffness map of the environment.	Nonlinear regression with uncertainty-incorporated predictions can be reasonably used to build a terrain stiffness map based on the provided foothold stiffness measurements.	Gaussian processes that accept and interpolate between foothold measurements to provide a predicted mapping from foothold positions to stiffnesses.	A stiffness terrain map built using Gaussian Processes not only provide reasonable stiffness predictions, but also provide us with the uncertainty of such predictions.
5	To encourage the robot to explore the environment and locate the hidden objects of interest.	A planning strategy that optimises for the minimising of the environment's covariance trace will encourage the exploration of uncertain regions.	Adaptive CMA-ES Planner with an optimisation function that minimises the covariance trace of the sampled Gaussian Process.	The robot can explore regions on the map with less measurements to potentially uncover hidden objects of interest.

8.2 Future Work

The Adaptive CMA-ES planner that is used for the robot plans for base-link trajectories, resulting in a constraint on the orientation of the footholds with respect to the map. This does not fully utilise the omnidirectional capability of the 18-DoF hexapod robot, and planning for tripods instead of footfalls can be useful in encouraging the robot to achieve greater variations in footfall configurations. This is currently work in progress, and the potential planned paths in the Gaussian Process space using a tripod-based RRT planner can be seen in Figure 36.

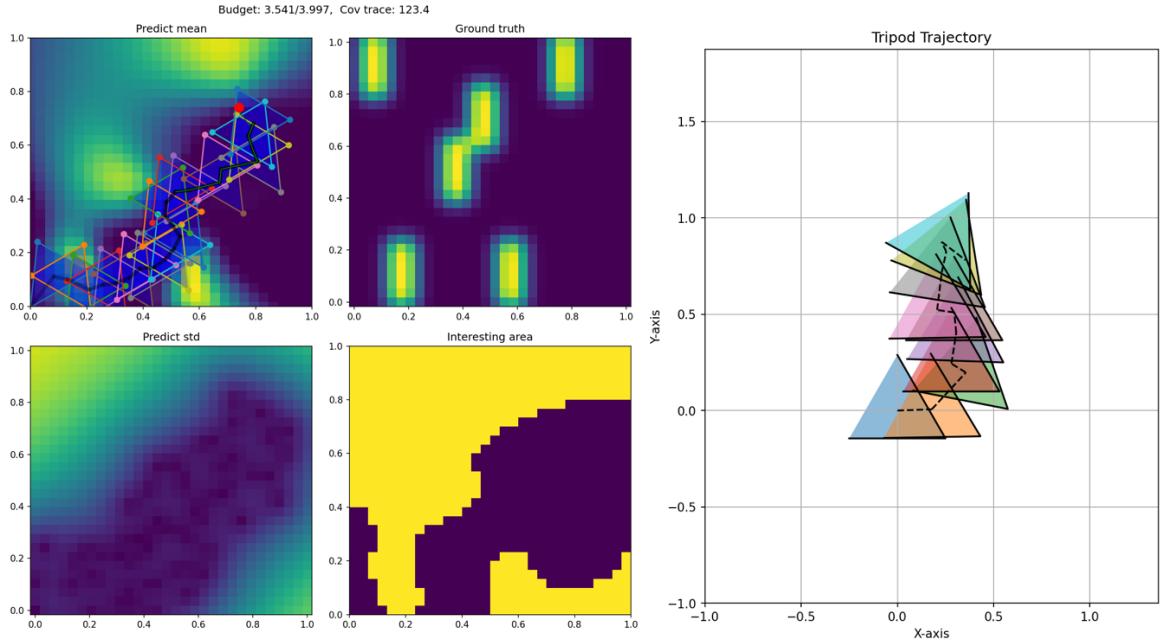


Figure 36: Tripod-based RRT Planning

Future work could also include planning for each individual footfall by treating each leg as an agent potentially through reinforcement learning. This could be an improvement from tripod-based planning and offer further freedom in how the robot can adapt its gait for footfall planning. Finally, we could also incorporate prior informed planning, where information of the object such as its shape and dimensions are already known. This information could be incorporated in the objective function of the planner, and the expected outcome could be like that seen in Figure 36.

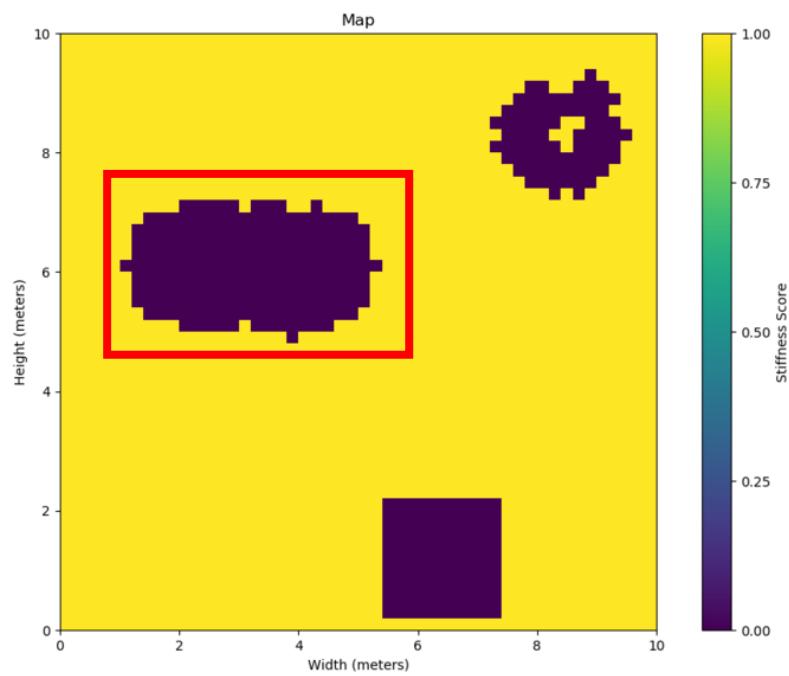


Figure 37: Detecting the known buried object in the environment
as outlined by the red rectangle

Thesis word count: 9879 words

9 References

- [1] J. Liu, M. Tan, and X. Zhao, “Legged robots — an overview,” *Transactions of the Institute of Measurement and Control*, vol. 29, no. 2, pp. 185–202, Jun. 2007, doi: 10.1177/0142331207075610.
- [2] R. Buchanan, J. Bednarek, M. Camurri, M. R. Nowicki, K. Walas, and M. Fallon, “Navigating by touch: Haptic Monte Carlo localization via geometric sensing and terrain classification - Autonomous Robots,” *SpringerLink*, 27-Aug-2021. [Online]. Available: <https://link.springer.com/article/10.1007/s10514-021-10013-w>. [Accessed: 09-Apr-2023].
- [3] E. Krotkov, “Active perception for legged locomotion: every step is an experiment,” in *International Symposium on Intelligent Control*, 1990.
- [4] M. A. Hoepflinger, M. Hutter, C. Gehring, M. Bloesch, and R. Siegwart, “Unsupervised identification and prediction of foothold robustness,” in 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 2013, pp. 3293–3298. doi: 10.1109/ICRA.2013.6631036.
- [5] R. Buchanan, M. Camurri, and M. Fallon, “Haptic Sequential Monte Carlo Localization for Quadrupedal Locomotion in Vision-Denied Scenarios,” in 2020 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 3657–3663. doi: 10.1109/IROS45743.2020.9341128.
- [6] G. Haddeler *et al.*, “Traversability analysis with vision and terrain probing for safe legged robot navigation,” *Front. Robot. AI*, vol. 9, p. 887910, Aug. 2022, doi: 10.3389/frobt.2022.887910.
- [7] Zhao, Yue, et al. ‘Terrain Classification and Adaptive Locomotion for a Hexapod Robot Qingzhui’. *Frontiers of Mechanical Engineering*, vol. 16, no. 2, June 2021, pp. 271–84. DOI.org (Crossref), <https://doi.org/10.1007/s11465-020-0623-1>.
- [8] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization", in *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3019–3026, 2018.
- [9] Wagner, Lucas, et al. *Foot Contact Estimation for Legged Robots in Rough Terrain*. Sept. 2016. *Advances in Cooperative Roototics* <https://doi.org/10.3929/ETHZ-A-010643823>.

Appendix

a. 3D LiDAR for Contact State Estimation

The use of an omnidirectional sensor such as a 3D LiDAR could resolve the fixed FoV problem, but it still results in blind spots around the radius of the hexapod robot as seen in Figure 14. This is due to occlusion arising from the legs of the hexapod that intersect with the point cloud data and appear on the elevation map. The elevation mapping software suite does provide an option to clip elevation data surrounding the camera of a particular pre-defined radius, but this will result in the elevations used for the computation of the contact Boolean C_i to be outdated and thus inaccurate. Hence, 3D vision based methods are inadequate in accurate contact state estimation.

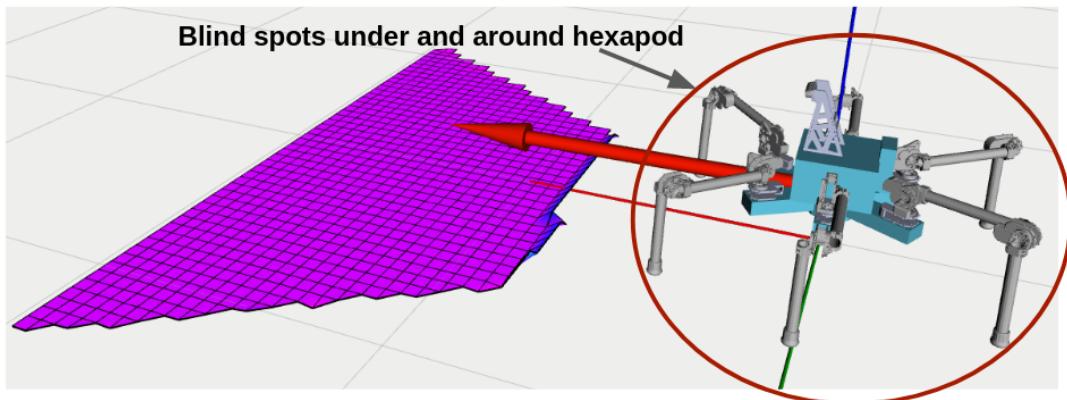


Figure 38: Blind spots arising from occlusion due to hexapod legs