GAIT SIMULATION OF AN HEXAPODAL ROBOT WALKING ACROSS VARIOUS TERRAINS

Antoine Sion[1], Lionel Birglen[1], Alain Vande Wouwer[2]
[1]*Robotics Laboratory, Department of Mechanical Engineering, Polytechnique Montréal, Montréal, QC, Canada*
[2]*Systems, Estimation, Control and Optimization, University of Mons, Mons, Belgium*
Email: antoine.sion1310@gmail.com; lionel.birglen@polymtl.ca ; alain.vandewouwer@umons.ac.be

ABSTRACT

This paper presents the simulation of different gaits of an hexapodal robot when it is traversing a range of terrains with various geometries. At first, the specific architecture of the robot considered here and its components are described. Then, to prepare for the real life implementation of a gait algorithm on this robot, simulations are run to establish its capability to efficiently traverse terrains. To this aim, the robot as well as various terrains are implemented on a robotic software package, namely CoppeliaSim. How the software works and in particular the specificity of the selected physics engine are briefly discussed. Then, a gait algorithm is proposed for which an exact solution to the inverse kinematic problem exist. Subsequently, simulations are performed and results discussed. The main metric used in this paper to measure the performance of the gait is the energetic cost of travel from two fixed points. This metric is used to deduce optimal geometric parameters of the gait, i.e. body and step heights as well as the radius of the contact circle of the feet on the ground. The main terrains simulated in this work are a flat surface and an inclined plane. The latter is used to evaluate the maximal angle of the slope that the robot can safely climb. However, another final simulated terrain consisting of a rough random surface, mimicking a rocky terrain, is also used to illustrate the limits of a fixed gait algorithm.

Keywords: Numerical simulation; hexapodal robot; inverse kinematics.

SIMULATION DE DÉMARCHES D'UN ROBOT HEXAPODE TRAVERSANT DIFFÉRENTS TERRAINS

RÉSUMÉ

Cet article porte sur la simulation de démarches pour un robot hexapode quand ce dernier traverse des terrains avec des géométries variées. Dans un premier temps, le robot considéré ici ainsi que les composants le constituant sont décrits. Ensuite, afin de préparer l'implémentation d'un algorithme de démarche sur ce robot, des simulations logicielles sont utilisées et permettent de mesurer la capacité de celui à traverser efficacement ces terrains. Le robot ainsi que les terrains sont ainsi implantés dans le logiciel de simulation robotique CoppeliaSim. Comment ce logiciel fonctionne et en particulier les caractéristiques du moteur de simulation physique choisi pour les simulations discutées ici sont aussi brièvement présentés. Un algorithme de démarche est ensuite développé pour lequel une solution simple et unique au problème géométrique inverse existe. Finalement, des simulations sont conduites et leurs résultats sont discutés. L'indice de performance choisi pour mesurer l'efficacité des démarches est le coût énergétique du transport entre deux positions fixes des terrains. L'étude de cet indice permet de fixer un ensemble de paramètres géométriques optimaux pour la démarche comme par exemple, la hauteur du corps du robot et de la phase de vol de la trajectoire de son pied ainsi que le rayon du cercle des points de contact au sol. Parmi les terrains simulés, en plus d'une simple surface plane, des pentes inclinées seront utilisées afin de déterminer la pente maximale que le robot peut traverser. Finalement, un autre type de terrain simulé présentant une surface accidentée et simulant un terrain rocailleux permettra de montrer les limites d'une démarche fixe.

Mots-clés : Simulation numérique ; robot hexapode ; cinématique inverse.

1. INTRODUCTION

Legged robots are a type of moving machines similar in function to the more common vehicles using wheels or tracks. Legs offer advantages and disadvantages as discussed in [1]. Mainly, walking robots have more degrees of freedom (DOF) than wheeled ones and thus, aren't limited to (relatively) flat terrains. They are also often less destructive for their environment because they rely only on discrete points of contacts with the latter where wheeled/tracked robots typically leave large traces after their passage. Additionally, legged robots commonly have sensors on their feet to gather information about contact properties with the ground and can use the latter to improve their control, typically focusing on balance robustness. However, the complexity of the controller of a legged robot is significantly greater because of this increase in DOF, the multiple actuators used, and the necessity for accurate coordination between the legs. Energy consumption of walking robot is generally also an issue because the potential energy of the mechanism is changing, even on flat terrains, and there are numerous phases of accelerations and decelerations of the actuated joints. Finally, the walking speed of a legged robot and its payload are both usually lower than that of a similar size wheeled vehicle. Thus, legged robots are better suited for rescue or exploration missions on rough terrains where agility is key and this is the focus of most recent works in the literature. For example, the Teleman European project [2] focused on the development of this type of robots for nuclear maintenance.

Looking at the literature on walking machines, it is clear that it has been a topic of interest for many decades. Walking mechanical contraptions have been created since at least the 19th century, e.g. the famous Chebyshev's machine [3] in 1850. These machines weren't robots per se as they lacked programmable controllers but were the obvious precursors to more modern development starting with the General Electric Walking Truck from 1969 [4] (also using force feedback). However, more significant progress arose with the rapid development of modern computers which allowed for more complex control strategies. One of the first walking robot with a numerical control was the Ohio State University Hexapod shown in 1976 [5].

More recently, many research teams have been developing their own robots due to the still decreasing cost of actuators and massive democratization of 3D printing. Many examples can be found in modern literature such as the probably most well-known of legged robots: the Big Dog (2004) and Spot (2019) from Boston Dynamics. Prof. Hirose from the Tokyo Institute of Technology also led the development of several innovative legged robots since the 1980's such as Titan IV, Titan VIII or Titan XI [6] robots. In Germany, the Lauron V (2014) [7] is the latest robot designed by the team from FZI Karlsruhe while the Scorpion robot was produced by the university of Bremen. Previous works from the authors of this work also introduced mechanical legs which are able to adapt their kinematic structures when colliding with obstacles to overcome the latter [8–10].

In this paper, a robot based on a hexapodal (6 legs) architecture is studied and the impact of its gait parameters on its energy comsumption will be precisely quantified. An hexapodal design is very common for hobby walking machines and readily commercially available. However, in our case, the robot is lacking a controller and therefore, a control algorithm must be designed to coordinate the motion of the legs to produce efficient walking patterns. To this aim, a simulation approach is used to test different algorithms and how well they perform. There are many software available to simulate robotic systems [11, 12], and the most well-known are probably CoppeliaSim (previously known as V-REP), Gazebo, and ARGoS. CoppeliaSim is used in this paper because it allowed more flexibility and options compared to the others. CoppeliaSim also supports multiple physics engines, mesh manipulation, and the user can interact with the environment during simulation. However, one of its limitation is that it is impossible to automate simulations while changing parameters because all editing has to be done through the software's interface. It also performs poorly when simulating a large number of robots compared to Gazebo or ARGoS but the latter is not an issue for the task at hand since a single robot is simulated in this paper.

2. PRESENTATION OF THE ROBOT

The robot discusses in this paper has six identical legs and was used during filming of the movie "Eye On Juliet", directed by Kim Nguyen and produced by the Montreal based studio Item7. The robot was donated to Polytechnique Montréal to be reused and serve as a platform for research. The robot is illustrated in Fig. 1.

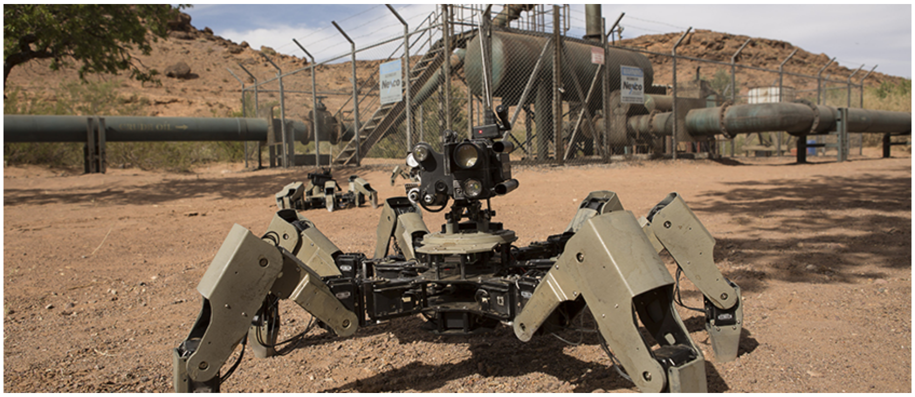


Fig. 1. The hexapodal robot from Eye On Juliet (picture from [13])

Each of the six legs is constituted by four independent revolute joints, as depicted in Fig. 2, actuated by an on-axis servomotor. The links between these joints are called the coxa, femur, tibia and tarsus respectively, starting from the body of the robot and descending to the foot of the leg. The 24 servomotors in the all the legs are identical: the MX106T model from Robotis. This servomotor embeds a microcontroller for low level control as well as an encoder for position control. The servomotors of each leg are daisy chained through a simple line bus topology and then, connected to a main controller, the ODROID-C1, overall creating a star topology. This configuration is helpful because a command can be sent to all of the motors with only a single instruction passed on the bus from one device to the next. All the motors and the controller are powered by a Li-Po battery pack.

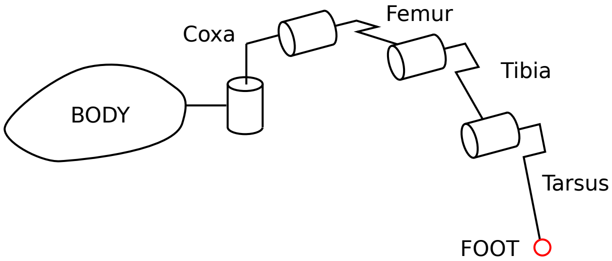


Fig. 2. Kinematic model of a leg

3. GAIT ALGORITHM

3.1. Introduction

Since the robot was custom built for the movie, it does not correspond to a standard architecture commercially available for which a CAD model is available. Thus, the important geometric parameters of the robot were physically measured, modeled in Solidworks, and imported as STL files in CoppeliaSim. CoppeliaSim allows its user to select among four different physics engines: Bullet, ODE, Vortex Studio, and Newton. Bullet and ODE have both been developed primarily for video games and rely on an iterative solver, Newton is also sometimes used for video games but uses a deterministic solver which gives more accurate results. Finally, Vortex Studio is an engine specialized in simulations of land or sea equipment and their environments. Vortex Studio is selected in this work because of the many modifiable properties it has compared to the other engines. It is used in industry and research for testing mobile robots, vehicles, or even virtual training simulators for operators.

A fixed gait algorithm is chosen here as a simple trajectory planner for the leg to move the robot. The definition of this gait can be split in two parts: first, the gait of the hexapod in itself must be produced, i.e. where it must place its feet and with which sequence; then, these positions must be translated into joint angles to be controlled by the servomotors.

3.2. Gait generation

What propels the robot in a given direction is the motion of the feet which can be divided in two phases: a support and transfer phases as illustrated in Fig. 3. The two extreme points of this trajectory along the horizontal direction are called the Anterior Extreme Position (AEP) and the Posterior Extreme Position (PEP) as defined in [1]. During the support phase, the foot moves in the opposite direction of the walking direction of the hexapod, providing a contact point with the ground and propelling the robot forward. During the transfer phase, the leg rises from the ground to go to another contact point further away in the walking direction. The trajectory of the support phase is ideally linear to displace the body of the robot in a straight line forward. A straight line of the foot translates into a straight horizontal translation of the robot body (and its center of mass) which in theory requires no energy since the potential energy of the body is then constant during this phase. As for the transfer phase, its geometry can be chosen more freely, in this paper an ellipse was selected since its mathematical representation is simple.

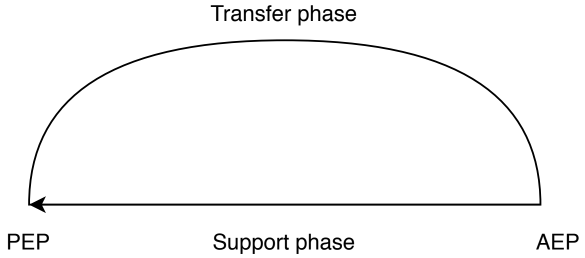
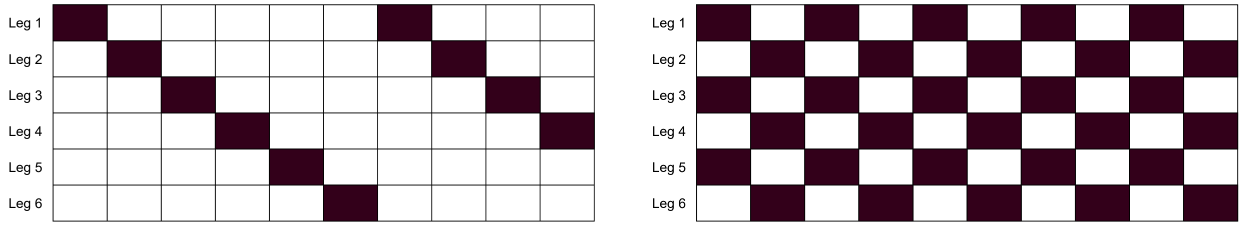


Fig. 3. Motion of a leg

The durations of the transfer and support phases are referred to as t and s respectively. Thus, the cycle time of a moving leg is $T = s + t$. A duty factor $\beta = \frac{s}{T}$ can be defined that will be used to characterize different gaits for the robot as discussed in [1]. The legs alternate between the two phases following a sequence thereby producing a chosen gait of the robot. Two types of gaits amongst the most common from the literature are the wave gait and the tripod one. To switch between these gaits, β can be set to a specific value, for instance $\beta = \frac{5}{6}$ for the wave gait and $\beta = \frac{1}{2}$ for the tripod gait. For example, with a cycle time of 6 seconds for a wave gait, when a leg has a duty factor of $\frac{5}{6}$, it spends 5 seconds on the ground and 1 second in the air. By alternating the start of each cycle for each leg, only one leg is in the air while all the other ones are touching the ground. With the same logic, for a tripod gait, each leg thus spends 3 seconds in the air and 3 seconds on the ground. By synchronising the start of the cycles of 3 legs, the hexapod is walking at all times with 3 legs in the air and 3 legs on the ground. A wave gait makes the legs alternate their transfer phases one after the other while the tripod gait allows two pairs of three legs to move synchronously. An illustration of the sequence of phases for each leg can be found in Figs. 4a and 4b.

The determined leg positions from the selected gait, given in the frame of reference of the body of the



(a) Wave gait: black rectangles represent the transfer phase

(b) Tripod gait: black rectangles represent the transfer phase

Fig. 4. Visual illustration of gait phases' sequences

robot, must be transformed into the frame of reference of each leg to compute the inverse kinematics of the linkage and thus, to produce the angular position commands for the joints. This can be done using simple homogeneous transformation matrices. If the position of a leg in the frame of reference of the body is denoted $\left[\mathbf{r}_{P/0}\right]_0$, the coordinates of this vector are related to the same position expressed in the frame of reference of the leg i $\left[\mathbf{r}_{P/i}\right]_i$ by:

$$\begin{pmatrix} \left[\mathbf{r}_{P/0}\right]_0 \\ 1 \end{pmatrix} = \mathbf{T}_{0,i} \begin{pmatrix} \left[\mathbf{r}_{P/i}\right]_i \\ 1 \end{pmatrix} \tag{1}$$

where $\mathbf{T}_{0,i}$ is the classical homogeneous transformation matrix [14] between the associated frames.

3.3. Inverse kinematics of a leg

The inverse kinematics of the robot legs consists in finding the four joint angles for a given endpoint (foot position). Namely, computing the four configuration parameters of a leg (q_1, q_2, q_3, q_4) to place the tip of this leg in a position X, Y, Z expressed in the leg reference frame. This inverse kinematic problem can be solved either algebraically or numerically. An exact algebraic closed form solution is preferable for an implementation in real-time to minimize processing time but also produces a solution in a fixed number of operations. Since the leg has four degrees of freedom, an exact solution can be found by reducing the problem to three degrees of freedom as shown in [15–18]. A condition must therefore be added to solve the redundancy or else there will be infinitely many solutions to choose from. Here, the problem is first solved for 3 configuration parameters without the tarsus of the leg. It can be decomposed in two planes with the first joint rotating in the first plane (X,Y) and the two other joints rotating in a plane perpendicular to the first one. A top view of the first plane can be seen in Fig. 5a.

The angle γ is equal to the first joint angle and represents the first configuration parameter. It only depends of X and Y, i.e.:

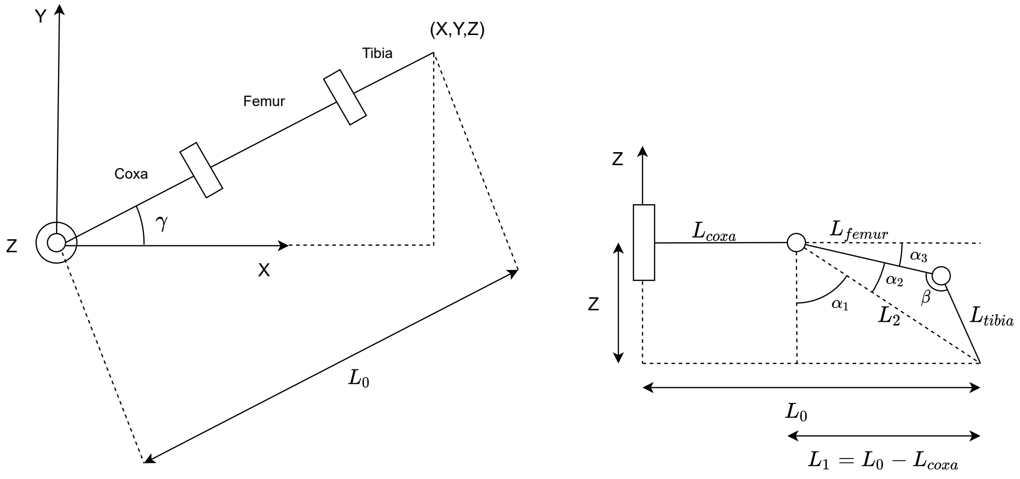
$$q_1 = \gamma = \arctan\left(\frac{Y}{X}\right). \tag{2}$$

A side view of the plane of the last three joints of the leg is shown in Fig. 5b. With the different parameters illustrated in this last figure, the geometry of the leg can be established. One can define L_0 as the length from the start of the leg to the point of contact $(X,\ Y,\ Z)$ projected on the $(X,\ Y)$ plane. From the same triangle shown in Fig. 5a and used to find Eq. (2), the Pythagorean theorem gives:

$$L_0 = \sqrt{X^2 + Y^2}. \tag{3}$$

With L_2 the length from the end of the coxa to the point of contact and L_1 its projection on the $(X,\ Y)$ plane, one has:

$$L_2 = \sqrt{L_1^2 + Z^2} \tag{4}$$



(a) First plane

(b) Second plane

Fig. 5. Geometric planes of the linkage

and α_1 (see Fig. 5b) is expressed as:

$$\alpha_1 = \arctan\left(\frac{L_1}{Z}\right). \tag{5}$$

Then, the law of cosines applied to the triangle with lengths L_{femur}, L_2, and L_{tibia} yields:

$$L_{tibia}^2 = L_2^2 + L_{femur}^2 - 2L_2L_{femur}cos(\alpha_2) \tag{6}$$

which can be reordered into:

$$\alpha_2 = \arccos\left(\frac{L_{tibia}^2 - L_2^2 - L_{femur}^2}{-2L_2L_{femur}}\right). \tag{7}$$

The angle α_3 can then be found from its complementary angles α_1 and α_2:

$$q_2 = \alpha_3 = \frac{\pi}{2} - (\alpha_1 + \alpha_2). \tag{8}$$

Once again, the law of cosines applied to the same triangle as before but this time with the angle β yields:

$$\beta = \arccos\left(\frac{L_2^2 - L_{femur}^2 - L_{tibia}^2}{-2L_{femur}L_{tibia}}\right) \tag{9}$$

from which the last configuration parameter is deduced:

$$q_3 = \pi - \beta. \tag{10}$$

The leg being a classical RRR linkage there exists another symmetric solution to this inverse problem as illustrated in Fig. 6. Eqs. (8) and (10) in this second solution becomes:

$$q_2 = \frac{\pi}{2} - (\alpha_1 + \alpha_2) + 2\alpha_2 \tag{11}$$

$$q_3 = -(\pi - \beta). \tag{12}$$

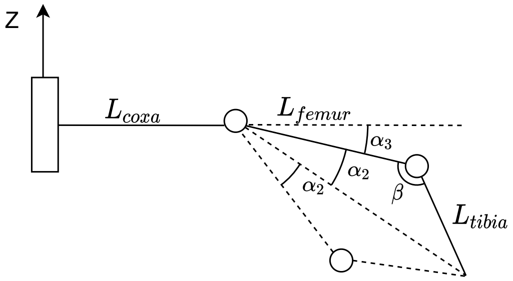


Fig. 6. Second solution

For the simulation, the first solution is chosen because the configuration of the leg is consistent with the literature of walking robots. Finally, to compute the fourth configuration parameter a condition is imposed, namely, that the tarsus stays parallel to the z-axis. This geometric constraint is chosen in the hopes of establishing a stronger contact between the leg and the ground. Then, one obtains:

$$q_4 = \frac{\pi}{2} - q_2 - q_3. \tag{13}$$

3.4. Algorithm

The gait trajectory that has been established to command the servomotors has to be sent to the latter at a given sample frequency. In CoppeliaSim, the default simulation time step is 50 ms, corresponding to a sampling frequency of 20 Hz, which was therefore selected for the default sampling time. This relatively low frequency should be easy to transpose on the real robot even by taking into account the processing time of the controller and the expected communication delays between the controller and the motors.

The gait algorithm in itself can be divided in two parts (see Fig. 7): first, the initialization of the robot and then, the main loop. During the initialization, all the variables and parameters needed to place the robot in a standing stance are set by sending the calculated rest positions of the legs to all the motors. The initial positions of the feet are chosen to be on a circle whose center coincides with the one of the body but the radius of that former circle is a variable whose influence on the walking efficiency of the robot will be studied. Other parameters such as the height of the robot's body, the step height, and the cycle time are also input at this point. The duration of the support phase is selected and the duration of the transfer phase is then obtained with:

$$t = s\left(\frac{1-\beta}{\beta}\right). \tag{14}$$

The main loop is then called at each simulation step. The speed is chosen by the user and then, the state of each leg is verified by a boolean that switches states when the phase changes. If a leg is in support phase, the next position $\mathbf{P}_i^{k+1}$ is recursively computed with [19]:

$$\mathbf{P}_i^{k+1} = \mathbf{R}_z\left(\frac{\omega_z}{f}\right)\left(\mathbf{P}_i^k - \frac{\mathbf{V}}{f}\right) \tag{15}$$

The desired speed $\mathbf{V}$ normalized by the frequency f of the simulation is first subtracted from the current position. Then, the homogeneous transformation matrix $\mathbf{R}_z$ is multiplied with the new position to rotate along the z axis. This homogeneous matrix is using a rotation angle $\frac{\omega_z}{f}$ because the rotation speed is also normalized by the frequency f. However, if the leg is in the transfer phase, the next position is simply taken from the pre-calculated elliptic trajectory at the given time. Then, the inverse kinematic model detailed in

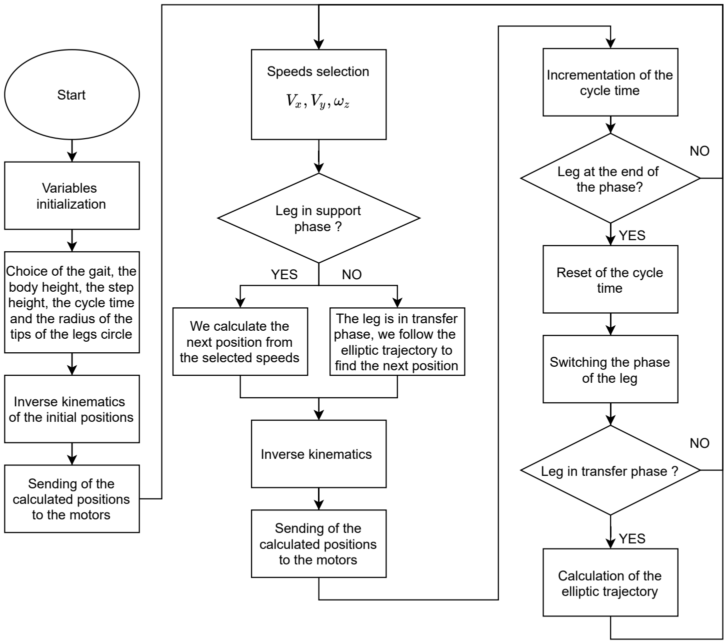

Start
Variables initialization
Choice of the gait, the body height, the step height, the cycle time and the radius of the tips of the legs circle
Inverse kinematics of the initial positions
Sending of the calculated positions to the motors
Speeds selection
V_x, V_y, ω_z
Leg in support phase ?
YES
NO
We calculate the next position from the selected speeds
The leg is in transfer phase, we follow the elliptic trajectory to find the next position
Inverse kinematics
Sending of the calculated positions to the motors
Incrementation of the cycle time
Leg at the end of the phase?
NO
YES
Reset of the cycle time
Switching the phase of the leg
Leg in transfer phase ?
NO
YES
Calculation of the elliptic trajectory

Fig. 7. The fixed gait algorithm

Section 3.3 can be used to get the configuration parameters and send appropriate control input to the motors. Finally, the state of each leg is again verified: if a leg is at the end of its phase, the algorithm switches from this phase to the next. If the next phase is the transfer phase, the elliptic trajectory that the leg will follow is calculated, etc.

4. TESTS AND RESULTS

4.1. Introduction

CoppeliaSim allows for PID position control of the robot joints with both a maximal speed and torque to be set. In the simulations, this maximum speed was 45 rpm and the maximum torque 5 Nm to match the specifications of the real servomotors. The proportional, integral and derivative parameters are respectively: $K_P = 0.2$, $K_I = 0.05$ and $K_D = 0$. Typical resulting gaits can be seen in the following videos:

- tripod gait: `https://youtu.be/A2j3-6F_7K4`
- wave gait: `https://youtu.be/vpdVf7q5l8A`

The simulation results in this section have been obtained for a walking speed of 10 mm/s in one direction on a flat terrain and a total cycle time of 6 seconds while the robot is following a wave gait. The speed and cycle time were chosen arbitrarily because the goal was to see the evolution of the sum of the RMS powers of the motors in function of the geometrical parameters. Since one point on the graphs (see the next sections) represents one simulation of approximately 5 to 10 minutes, only the wave gait was considered. The friction coefficient between the legs and the ground was set to a value of 1, which represent a contact with relatively high friction.

4.2. Optimal Height of the Robot Body

To find the optimal height of the body of the robot, the sum of the RMS powers of all the motors during the travel were measured while varying the height of the robot and the step height. This travel corresponds to one full cycle of the gait, i.e. when each of the legs has completed a single full step. The radius of the circle of the initial positions of the legs was kept at 350 mm. The resulting power consumption can be seen in Fig. 8. The duration of the cycle being set at six seconds for all cases, this sum of the RMS powers is directly proportional to the energy required for the displacement.

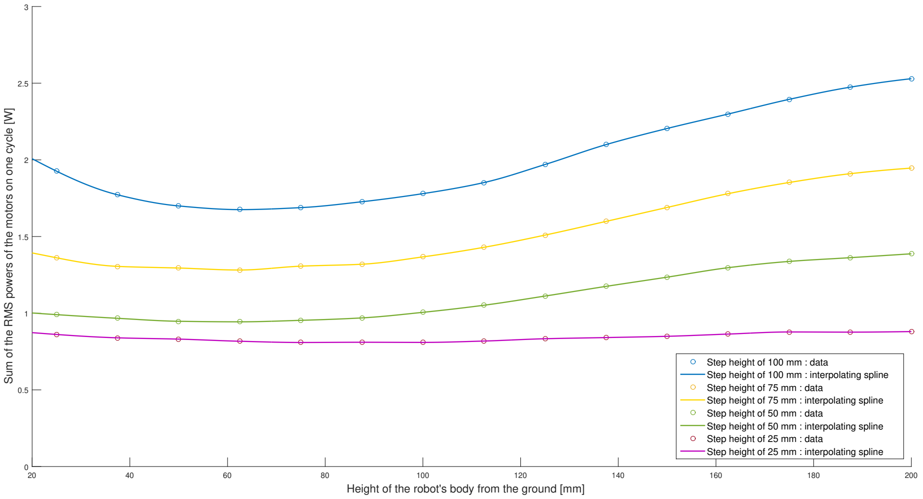


Fig. 8. Sums of the RMS powers of the motors during one cycle while varying the height of the robot

As can be seen, the required power is reduced with the step height. Indeed, a smaller step height means a smaller movement for the leg and thus, a reduction in the power needed. Each curve corresponding to a specific step height shows a point with minimum power, and noticeably these minima seem around 62.5 mm for most of these step heights. This optimal value can be explained by the fact that the configuration of the legs is changing with the body height. At the extreme heights of 25 and 200 mm, one of the four leg joints is significantly more solicited than the others while at the optimal body height, the power is evenly distributed between the joints thereby producing a more natural and efficient posture.

4.3. Optimal Placement of the Legs

The tips of the legs are positioned on a circle centered on the body of the robot which correspond to the start of the leg trajectory and at the end of each cycle, each leg is coming back to its initial position on this circle. As obtained in the previous section an optimal height of the body of 62.5 mm is selected and now, the power consumption is measured for different base circles and different step heights. The results are shown in Fig. 9.

A steep slope appears in the figure at the left of the plots. This sudden drop of power is due to mechanical interferences happening between the legs and the body which disappear when the circle radius is increased. These collisions explain the great augmentation of the power for small radii. It is therefore required to impose a minimal radius depending on the step height, e.g. 312.5 mm for a step height above 75 mm or 300 mm for a step height below 50 mm. These values are thus the optimal parameters. A small step height allows for a bigger radius because the legs are lifted further away from the body and do not collide with the latter. A small reduction in power can also be seen while the radius is decreased (right to left in the plots).

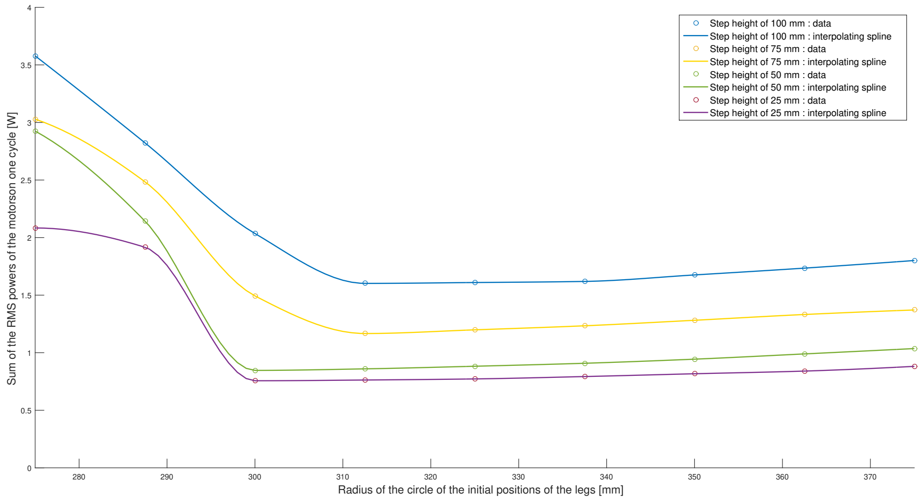


Fig. 9. Sums of the RMS powers of the motors during one cycle while varying the radius of the circle of the positions of the legs

4.4. Walking on a slope

The ability of the robot to climb a slope was also simulated with the same parameters obtained from the previous sections, namely a radius of the circle of the positions of the legs of 300 mm and a step height of 25 mm. The power required while varying the angle of the slope is shown in Fig. 10.

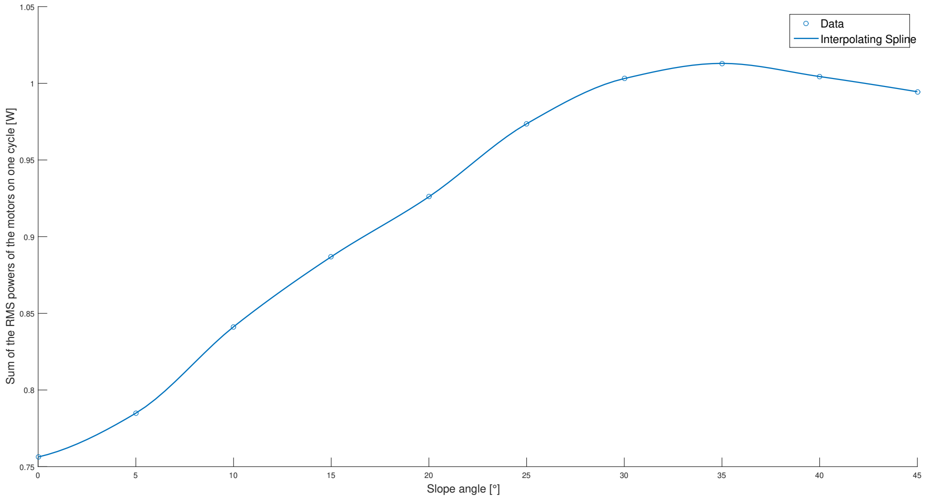


Fig. 10. Sums of the RMS powers of the motors during one cycle while varying the angle of the slope

The power needed is clearly rising with the slope as expected. This result is obvious because the servomotors have to generate more and more torque to oppose gravity when the slope gets steeper. The power is diminishing for 40° and 45° angles but only because the hexapod starts to slip at these angles. After 45°, the robot cannot actually climb the slope and falls. The maximum angle of a slope the robot can traverse is thus

45° with our parameters.

4.5. Walking on a Rough Terrain

The fixed gait algorithm was also experimented with a randomly generated terrain to see how well it would fare (an example of this terrain can be seen in Fig. 11). A matrix representing the altitude of a grid of points with an uniform distribution between 0 and the maximum desired altitude of the terrain was generated for different maximal altitudes. The latter were tried for 25 mm, 50 mm, 75 mm and 100 mm. What can be seen from these simulations is that the robot is able to keep a somewhat straight trajectory for 25 mm and 50 mm of maximum altitude but could not perform well for higher altitudes. This shows the limits of a fixed gait which works well on a flat floor but not so much on rough terrains due to the lack of feedback on its environment.

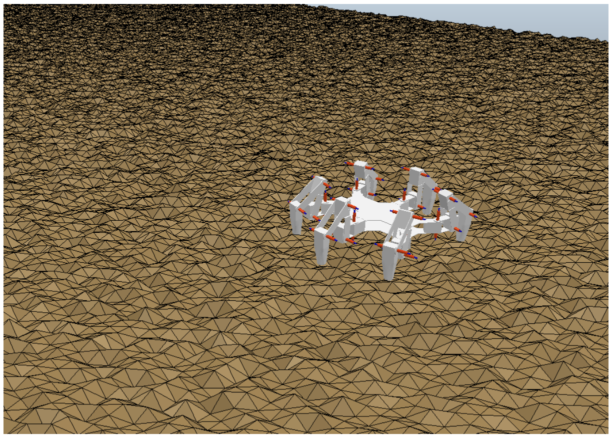


Fig. 11. Robot traversing a rough terrain with a maximum altitude of 50 mm

5. CONCLUSIONS

In this paper, a basic gait algorithm was developed to allow an hexapodal robot to efficiently move on a flat surface. It was possible to select the type of gait, the walking and rotation speeds as well as the duration of the leg's phases. A simple inverse kinematic model was used to get an exact solution for the positioning of the robot's 4 degrees of freedom legs to minimize processing time. The robot was simulated using CoppeliaSim and different virtual experiments were run to obtain optimal gait parameters minimizing power consumption. The impact on the latter of the principal parameters of the gait was established and precisely measured in the simulations. This work lays the bases for the control of the actual hexapodal robot prototype as will be shown in future works and the next step will be to validate the algorithm on the real robot.

ACKNOWLEDGEMENTS

The authors wish to thank Item7 and more specifically Jeanette Garcia for donating them the robot used in this paper. This donation had no influence on the results shown in this work which was written without interference.

REFERENCES

1. Alexandre, P. *Le contrôle hiérachisé d'un robot marcheur hexapode*. Ph.D. thesis, Université libre de Bruxelles, November 1996.
2. Robertson, B. "A preview of the european commission teleman programme for telerobotics research." *IEEE Robotics and Automation Magazine*, Décembre 1997.
3. "Chebyshev walking platform."
   URL `http://cyberneticzoo.com/walking-machines/1892-mechanical-horse-l-a-rygg-american/`
4. "Ge walking truck."
   URL `http://cyberneticzoo.com/walking-machines/1969-ge-walking-truck-ralph-mosher-american/`
5. Mcghee, R.B. and Iswandhi, G.I. "Adaptive locomotion of a multilegged robot over rough terrain." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 4, pp. 176–182. doi:10.1109/TSMC.1979.4310180, 1979.
6. Hirose, S., Fukuda, Y., Yoneda, K., Nagakubo, A., Tsukagoshi, H., Arikawa, K., Endo, G., Doi, T. and Hodoshima, R. "Quadruped walking robots at tokyo institute of technology design, analysis, and gait control methods." *Robotics Automation Magazine, IEEE*, Vol. 16, pp. 104 – 114. doi:10.1109/MRA.2009.932524, 07 2009.
7. Roennau, A., Heppner, G., Nowicki, M. and Dillmann, R. "Lauron v: A versatile six-legged walking robot with advanced maneuverability." doi:10.1109/AIM.2014.6878051, 07 2014.
8. Fedorov, D. and Birglen, L. "Design of a self-adaptive robotic leg using a triggered compliant element." *IEEE Robotics and Automation Letters*, Vol. 2, No. 3, pp. 1444–1451. doi:10.1109/LRA.2017.2670678, 2017.
9. Fedorov, D. and Birglen, L. "Geometric optimization of a self-adaptive robotic leg." *Transactions of the Canadian Society for Mechanical Engineering*, Vol. 42, No. 1, pp. 49–60. doi:10.1139/tcsme-2017-0010, 2018.
10. Fedorov, D. and Birglen, L. "Kinematic and potential energy analysis of self-adaptive robotic legs." In "ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference," American Society of Mechanical Engineers Digital Collection, 2018.
11. Pitonakova, L., Giuliani, M., Pipe, A. and Winfield, A. "Feature and performance comparison of the v-rep, gazebo and argos robot simulators." In M. Giuliani, T. Assaf and M.E. Giannaccini, eds., "Towards Autonomous Robotic Systems," pp. 357–368. Springer International Publishing, Cham, 2018.
12. Shamshiri, R., Hameed, I., Pitonakova, L., Weltzien, C., Balasundram, S., Yule, I., Grift, T. and Chowdhary, G. "Simulation software and virtual environments for acceleration of agricultural robotics: Features highlights and performance comparison." *International Journal of Agricultural and Biological Engineering*, Vol. 11, pp. 12–20. doi:10.25165/j.ijabe.20181103.4032, 01 2018.
13. URL `https://teaser-trailer.com/eye-on-juliet-movie/`
14. VERLINDEN, O. *Computer-Aided Analysis of Mechanical Systems*, february 2016.
15. S. Manoiu-Olaru, M.N. and Stoian, V. "Hexapod robot. mathematical support for modeling and control." *15th International Conference on System Theory, Control, and Computing (ICSTCC)*, pp. 1–6, october 2011.
16. Ollervides, J., Orrante-Sakanassi, J., Santibanez, V. and Dzul, A. "Navigation control system of walking hexapod robot." *2012 Ninth Electronics, Robotics and Automotive Mechanics Conference*, pp. 60–65, 11 2012.
17. Sun, J., Ren, J., Jin, Y., Wang, B. and Chen, D. "Hexapod robot kinematics modeling and tripod gait design based on the foot end trajectory." *2017 IEEE International Conference on Robotics and Biomimetics*, pp. 2611–2616, 12 2017.
18. Roy, S.S. and Pratihar, D. "Kinematics, dynamics and power consumption analyses for turning motion of a six-legged robot." *Journal of Intelligent and Robotic Systems*, Vol. 74, 06 2013.
19. Thilderkvist, D. and Svensson, S. "Motion control of hexapod robot using model-based design.", 2015.