Gait generation for real-time quadruped locomotion

Ernst Jordaan[1], and *Callen* Fisher[2]

[1]Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa, 21063990@sun.ac.za

[2]Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa, cfisher@sun.ac.za

We thank the NRF for funding this research. Grant number 129830.

Abstract. Gait generation for a legged robot in real-time is no simple task. Typically, powerful processors are required to perform real-time trajectory optimisation techniques to generate the trajectories , or the platform requires full state feedback and controllers to track a pre-generated trajectory. Here we present a novel method for generating gaits in real-time by varying three parameters (RideHeight, StepLength, and StepHeight). Various successful walking experiments were performed with the developed quadrupedal platform, demonstrating that this method worked with minimal state feedback (only leg angles) and limited processing power (Teensy 4.0 microcontroller was used). The resulting gait was generated in real-time and maintained stability in open loop. The platform was capable of traversing obstacles (50mm x 50mm) and ascending or descending slopes (15°).

1 Introduction

Current state-of-the-art legged robotic platforms in the literature focus on high-speed locomotion with dynamic manoeuvres [1, 2, 3], exemplified by platforms like Boston Dynamics' Spot and Agility Robotics' Digit. These platforms are capable of running [4], jumping [5], and traversing stairs [6], finding applications in mapping and industrial inspection [7, 8]. However, there are other valuable applications for legged robots that don't require high-speed locomotion. For instance, they can explore rough terrains, carry heavy loads, and operate in challenging areas [9, 10, 11]. This is our focus area.

Cost becomes crucial when developing such a platform, considering the increased risk of damage or loss [12]. Current platforms suited to these environments are expensive due to their emphasis on dynamic manoeuvres. These platforms require complex gait generation methods considering robot dynamics, terrain, and obstacles, resulting in high computation costs and the need for expensive components [8, 13]. Operating such a platform in a risky environment is impractical.

In these environments, where fast locomotion and agility are not a priority, mechanical design complexity and cost can be reduced while improving durability. However, most gait generation techniques in the literature remain complex and require powerful processors [1]. Our focus is on computationally inexpensive gait generation methods adaptable in real-time to the changing terrain. This allows the onboard processors to prioritize other tasks.

The paper is structured as follows: Section 2 describes the developed quadruped and the testing environments. Section 3 details the gait generation method. Section 4 presents and discusses the platform testing and results. The paper concludes in Section 5.

2 Developed Platform and Environment

The gait generation method presented in this paper was developed with two primary objectives: achieving successful quadrupedal locomotion and ensuring computational efficiency in implementation. To accomplish these goals, a quadrupedal platform was specifically designed with low-cost control hardware to accurately evaluate the performance of the gait generation method.

To comprehensively assess the effectiveness of both the robot and the gait generation method, testing was conducted in diverse and complex environments rather than limiting it to isolated scenarios. It was crucial to perform these tests without any external support or human assistance to prioritize the platform's stability, which is a critical requirement in the scenarios our research focuses on. By achieving unassisted locomotion in these challenging environments, the developed platform would validate the success of the proposed gait generation method. The developed platform and testing environments are discussed below.

2.1 Mechanical design and testing environments

A quadrupedal (4-legged) robot with two active joints per leg was developed and can be seen in Figure 1. Each leg consisted of three limb sections: hip, femur, and tibia. Abduction and adduction were not incorporated into the design, thus yielding only 2 pitching degrees of freedom per leg.

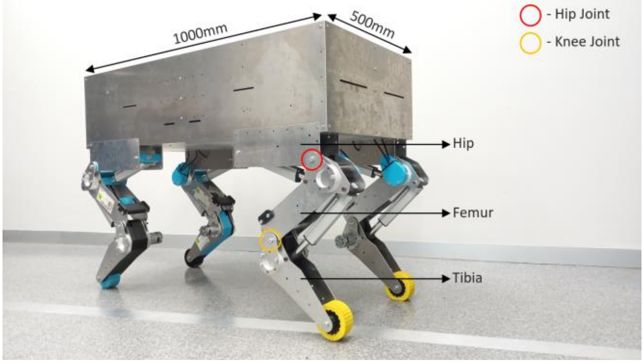

Fig. 1. The developed platform.

The four legs were attached to the body of the robot in the X-configuration (front and rear knees bending inwards) [14]. This configuration was selected through experimental testing as it yielded the best stability by keeping the centre of mass in the middle of the body. With this configuration, open loop stable walking was achieved. The length of the robot was 1 m, the width 0.5 m and the height could be varied between 0.17 m and 0.45 m, with a mass of 25 kg.

Each of the active joints were driven by a 12VDC electrical linear actuator and an optical rotary encoder (1000 PPR) was used to measure each joint angle. The onboard processing was done with four microcontrollers: three Teensy 4.0's and one ESP-32. Two of the Teensy's were used to read the encoder (angle) values and then transmit the readings to the main Teensy which was used for gait generation and controlling the robot. The use of multiple Teensy's were purely due to a single unit not having enough pins for all the required connections. The ESP-32 was used to connect a controller to the robot via Bluetooth. The ESP-32 also connected to the main Teensy to transmit the received user inputs.

Figure 2 shows the environments that the developed platform was set to walk in. Environment 1: A level surface with 50mm x 50mm obstacle placed in the robot's path, followed by a slope with an incline of 15º. Environment 2: A slope with a decline of 15º leading to level ground.

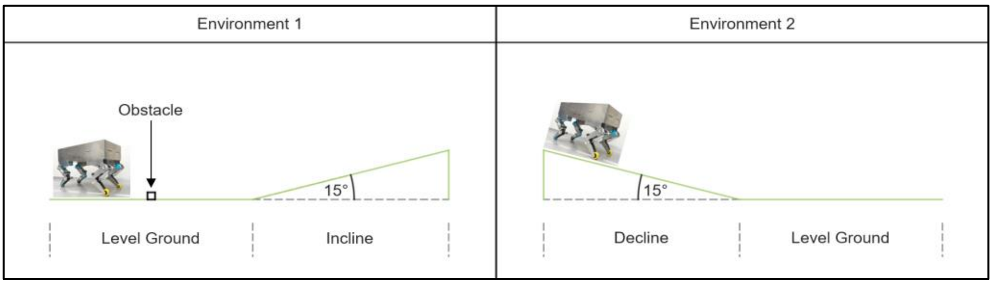

Fig. 2. Testing environments for the developed quadruped.

3 Gait Generation Methodology

A gait is a pattern of foot placements performed in a specific sequence. The trajectory of a foot refers to its path in three-dimensional space during the gait. Quadrupedal robot gaits are divided into static and dynamic categories. Static gaits are suited for slower locomotion tasks that prioritize stability, while dynamic gaits are more suitable for faster locomotion, prioritizing speed over static stability. Our research focuses on navigating rough terrain with heavy loads, prioritizing stability over speed. Hence, we concentrate on implementing static gaits. The selected static gait is the crawl gait, known for its superior stability in quadruped applications.

3.1 Crawl gait planning

Two approaches were evaluated: utilising three feet, or all four feet in contact with the ground during forward motion. After careful analysis, the decision was made to adopt the approach of employing all four feet.

One critical factor that had to be considered was the inertial forces exerted during movement. These forces significantly affect stability, particularly when traversing slopes or during motion initiation and termination. The limited processing power of the developed platform also posed a constraint on the implementation of complex control algorithms that could account for such factors. Given these limitations, ensuring stability with only three feet in contact with the ground during forward motion was considered infeasible.

By employing all four feet in contact with the ground during forward motion, a larger support polygon was created, resulting in enhanced stability. This larger support polygon allowed for better weight distribution and improved balance, mitigating the risk of tipping or losing stability. Additionally, the inclusion of an extra foot in contact with the ground during the crawl gait minimised the effect of inertial forces on stability, particularly during critical moments like motion initiation or termination, and when traversing slopes. Therefore, by utilising all four feet in the support phase, the crawl gait ensured a higher degree of stability, making it well-suited for rough terrain locomotion and carrying heavy loads.

The crawl gait sequence followed a specific pattern: front left, rear right, front right, and rear left, with one foot transitioning to the swing phase while the other three feet maintained contact with the ground. Balance was achieved through weight transfer, and forward motion was initiated only when all four feet were in the support phase.

3.2 Foot trajectory planning

The above pattern, however, only described the sequence in which the legs of the robot were to move during locomotion and did not contain specifics regarding individual leg movement. This was determined by the desired trajectory that a foot needed to follow.

Successfully handling challenges such as stepping over obstacles or navigating underneath obstructions required the robot to readily adapt its foot trajectory and the positioning thereof. The crawling gait pattern, ensuring stability, thus remained the same, with the foot trajectory being adapted to ensure optimal foot placement and movement through space.

3.2.1 Trajectory Generation

The foot trajectory of a legged robot can be described as the path, in three-dimensional space, along which a foot travels when completing a step. Two attributes of the gait can be controlled by adapting this path: The distance that the robot travels forward in a single step (*StepLength*), and how high the foot is lifted from the ground during the swing phase (*StepHeight*). Where this path is situated, relative to the body of the robot, can be referred to as the position of the foot trajectory. Adapting this position allows the robot to control its height from the ground during locomotion (*RideHeight*). Figure 3 illustrates a foot trajectory, with the parts of the trajectory relating to the *StepLength*, *StepHeight* and *RideHeight* indicated.

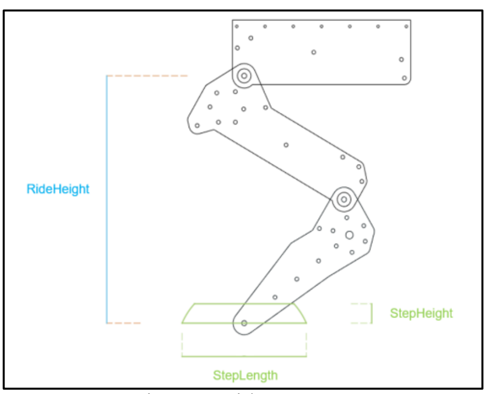

Fig. 3. Foot trajectory with parameters.

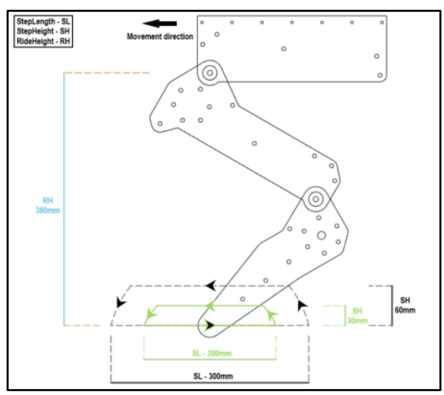

Fig. 4. Changing foot trajectory.

To generate foot trajectories at low computational cost, an approach was formulated based only on the three parameters mentioned above (*StepLength*, *StepHeight* and *RideHeight*). Trajectory generation was split into two parts, namely the Support phase and the Swing phase. The support phase being the part of the trajectory where the foot is in contact with the ground, and the swing phase being the part of the trajectory where the foot lifts from the ground, swings forward, and then return to ground level. The support phase had to satisfy the *StepLength* and *RideHeight* parameters, while the swing phase had to satisfy the *StepHeight* parameter. The two phases together forming the complete foot trajectory. Figure 4 shows how the generated foot trajectory changes when varying the *StepLength* and *StepHeight* parameters. The *RideHeight* parameter only affects the position of the trajectory and not its shape. This approach allowed for the implementation of real-time foot trajectory generation on the developed platform as a maximum of three parameters had to be varied to adapt the trajectory, thus keeping the required computational power to a minimum.

3.2.2 Trajectory Implementation

To control the foot position, the Femur and Tibia angles had to be adjusted. The developed method allowed for a low computational cost implementation of the foot trajectory, requiring calculations of only three angle values. The Cartesian coordinate system (*X*-*Y*) was used to describe the position of the foot, as seen in Figure 5.

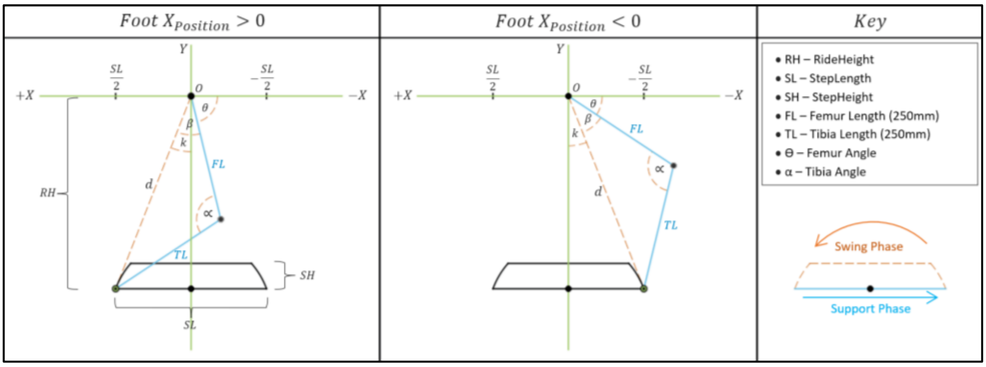

Fig. 5. Foot positioning coordinate system.

The origin, indicated with "O", being where the Femur connected to the Hip. The *X*-*Y* coordinates of the starting point and end point of the support phase was required to calculate the Femur's *Start* and *End* angle. The vertical position or *Y*-coordinate of the support phase was determined by the *RideHeight* parameter and could thus not change. The horizontal or *X* position of the support phase, however, was dependant on the position of the foot trajectory relative to the origin ("O"). The foot trajectory was thus positioned with the centre point of the support phase located at $X = 0$, as seen in Figure 5. This was done to minimise the change in foot height when taking a step. Consequently, minimising the change in Tibia angle required to maintain the *RideHeight* during locomotion. With the foot trajectory now positioned and hence the coordinates of the starting and end point of the support phase known, the Femur's *Start* and *End* angle could be calculated using inverse kinematics.

$$d = \sqrt{\left(\frac{SL}{2}\right)^2 + RH^2} \tag{1}$$

$$\beta = \cos^{-1}\left[\frac{d}{2(FL)(TL)}\right] \tag{2}$$

$$k = \cos^{-1}\left(\frac{RH}{d}\right) \tag{3}$$

$$\theta_{Start} = \frac{\pi}{2} - (\beta - k) \tag{4}$$

$$\theta_{End} = \frac{\pi}{2} - (\beta + k) \tag{5}$$

During the support phase, the Femur thus started at the calculated *Start* angle and rotated until it reached the *End* angle. As the Femur rotated, the Tibia angle had to be constantly adjusted to maintain the specified *RideHeight* and hence make the foot follow the desired trajectory. The required Tibia angle was calculated as follows.

$$\alpha = \sin^{-1}\left[\left(\frac{RH}{TL}\right) - \sin(\theta)\right] + \theta \tag{6}$$

At the end of the support phase, the foot had to lift from the ground and transition to the swing phase. To lift the foot, both the Femur and Tibia was set to retract. As the Femur and Tibia retracted, their respected angles were used to constantly calculate the foot height as it decreased. This was done using forward kinematics.

$$FootHeight = FL\sin(\theta) + TL\sin(\alpha - \theta) \tag{7}$$

Once the foot reached the specified *StepHeight*, the Femur was set to rotate from its current angle to the *Start* angle. As the Femur rotated to the *Start* angle, the Tibia angle had to be constantly adjusted as with the support phase. The only difference now being that it had to be adjusted to maintain the specified *StepHeight* and hence make the foot follow the swing phase of the desired trajectory. The required Tibia angle was calculated as follows.

$$\alpha = \sin^{-1}\left[\left(\frac{RH - SH}{TL}\right) - \sin(\theta)\right] + \theta \tag{8}$$

At the end of the swing phase, the foot had to transition back to the start of the support phase. This was done by simply changing the height parameter that had to be satisfied from *StepHeight* back to *RideHeight*.

Throughout this sequence, the linear actuators were turned on or off, with the encoder values being measured. Once the link had reached its desired position, the actuators were turned off. Due to the slow nature of the actuators, a bang-bang controller was deemed sufficient.

4 Testing and Results

To test the gait generation ability, the robot was made to traverse the two environments discussed in Section 2. The robot was not manually adjusted in any way during testing. The only human interaction was through the controller to test the gait generation by adjusting the three parameters and providing the robot with a pitch angle during ascent and descent. This all took place in real-time. Currently the robot is tele-operated, with future work focusing on automating the adjustment of these parameters through sensor feedback and vision algorithms. The recorded results are discussed below.

4.1 Environment 1: Level ground walking

Using the controller, the user set the parameters to the following values: *RideHeight* = 0.4 m, *StepLength* = 0.3 m, *StepHeight* = 0.025 m. These values were chosen to ensure that the feet

Fig. 6. Level ground walking.

would bump into the obstacle if the gait didn't successfully change to step over it. The user then initiated forward motion (walking) by pressing the joystick on the controller forward. The robot can be seen to move forward and remain balanced, as seen in Figure 6.

4.2 Environment 1: Clearing obstacle

When the frontmost foot of the robot reached a distance of 0.3 m to the obstacle, the user adjusted the *StepLength* parameter to 0.2 m. This was done to move the robot closer to the obstacle as it was still too far away to successfully step over it whilst simultaneously ensuring that the foot wont step too far and hit the obstacle. To step over the obstacle (0.05 m x 0.05 m) the user adjusted the *StepHeight* parameter to 0.06 m. These parameters were chosen by trial and error.

With the robot stepping over the obstacle successfully it could be concluded that the gait change happened as planned. Figure 7 shows the robot stepping closer to and clearing the obstacle. Balance was maintained throughout.

Fig. 7. Robot navigating over obstacle.

4.3 Environment 1: Incline walking

Maintaining balance becomes significantly more challenging when navigating slopes, whether ascending or descending. This difficulty arises from the slope's impact on the robot's centre of gravity, causing an offset. The subsequent sections discuss the measures undertaken to minimise this offset's effects and present the results obtained while traversing a slope with a pitch angle of 15º.

4.3.1 Pre slope adjustments

Before reaching the start of the incline, the user adjusted the parameters as follows: *RideHeight* = 0.35 m and *StepLength* = 0.15 m. These parameters were chosen by trial and error. The *RideHeight* parameter was decreased to bring the robot's centre of mass closer to the ground. Doing so increased the stability of the robot and decreased the offset in centre of gravity that would be experienced once traversing the incline. The *StepLength* parameter was decreased as previous tests have shown that taking smaller steps when traversing a slope improved overall stability and decreased chances of the feet slipping. The robot successfully adjusted its gait whilst retaining forward motion towards the start of the incline.

4.3.2 Incline

Reaching the start of the incline, the front two feet successfully stepped onto the slope, consequently pitching the body of the robot. This is shown in Figure 8.

Fig. 8. Front feet stepping onto incline.

Taking the user provided pitch angle into account, the robot adjusted its gait and successfully traversed up the incline. This can be seen in Figure 9. The robot remained balanced and showed good stability. Traction was also good as no slipping was observed.

Fig. 9. Robot traversing incline.

The pitching of the body caused the centre of gravity to shift towards the rear of the robot. To maintain stability, the foot trajectory thus also had to shift backwards. The user provided pitch angle was used to calculate the magnitude of the shift in centre of gravity. Which in turn determined the amount that the foot trajectory had to shift. The required shift in the positioning of the foot trajectory was realized by simply incrementing the *Start* and *End* coordinates (refer to Section 3.2) of the support phase of the foot trajectory.

4.4 Environment 2: Decline to level ground walking

The same parameter values as with the incline walking was used for the decline to level ground walking. Figure 10 shows the platform successfully descending the slope and moving onto level ground. Leading to the conclusion that the gait changes, taking the provided pitch angle into account, occurred as expected.

Fig. 10. Robot traversing decline that transitions to level ground.

Balance was maintained throughout, and overall stability was good. Traction was also good as no slipping was observed.

5 Conclusion

In this paper a gait generation method for real-time quadrupedal locomotion capable of executing on low-end hardware was developed. A quadrupedal robot was also developed to test the gait generation method on a physical platform. Several tests were successfully performed that included navigating over obstacles and traversing slopes. The results obtained were very promising and showed that the robot could adjust its gait in real-time, while maintaining stability at all times and in all testing scenarios. This was achieved while employing a partial open-loop architecture. Overall, gait generation and implementation was performed in real time on a Teensy 4.0 micro controller, showing the low computation requirements of the algorithms.

References

1. H. Gao, J. Li, Y. Wan, H. Yu, C. Zhou, *A real-time planning and control framework for robust and dynamic quadrupedal locomotion, Journal of Bionic Engineering*, (2023)
2. T. Attia, A.M. El-Dalatony, H. Ragheb, A.M. Sharaf, *Cascaded PID Trajectory Tracking Control for Quadruped Robotic Leg,* International Journal of Mechanical Engineering and Robotics Research, **12**, 1, (2023)
3. C. Fisher, *Trajectory Optimisation Inspired Design for Legged Robotics*, Faculty of Engineering, and the Built Environment of Department of Electrical Engineering, (2021)
4. S. Seok, A. Wang, M.Y. Chuah, J.H. Dong, L. Jongwoo, D.M. Otten, J.H. Lang, K. Sangbae, *Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot*. IEEE/ASME transactions on mechatronics. **20**, 3, (2015)
5. R. Niiyama, A. Nagakubo, Y. Kuniyoshi, *Mowgli: A Bipedal Jumping and Landing Robot with an Artificial Musculoskeletal System*, In Proceedings IEEE International Conference on Robotics and Automation, (2007)
6. M. Hutter, C. Gehring, D. Jud, A. Lauber, C.D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, *ANYmal - a highly mobile and dynamic quadrupedal robot*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2016)
7. M. Ramezani, M. Brandao, B. Casseau, I. Havoutis, and M. Fallon, *Legged robots for autonomous inspection and monitoring of offshore assets*, Offshore Technology Conference, (2020)
8. C. D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Gunther, M. Tranzatto, P. Fankhauser, and M. Hutter, *Advances in real-world applications for legged robots*, Journal of Field Robotics, **35**, 8, (2018)
9. K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, and S. Kawatsuma, *Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots*, Journal of Field Robotics, **30**, 1, (2013)
10. A. Agha, K. L. Mitchell, and P. J. Boston, *Robotic Exploration of Planetary Subsurface Voids in Search for Life*, AGU Fall Meeting Abstracts, (2019)
11. H. Kolvenbach, D. Wisth, R. Buchanan, G. Valsecchi, R. Grandia, M. Fallon, and M. Hutter, *Towards autonomous inspection of concrete deterioration in sewers with legged robots*, Journal of Field Robotics, (2020)
12. A. Cully, J. Clune, D. Tarapore, *Robots that can adapt like animals*, Nature **521**, (2015)
13. M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, *Bigdog, the rough-terrain quadruped robot*, IFAC Proceedings Volumes, **41**, 2, (2008)
14. L. Raw, C. Fisher, A. Patel, *Effects of Limb Morphology on Transient Locomotion in Quadruped Robots*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2019)