Modelling, Trajectory Planning and Control of a Quadruped Robot Using Matlab®/Simulink™

Italo Oliveira[1], Ramiro Barbosa[2], and Manuel Silva[3(✉)]

[1] Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brazil
italo_fernando10@hotmail.com
[2] Instituto Superior de Engenharia do Porto, Porto, Portugal
rsb@isep.ipp.pt
[3] Instituto Superior de Engenharia do Porto and INESC TEC, Porto, Portugal
mss@isep.ipp.pt

Abstract. Due to the difficulty of building and making control tests in real robots, it is usual to first have a simulated model that provides a good approach of a real robot's behaviour. The importance of a good control system in execution of a planned trajectory inspired this work, whose purpose is to design a control system for a quadruped robot and test its performance.

Keywords: MATLAB® · Modelling · Trajectory Planning · Control system · Simulation · Quadruped robots

1 Introduction

Mobile robotics are an important tool in today's society, since many scientific researches rely on instruments that should move in inhospitable places for humans. For the stated reasons, it is necessary to create means of locomotion capable of withstanding several adversities, such as high temperature and pressure, steep and irregular terrain, and places with high radioactivity.

Mobile robots are mainly divided into locomotion by wheels, by tracks and locomotion by legs. Although wheeled locomotion has more efficiency in regular terrain and indoor environments, a legged locomotion is more robust when obstacles appear on its path [1,2]. Inside legged locomotion, there is the sub-group of quadruped robots, which are robots that use four legs to move. Compared to other types of legged robots, quadrupeds take advantage of bipedal robots in terms of stability, and compared to hexapod and octopod robots, they are more efficient [3,4].

There are also a lot of studies about quadruped robot models, specially in control and gait planning, as in [5] which describes a Central Pattern Generator to generate its gait or in [6] that implements a free gait based on posture control.

A. Ollero et al. (eds.), *ROBOT 2017: Third Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 694, https://doi.org/10.1007/978-3-319-70836-2_62

Still, [7] presents an on-line technique to construct walking gaits in irregular terrains using a control basis and [8] investigates an operational space control based on hierarchical task optimisation for quadruped locomotion.

The quadruped robot presented in this work adopts the trot gait, originally found in horses [12], for its locomotion. Although factors such energy efficiency, stability, speed and mobility are still far from ideal, quadruped robots provide a good approach of the biological model when appropriate locomotion patterns are implemented [9].

The problem in the control of mobile systems is the proper application of parameters according to the goals to be achieved. Defining the sampling rate, which controller to use, how much accuracy the locomotion pattern requires, the energy efficiency and the specifications of the actuators and sensors, are necessary efforts that not always are a trivial solution. In most cases the mathematical models of the systems are complex and non-linear, and to find a proper controller for such systems require the application of linearization techniques in order to reach a simple and similar plant for the robot.

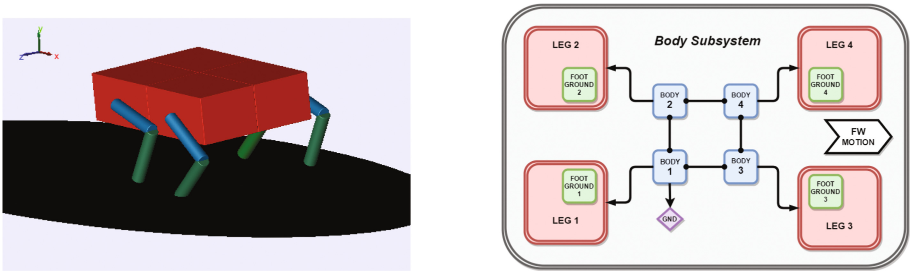
Having these ideas in mind, this paper presents an overview of the implemented robot model in Sect. 2, considerations about the planned trajectory are shown in the Sect. 3, the implementation and analysis of the control system using Matlab®/Simulink™ are presented in Sect. 4 and finally the conclusion and future developments are given in Sect. 5.

2 Model Overview

The model of the robot used in this work is made of a four-part body with two rotational joints in each leg, which provide angular movements along the Z axis. Between each two adjacent parts of the body there's a spring and damper system associated with a 2 DOF joint that gives more flexibility to the movements of the body [13]. In order to improve the simulated model, a foot-ground contact system, using equations of reaction forces in X and Y axis, was implemented. Each part of the robot is associated with an inertia matrix according to its shape. The robot has a total weight of 4 kg. Figure 1a shows the aspect of the robot model in SimMechanics™. The simplified diagram with connections between body, legs and foot-ground systems is presented in Fig. 1b. The diagram also has a GND block that is needed to provide the robot its position in relation to the ground [15].

2.1 Body Subsystems

The body has four interconnected blocks, each with its own rectangular shape inertia matrix I_b with mass $m_b = 0.9$ kg, which is described in Eq. (1) due to its symmetry around coordinated axis [13]. The robot body dimensions are width $(w_b) = 0.30$ m, depth $(d_b) = 0.20$ m and height $(h_b) = 0.15$ m. The complete body subsystem is shown in Fig. 2.



(a) Simulated quadruped robot aspect in SimMechanics™.

(b) Simplified diagram of the model.

Fig. 1. Quadruped robot model overview.

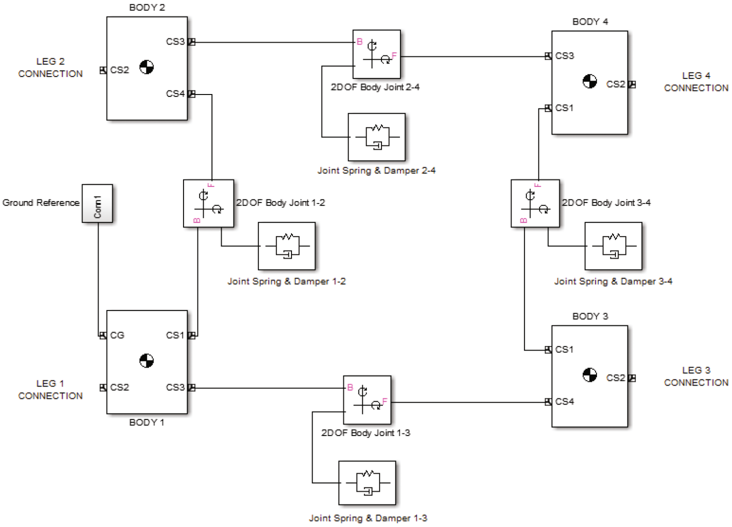


Fig. 2. Body scheme overview in SimMechanics.

$$I_b = \frac{1}{12}\begin{bmatrix} m_b(h_b^2 + d_b^2) & 0 & 0 \\ 0 & m_b(w_b^2 + d_b^2) & 0 \\ 0 & 0 & m_b(w_b^2 + h_b^2) \end{bmatrix} \quad (1)$$

The CS2 port present in each body block is used to connect the corresponding leg subsystem, and the ground reference is connected to the center of gravity of the BODY 1.

2.2 Leg Subsystems

As mentioned, each leg has two rotational joints: one in the hip and one in the knee. The implemented leg subsystem is depicted in Fig. 3.

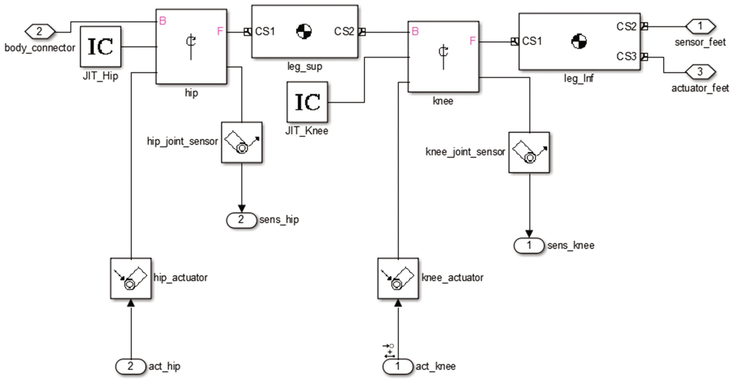


Fig. 3. SimMechanics diagram of the leg subsystems.

The two IC blocks provides the initial angle of each joint in the simulation. The subsystem is connected in its left to the corresponding body block and in its right to its foot-ground block. The control system sends the torque signal to the actuator blocks and its feedback is provided by the sensor blocks.

The cylindrical inertia matrix associated with the body blocks *leg_sup* and *leg_inf* has radius $r = 0.02$ m, length $L = 0.2$ m and mass $m_l = 0.05$ kg. Its equation, for the applied case of alignment in X axis, is provided by (2) [13].

$$I_l = \begin{bmatrix} \frac{1}{2}m_l r^2 & 0 & 0 \\ 0 & \frac{1}{12}m_l(3r^2 + L^2) & 0 \\ 0 & 0 & \frac{1}{12}m_l(3r^2 + L^2) \end{bmatrix} \tag{2}$$

2.3 Foot-Ground Subsystems

The foot-ground subsystems are included in the model to get a closer approach to a real world robot. Its architecture is shown in Fig. 4.

The sensor provides the velocity and position of the robot feet in order to compute friction and normal forces that the robot applies in the ground. The applied forces are calculated through Eqs. (3) and (4), which happens only when foot touch the ground.

$$F_x = -K_x(x - x_0) - B_x\dot{x} \tag{3}$$

$$F_y = -K_y y - B_y(-y)^m \dot{y} \tag{4}$$

The equation used to compute friction forces is based on a linear spring-damper system in which F_x is the applied force, K_x is the elastic constant of the spring, x and x_0 are the current and previous X axis position in relation to the ground, respectively, B_x is the constant of the damper and $\dot{x}$ is the velocity in X axis provided by the body sensor.

For the normal force calculation was chosen a non-linear spring-damper equation because it presented better results in tests [11]. In this case, F_y is the applied

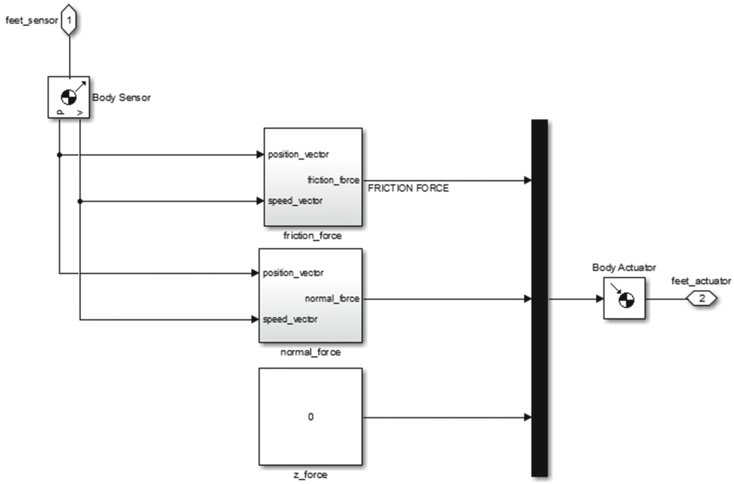


Fig. 4. SimMechanics diagram of the foot-ground subsystems.

normal force, K_y is the elastic constant of the spring, y is the current Y axis position, B_y is the damper constant, m is a factor that depends on the characteristics of the soil and $\dot{y}$ is the velocity in Y axis provided by the body sensor.

3 Trajectory Planning

Trajectory planning is crucial to the robot's locomotion. A good planning means higher efficiency because it maximises the generated movement and minimises the spent energy used to generate it. With this in mind, a good option to legged locomotion is to use a cycloid trajectory during the transfer phase of the leg. In this way, the implemented trajectory is given by the expressions:

$$P_T = \left[V_F \left[t - \frac{t_{T_0}}{2\pi} sin(\frac{2\pi t}{t_{T_0}}) \right], \frac{F_c}{2} \left[1 - cos(\frac{2\pi t}{t_{T_0}}) \right] \right]^T \tag{5}$$

$$P_S = [V_F T_0, 0]^T \tag{6}$$

where T_0 is the cycle time, V_F is the frontal velocity of the body defined by $V_F = L_S/T_0$, t_{T_0} is the transfer phase time given by $t_{T_0} = (1-\beta)T_0$ (where β is the duty factor of the leg), t is the time variable and F_c is the maximum height of the cycloid.

In order to make possible the implementation of the gait it is needed to analyse the kinematics of the robot. Kinematics are responsible for providing the reference angles to the controller based on the trajectory planned positions. The study of kinematics was performed for one leg only, since the others present the same shapes and sizes. The kinematic model of a leg with two degrees of freedom was used in the case.

4 Control System and Tuning

Due to the difficulty of finding a model that describes the robot's non-linear behaviour, the design of the control system appeals to linearization methods to find a state-space representation that describes the main dynamics of the robot. This was made in order to find a proper solution for the tuning of controller parameters. After the tuning, the controller will act in robot's non-linear system. The simplified block diagram of the system with the controller is shown in Fig. 5.

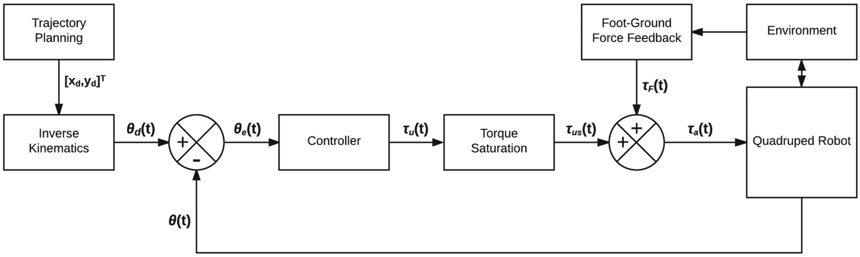


Fig. 5. Overall control system diagram.

The adopted procedure for the tuning of the controller was the following:

(a) Select the IO points of the robot for linearization;
(b) Select the parameters that will be trimmed;
(c) Obtain the linear state space of the plant using the trim model;
(d) Find a proper compensator for the linearized plant using classic control methods;
(e) Test the selected parameters in the non linear model of the robot;
(f) If necessary, make fine adjustments and repeat the last step.

The linearization of the system was made using the Simulink Linear Analysis tool. The linearized model was obtained using the trim model technique in the hip and knee of leg 1. All the joint's positions and velocities remained untrimmed, while all the other parameters were setted to steady state. Since all legs have the same length and shape, there was no need to reproduce the analysis for all of them. The step-by-step linearization process is not in the scope of this work, and both techniques can be found in Simulink Control Design reference [14].

After linearization, the space state model was used in the Simulink Control System Analysis in order to obtain a proper compensator. The parameters were tuned using the PID Tuning tool shown in Fig. 6. A good option to this case is to use the PD controller with first order filter due to the varying reference $\theta_d(t)$.

The transfer function of the PD controller with filter for the leg i and joint j is given by a lead type compensator formula, as described in Eq. (7).

$$C_{ij}(s) = K_{ij}\frac{s + Z_{ij}}{s + P_{ij}}\ ,\ Z_{ij} < P_{ij} \tag{7}$$

Due to its resemblance, the controller C_{i1} and the controller C_{i2} were tuned once and used in all hip and knee joints, respectively.

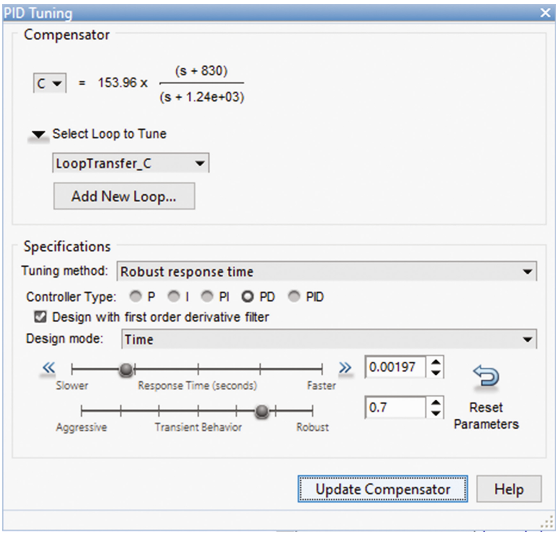


Fig. 6. Tuning interface for the compensator.

5 Tests and Results

The tests were carried while the robot was adopting the trot gait. The velocity V_F and the parameters K_{ij}, Z_{ij} and P_{ij} of the controllers were varied in tests for a saturation limit of ± 20 Nm in order to obtain a first approach. After that, the best tuning was applied to different values of saturation and the results were compared. The number of simulated cycles was $n_c = 5$ corresponding to a simulation time of $t = n_c * T_0$. The sampling period was 0.005 s and the step length was $L_S = 0.2$ m. A total of ten tests were performed varying the controller parameters, and each test considers five different periods. Finally, for the best case the system was tested for five different values of saturation.

5.1 Performance Indices

The used performance indices were the Integral of Absolute Error (I_{ae}), the Integral of Squared Error (I_{se}) and its percentage in relation to the reference θ_d area, $\%I_{ae}$ and $\%I_{se}$, given respectively by expressions (8)–(11).

$$I_{ae_{hip}} = \sum_{i=1}^{4} \int_0^{nT_0} |\theta_{d_{i1}} - \theta_{i1}| dt \quad , \quad I_{ae_{knee}} = \sum_{i=1}^{4} \int_0^{nT_0} |\theta_{d_{i2}} - \theta_{i2}| dt \qquad (8)$$

$$I_{se_{hip}} = \sum_{i=1}^{4} \int_0^{nT_0} |\theta_{d_{i1}} - \theta_{i1}|^2 dt \quad , \quad I_{se_{knee}} = \sum_{i=1}^{4} \int_0^{nT_0} |\theta_{d_{i2}} - \theta_{i2}|^2 dt \qquad (9)$$

$$\%I_{ae} = \sum_{i=1}^{4}\sum_{j=1}^{2} \frac{\int_0^{nT_0} |\theta_{d_{ij}} - \theta_{ij}|dt}{\int_0^{nT_0} |\theta_{d_{ij}}|dt} * 100 \tag{10}$$

$$\%I_{se} = \sum_{i=1}^{4}\sum_{j=1}^{2} \frac{\int_0^{nT_0} |\theta_{d_{ij}} - \theta_{ij}|^2 dt}{\int_0^{nT_0} |\theta_{d_{ij}}|dt} * 100 \tag{11}$$

where $\theta_{d_{ij}}(t)$ is the desired angle and $\theta_{ij}(t)$ is the measured angle for the leg i and joint j.

As a measure of efficiency of the system is used the average energy consumption by travelled distance, E_{av}, given by:

$$E_{av} = \frac{1}{d}\sum_{i=1}^{4}\sum_{j=1}^{2}\int_0^{T_0} |\tau_{a_{ij}}(t)\dot{\theta}_{ij}(t)|dt \tag{12}$$

where $\tau_{a_{ij}}$ is the applied torque and $\dot{\theta}_{ij}$ is the measured velocity in each leg i and joint j of the robot. Note that E_{av} assumes that energy regeneration is not available by actuators doing negative work [10].

For actuator size considerations, the maximum torque of the hips τ_{hip} and knees τ_{knee}, which corresponds to the stall torque needed to move the joint during a brief period of time, are also computed.

5.2 Results

After several tests with different control parameters, the results presented in Table 1 are given by test number 5, which best satisfied the desired performance in terms of error and efficiency. The best parameters are the ones that have the lowest values of $E_{av} * \%I_{ae}$ and $E_{av} * \%I_{se}$, as shown in Figs. 7 and 8, respectively.

Table 1. Best results of the tests.

V_F (ms^{-1})	Iae_{hip} (deg)	Iae_{knee} (deg)	Ise_{hip} (deg)	Ise_{knee} (deg)	$\%Iae$	$\%Ise$	d (m)	E_{av} (Jm^{-1})
0.1	6.176	2.931	2.170	0.401	0.161	0.045	0.745	22681.668
0.05	6.571	4.344	1.548	0.414	0.096	0.017	1.126	9756.350
0.033	8.110	5.993	1.347	0.441	0.083	0.011	1.030	7425.555
0.025	11.249	7.892	4.216	0.631	0.084	0.021	0.981	5942.543
0.02	13.717	9.802	5.039	0.810	0.083	0.021	1.000	4645.231

The controller parameters that correspond to the results of Table 1 are shown in Table 2.

Using the parameters of Table 2, the next test was to vary the saturation limit. For all five tests performed, the results are shown in Table 3. As expected,

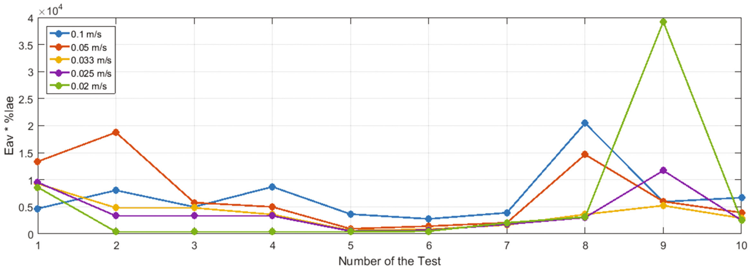


Fig. 7. Values of E_{av} multiplied by $\%I_{ae}$ for each test.

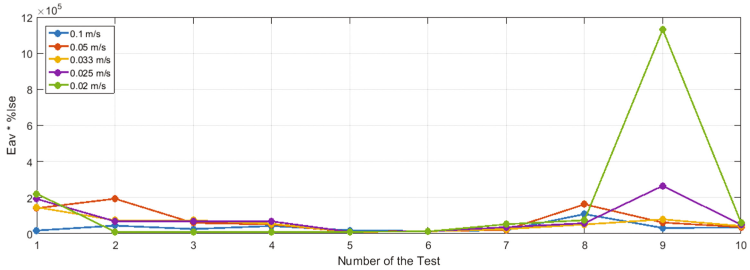


Fig. 8. Values of E_{av} multiplied by $\%I_{se}$ for each test.

Table 2. Controller parameters for the best case.

Parameter	Hip joint	Knee joint
K_{ij}	77	204
Z_{ij}	20.88	20.88
P_{ij}	150	150

Table 3. Saturation tests in the best case of the controller.

τ_{sat} (Nm)	Iae_{hip} (deg)	Iae_{knee} (deg)	Ise_{hip} (deg)	Ise_{knee} (deg)	$\%Iae$	$\%Ise$	d (m)	E_{av} (Jm^{-1})
20	6.571	4.344	1.548	0.414	0.096	0.017	1.126	9756.350
15	6.867	4.814	2.368	0.639	0.103	0.027	1.071	9869.309
10	8.810	5.401	5.488	1.130	0.125	0.058	0.827	9789.381
5	73.618	11.784	652.621	8.986	0.753	5.836	0.862	5179.345
4	103.717	14.724	1292.455	17.459	1.045	11.555	0.905Z	3974.033

Fig. 9 shows that the higher energy efficiency E_{av} is obtained for the lowest saturation. On the other hand, higher precision leads to higher torques, as shown in Fig. 10. The joint trajectories for a maximum torque of 10 Nm are shown in

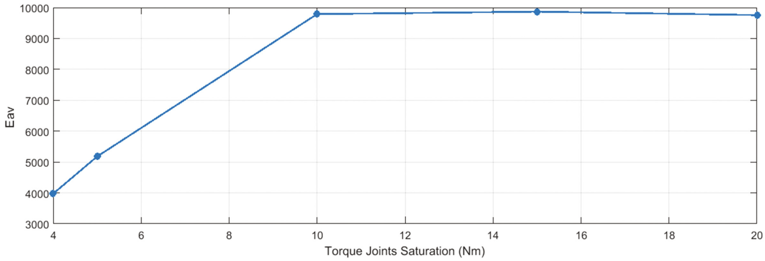


Fig. 9. Values of E_{av} versus saturation torque.

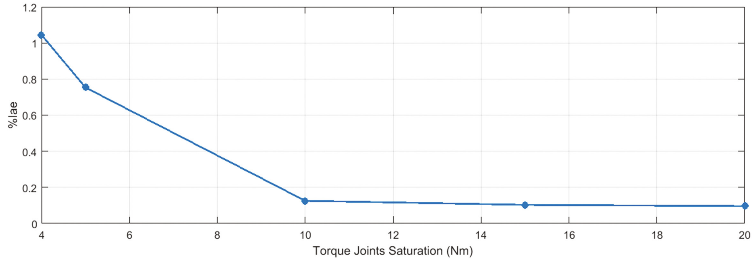


Fig. 10. Values of $\%I_{ae}$ versus saturation torque.

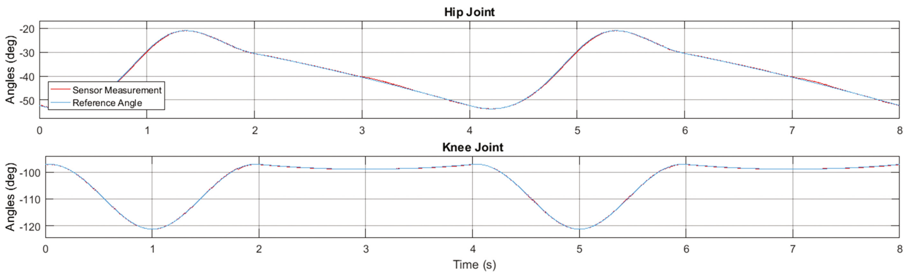


Fig. 11. Joint trajectories of leg 1 for a saturation of 10 Nm.

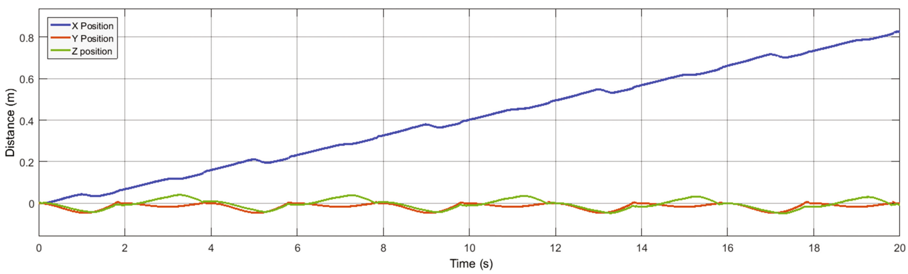


Fig. 12. Total distance travelled by the robot for a saturation of 10 Nm in the joints.

Fig. 11 while in the Fig. 12 the position of the robot's body is illustrated. As can be seen, the robot follows the desired joint trajectories with good accuracy, travelling a total distance of approximately 1 m.

6 Conclusions

The presented work focused in the implementation of a control system for a quadruped robot using linear analysis and trajectory planning. The planned trajectory presents good results when adopted the trot gait, especially for a velocity of $V_F = 0.05\,\mathrm{ms}^{-1}$. The linearization provides a good response to the behaviour of the controllers, but still require fine adjustments to get better results. It is also concluded that the error is directly associated with the joint's torque, which may vary with the presence of saturation in the controllers. Although several parameters can still be changed to improve the results, this work provided an analysis of some relevant aspects of modelling and control of a quadruped robot in simulation environment and represents the first steps for further developments.

Acknowledgments. This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

1. Hutter, M., et al.: ANYmal - a highly mobile and dynamic quadrupedal robot. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, pp. 38–44 (2016). https://doi.org/10.1109/IROS.2016.7758092
2. Raibert, M.H.: Legged Robots that Balance. MIT press, Cambridge (1986)
3. Cao, Q., van Rijn, A.T., Poulakakis, I.: On the control of gait transitions in quadrupedal running. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, pp. 5136–5141 (2015). https://doi.org/10.1109/IROS.2015.7354100
4. Gao, F., Qi, C., Sun, Q., Chen, X., Tian, X.: A quadruped robot with parallel mechanism legs. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, p. 2566 (2014). https://doi.org/10.1109/ICRA.2014.6907223
5. Rutishauser, S., Sprowitz, A., Righetti, L., Ijspeert, A.J.: Passive compliant quadruped robot using Central Pattern Generators for locomotion control. In: 2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics, Scottsdale, AZ, pp. 710–715 (2008). https://doi.org/10.1109/BIOROB.2008.4762878
6. Igarashi, H., Machida, T., Harashima, F., Kakikura, M.: Free gait for quadruped robots with posture control. In: 9th IEEE International Workshop on Advanced Motion Control (2006). https://doi.org/10.1109/AMC.2006.1631698
7. MacDonald, W.S., Grupen, R.A.: Building walking gaits for irregular terrain from basis controllers. In: 1997 IEEE International Conference on Robotics and Automation, Proceedings, vol. 1. IEEE (1997)

8. Hutter, M., et al.: Quadrupedal locomotion using hierarchical operational space control. Int. J. Robot. Res. **33**(8), 1047–1062 (2014)
9. Suzuki, H., Nishi, H., Aburadani, A., Inoue, S.: Animal gait generation for quadrupedal robot. In: Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007), Kumamoto, p. 20 (2007). https://doi.org/10.1109/ICICIC.2007.169
10. Silva, M.F., Machado, J.A.T.: Kinematic and dynamic performance analysis of artificial legged systems. Robotica **26**(1), 19–39 (2008)
11. Silva, M., Machado, J., Lopes, A.: Modelling and simulation of artificial locomotion systems. Robotica **23**(5), 595–606 (2005). https://doi.org/10.1017/S0263574704001195
12. Muybridge, E.: Animals in Motion. Courier Corporation (2012)
13. Silva, M., Barbosa, R., Castro, T.: Multi-legged walking robot modelling in MATLAB/Simmechanics and its simulation. In: 2013 8th EUROSIM Congress on Modelling and Simulation (EUROSIM). IEEE (2013)
14. Mathworks. Simulink Control Design Reference. Version R2016b. User guide (2016)
15. Mathworks. Simscape Multibody Reference. Version R2016b. User guide (2016)