Modelling and Control of Single Legged Hopping Robot

Erk Bamyaci[1], Musa N. Yazar[2], S. Murat Yesiloglu[3]

Control and Automation Engineering Department
Istanbul Technical University, Istanbul
[1] bamyaci@itu.edu.tr, [2] mnyazar@itu.edu.tr, [3] yesiloglu@itu.edu.tr

Abstract

In the scope of legged robotics, this study is about dynamical modelling and control of a single legged two degrees of freedom planar hopping robot while keeping the focus on the interaction forces between the robot and the ground based on a given ground contact model. Based on Raibert's three step algorithm, forward speed control is achieved by adjusting the foot position of the robot at touchdown and height control is achieved by adjusting the compression length of a linear spring. Using Newton-Euler based Spatial Operator Algebra (SOA) method, simulation results of the height control of hopping, forward speed control and the body attitude control of the robot are given in the paper.

1. Introduction

Until this day, human beings were always interested in developing machinery and analyzing how they work, to make things easier or just out of curiosity. In 1970s, Kato and Vukobratovic were the first ones who studied on legged robots which is one of the most important branches of those studies. Especially wheeled robots drew the most attention for many years, because it has been known how hard it is to control and calculate legged robots. It is harder to stabilize and control legged robots than the wheeled ones. However, as human beings wanted to achieve more tasks by using robots, the studies made on legged robots increased because of their versatility [3].

At present, legged robots can interact better with the environment that is designated for human beings - legged movement. For this reason legged robots can reach the places where wheeled robots cannot. And this enables legged robots to execute tasks which are dangerous for humans.

At the present time, wheeled robots are the fastest moving robots on ground. However, they may be insufficient when agility is needed. At this point legged robots come into play. Legged robots can change their body attitude faster. And using this ability, they are able to stabilize better on uncertain grounds.

While it is impossible for wheeled robots in conditions where vertical climbing is needed, specially designed legged robots are also able to perform these tasks [2].

Wheels can move only on continuous ground and very little on non-continuous or uneven ground. Therefore, legged robots submit a wider range of solution space. Thereby they own the capacity to perform various tasks.

The main difference which separates hopping robots from walking ones is the interaction between the foot and the ground. Another difference is while walking, in every step at least one foot serves as a support, thus there is no flight phase. On the contrary, hopping robots has a flight phase. While walking robots are statically stable in every step, since they have a flight phase, hopping robots are not statically stable.

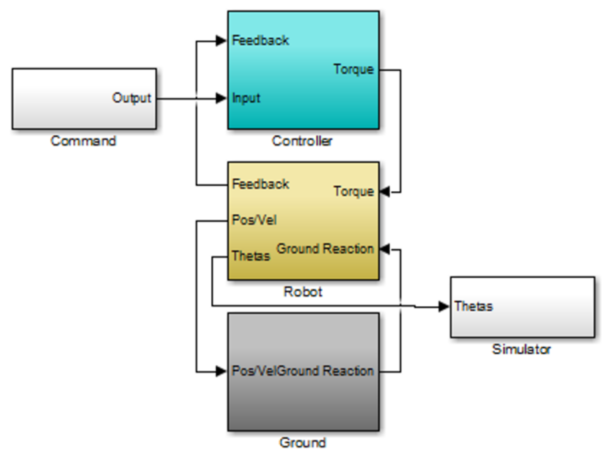

Figure 1. Functional decomposition of the hopping robot

The purpose of this study is to understand the dynamics which direct legged robots, especially the hopping robots. Also to control the basic movements such as hopping height, forward speed and body attitude of the robot. At the same time, to analyze its interaction with the ground and add it to the model.

To complete this study, first of all, the kinematic model of the robot which has a total of eight degrees of freedom that includes the passive six degrees of freedom of the ground was formed. After obtaining the dynamical model, the ground type which will present the interaction of the robot with the ground in the most realistic way was added to the system. After completing the modelling process of the robot, the control system which will enable to control the movements was also added. Improved version of Raibert's hopping algorithms was used to control the model.

Using a Newton-Euler based method instead of a Euler-Lagrange based method made it easier to add the tip force that effects the robot, to the system. The modular design of the functions, as shown in figure 1, and using the algorithms formed by the study group made it possible to adapt the simulator to different legged robots, with minimum effort.

During this study, all of the simulations were carried out in MATLAB and Simulink. The 3D virtual reality model of the robot was realized in Virtual Reality Modelling Language (VRML) format. The goal of the study is, to make a realistic hopping simulation of the one-legged robot which has been detailed above by using the outputs obtained and to have a tested model in hand, if the physical implementation is done in future studies.

In this study which was done to understand the dynamics that control legged robots, especially the hopping ones, some simplifying assumptions were adopted to have the results in a shorter time. These assumptions are:

- The analysis, the designs and the models were realised in 3D world, but the simulation that has been made moves in 2D.
- Simple controllers were used for control purposes. Controllers aim to control the movements of the robot even if there is a steady-state error.
- The interaction between the foot and the ground was modelled as point contact.
- The actuators that real robots have, has been ignored in this study.
- The friction and the energy loss produced in the system were omitted within this thesis. Only the energy losses which were produced by outer forces were considered.

2. Model of Hopping Robot

There are two methods for solid system modelling on the computer. In the first, equation of motion for a system is done manually and then it is turned into computer code. In the second method which was used in this study, a system model that defines the given pattern is formed and this model is given as an argument to the model-based calculation routine [1]. This method is more desirable because it is easier to construct and to modify. This method gives the opportunity to use it in different systems by making small modifications on the model which is given in this study.

Dynamical modelling technique used in this paper is based on Spatial Operator Algebra [4]. The vectors belonging to every link according to this method are given in figure 2.

Figure 3 presents the physical pattern of the one-legged hopping robot that is on a mobile base that has 8 degrees of freedom. The robot consists of three parts which are base, hip and leg. The hip is spherical, the base and the leg are cylindrical. The base is attached to the hip by a rotational joint and the hip is attached to the leg by a prismatic joint. This structure was inspired from Raibert's single-legged hopping robot whose spring effect is provided by a control mechanism [5]. The moment of inertia of the base is defined considerably higher than the moment of inertia of the leg. This moves the hip joint to position the leg in the flight phase and prevents a big variation at the attitude of the base.

2.1. Kinematic Model

The angular velocities of the manipulator are,

$$\begin{aligned} \vec{\omega}_1 &= \vec{\omega}_b + \vec{h}_1\dot{\theta}_1 \\ \vec{\omega}_2 &= \vec{\omega}_1 + \vec{h}_2\dot{\theta}_2 \end{aligned} \tag{1}$$

And the linear velocities are,

$$\begin{aligned} \vec{v}_1 &= \vec{v}_b + \vec{\omega}_b \times \vec{l}_{0,1} = \vec{v}_b - \vec{l}_{0,1} \times \vec{\omega}_b = \vec{v}_b - \mathbf{L}_{0,1}\vec{\omega}_b \\ \vec{v}_2 &= \vec{v}_1 + \vec{\omega}_1 \times \vec{l}_{1,2} = \vec{v}_1 - \vec{l}_{1,2} \times \vec{\omega}_1 = \vec{v}_1 - \mathbf{L}_{1,2}\vec{\omega}_1 \end{aligned} \tag{2}$$

Here, $\mathbf{L}_{k-1,k} = \vec{l}_{k-1,k}\times$ is an operator in the form of 'skew symmetric' matrix. If equations (1) and (2) are written in matrix

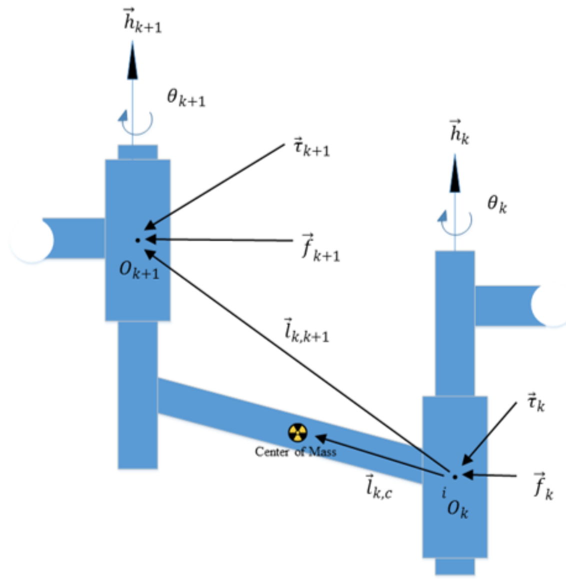

Figure 2. Vectors that belong to k'th link

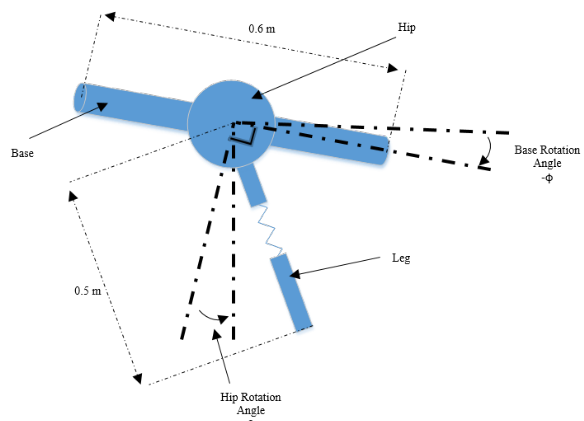

Figure 3. The physical pattern of the robot with 8 degrees of freedom

form, it is as below:

$$\vec{\vec{V}}_1 = \Phi_{1,0}\vec{\vec{V}}_b + \vec{\vec{H}}_1\dot{\theta}_1$$
$$\vec{\vec{V}}_2 = \Phi_{2,1}\vec{\vec{V}}_1 + \vec{\vec{H}}_2\dot{\theta}_2 \tag{3}$$

Here the spatial velocities of the links are:

$$\vec{\vec{V}}_1 = \begin{bmatrix} \vec{\omega}_1 \\ \vec{v}_1 \end{bmatrix} \qquad \vec{\vec{V}}_2 = \begin{bmatrix} \vec{\omega}_2 \\ \vec{v}_2 \end{bmatrix} \tag{4}$$

And distribution operators are:

$$\Phi_{1,0} = \begin{bmatrix} {}_3\mathbf{I} & {}_3\mathbf{0} \\ -\mathbf{L}_{0,1} & {}_3\mathbf{I} \end{bmatrix} \qquad \Phi_{2,1} = \begin{bmatrix} {}_3\mathbf{I} & {}_3\mathbf{0} \\ -\mathbf{L}_{1,2} & {}_3\mathbf{I} \end{bmatrix} \tag{5}$$

Finally, the spatial vectors of the axis of rotation of number one rotational joint and number two prismatic joint are as shown:

$$\vec{\vec{H}}_1 = \begin{bmatrix} \vec{h}_1 \\ \vec{0} \end{bmatrix} \quad \vec{\vec{H}}_2 = \begin{bmatrix} \vec{0} \\ \vec{h}_2 \end{bmatrix} \tag{6}$$

If spatial velocity of every vector is written from the base to the tip and simplifications are made it can be written as below:

$$\vec{\vec{V}}_1 = \Phi_{1,0}\vec{\vec{V}}_b + \vec{\vec{H}}_1\dot{\theta}_1$$
$$\vec{\vec{V}}_2 = \Phi_{2,0}\vec{\vec{V}}_b + \Phi_{2,1}\vec{\vec{H}}_1\dot{\theta}_1 + \vec{\vec{H}}_2\dot{\theta}_2. \tag{7}$$

The tip velocity of the manipulator is as follows:

$$\vec{\vec{V}}_t = \Phi_t \underline{V} \tag{8}$$

Here,

$$\underline{V} = \begin{bmatrix} \vec{\vec{V}}_1 \\ \vec{\vec{V}}_2 \end{bmatrix} \quad \Phi_t = \begin{bmatrix} {}_6\mathbf{0}, & {}_6\mathbf{0} & \Phi_{t,2} \end{bmatrix}$$

When necessary arrangements are made the kinematic equation below is derived:

$$\vec{\vec{V}}_t = \mathcal{J}\dot{\theta} + \Phi_{t,b}\vec{\vec{V}}_b \tag{9}$$

Here,

$$\mathcal{J} = \Phi_t\Phi\mathbf{H} \qquad \Phi_{t,b} = \Phi_{3,0} = \Phi_t\Phi\Phi_b$$

2.2. Dynamic Model

The derivatives of the angular and linear velocities should be taken to advance to dynamic analysis:

$$\begin{aligned}\dot{\omega}_1 &= \dot{\omega}_b + \vec{h}_1\ddot{\theta}_1 + \omega_1 \times \vec{h}_1\dot{\theta}_1 \\ &= \dot{\omega}_b + \vec{h}_1\ddot{\theta}_1 + \omega_1 \times (\omega_1 - \omega_b) \\ &= \dot{\omega}_b + \vec{h}_1\ddot{\theta}_1 + \omega_b \times \omega_1 \\ \dot{\omega}_2 &= \dot{\omega}_1 + \vec{h}_2\ddot{\theta}_2 + \omega_2 \times \vec{h}_2\dot{\theta}_2 \\ &= \dot{\omega}_1 + \vec{h}_2\ddot{\theta}_2 + \omega_2 \times (\omega_2 - \omega_1) \\ &= \dot{\omega}_1 + \vec{h}_2\ddot{\theta}_2 + \omega_1 \times \omega_2\end{aligned} \tag{10}$$

and

$$\begin{aligned}\vec{v}_1 &= \vec{v}_b + \dot{\omega}_b \times \vec{l}_{0,1} + \omega_b \times \left(\omega_b \times \vec{l}_{0,1}\right) \\ &= \vec{v}_b - \vec{l}_{0,1} \times \dot{\omega}_b + \omega_b \times \left(\omega_b \times \vec{l}_{0,1}\right) \\ \vec{v}_2 &= \vec{v}_1 + \dot{\omega}_1 \times \vec{l}_{1,2} + \omega_1 \times \left(\omega_1 \times \vec{l}_{1,2}\right) \\ &= \vec{v}_1 - \vec{l}_{1,2} \times \dot{\omega}_1 + \omega_1 \times \left(\omega_1 \times \vec{l}_{1,2}\right)\end{aligned} \tag{11}$$

If the equations in (10) and (11) are written in matrix form:

$$\dot{\vec{\vec{V}}}_1 = \Phi_{1,0}\dot{\vec{\vec{V}}}_b + \vec{\vec{H}}_1\ddot{\theta}_1 + \vec{\vec{a}}_1$$
$$\dot{\vec{\vec{V}}}_2 = \Phi_{2,1}\dot{\vec{\vec{V}}}_1 + \vec{\vec{H}}_2\ddot{\theta}_2 + \vec{\vec{a}}_2 \tag{12}$$

Here $\vec{\vec{a}}_k$ are the spatial bias velocities:

$$\vec{\vec{a}}_k = \begin{bmatrix} \vec{\omega}_{k-1} \times \vec{\omega}_k \\ \vec{\omega}_{k-1} \times \left(\vec{\omega}_{k-1} \times \vec{l}_{k-1,k}\right) \end{bmatrix}$$

If all links are gathered in a single matrix using equation (12),

$$\underline{\dot{V}} = \Phi\left(\mathbf{H}\underline{\ddot{\theta}} + \underline{a} + \Phi_b\dot{\vec{\vec{V}}}_b\right) \tag{13}$$

is obtained. After this, torque and force distributions of the links can be written. This is done from the tip to the base because it cannot be done from the base to the tip.

$$\vec{\mathcal{T}}_2 = \vec{\mathcal{T}}_t + \vec{l}_{2,3} \times \vec{f}_t + \vec{l}_{2,c} \times \dot{\vec{v}}_2 m_2 + \frac{d}{dt}\left(I_2\vec{\omega}_2\right)$$
$$\vec{\mathcal{T}}_1 = \vec{\mathcal{T}}_2 + \vec{l}_{1,2} \times \vec{f}_2 + \vec{l}_{1,c} \times \dot{\vec{v}}_1 m_1 + \frac{d}{dt}\left(I_1\vec{\omega}_1\right) \tag{14}$$

In equation (14), the first and the second terms at the right side are coming from the upper link, the third term is from translation and the fourth one is from rotation. If link forces are written like torque distribution, it is as follows:

$$\vec{\vec{F}}_2 = \Phi_{3,2}^T\vec{\vec{F}}_t + \mathbf{M}_2\dot{\vec{\vec{V}}}_2 + \vec{\vec{b}}_2$$
$$\vec{\vec{F}}_1 = \Phi_{2,1}^T\vec{\vec{F}}_2 + \mathbf{M}_1\dot{\vec{\vec{V}}}_1 + \vec{\vec{b}}_1 \tag{15}$$

Here $\vec{\vec{F}}_k$ is the term for link spatial forces, $\mathbf{M}_k$ is the term for link mass matrix and $\vec{\vec{b}}_k$ is the term for link spatial forces and are defined as follows:

$$\vec{\vec{F}}_k = \begin{bmatrix} \vec{\mathcal{T}}_k \\ \vec{f}_k \end{bmatrix} \qquad \mathbf{M}_k = \begin{bmatrix} I_k & m_k\mathbf{L}_{k,c} \\ -m_k\mathbf{L}_{k,c} & {}_3\mathbf{I}m_k \end{bmatrix}$$

$$\vec{\vec{b}}_k = \begin{bmatrix} \vec{\omega} \times I_k\vec{\omega}_k \\ m_k\vec{\omega}_k \times \left(\vec{\omega}_k \times \vec{l}_{k,c}\right) \end{bmatrix}$$

If all link spatial vectors are gathered together, it can be written as:

$$\underline{F} = \Phi^T\left(\mathbf{M}\underline{\dot{V}} + \underline{b} + \Phi_t^T\vec{\vec{F}}_t\right) \tag{16}$$

Here,

$$\underline{F} = \begin{bmatrix} c\vec{\vec{F}}_1 \\ \vec{\vec{F}}_2 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 \end{bmatrix} \qquad \underline{b} = \begin{bmatrix} \vec{\vec{b}}_1 \\ \vec{\vec{b}}_2 \end{bmatrix}$$

Then, if equation (13) is located in its place at equation (17), it will be:

$$\underline{F} = \Phi^T\left(\mathbf{M}\Phi\mathbf{H}\underline{\ddot{\theta}} + \mathbf{M}\Phi\underline{a} + \mathbf{M}\Phi\Phi_b\dot{\vec{\vec{V}}}_b + \underline{b} + \Phi_t^T\vec{\vec{F}}_t\right) \tag{17}$$

The torques applied here are the projection of link spatial forces according to rotation axis. This can be written in mathematical equation as below:

$$\underline{\mathcal{T}} = \mathbf{H}^T\underline{F} \tag{18}$$

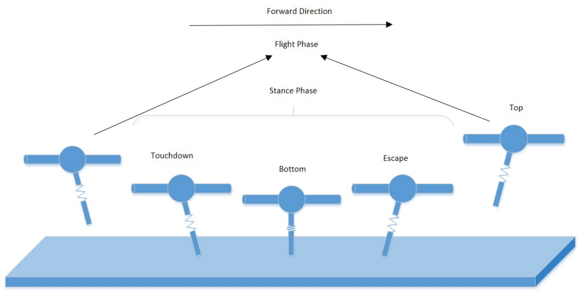

Figure 4. Schematic of Single Legged Hopping Robot's Functioning

For this reason, if equation (17) is multiplied with $\mathbf{H}^T$, the equation gives us the torques applied. By this way we can get the opposite dynamic equation of the manipulator,

$$\underline{\mathcal{T}} = \mathcal{M}\ddot{\underline{\theta}} + \underline{\mathcal{C}} + \mathcal{M}_b \dot{\vec{\vec{V}}}_b + \mathcal{J}^T \vec{\vec{F}}_t \tag{19}$$

Here,

$$\mathcal{M} = \mathbf{H}^T \Phi^T \mathbf{M}\Phi\mathbf{H}$$
$$\underline{\mathcal{C}} = \mathbf{H}^T \Phi^T (\mathbf{M}\Phi\underline{a} + \underline{b})$$
$$\mathcal{M}_b = \mathbf{H}^T \Phi^T \mathbf{M}\Phi\Phi_b$$

Above, $\mathcal{M}$ is generalized mass matrix, $\underline{\mathcal{C}}$ are bias terms that includes coriolis and gravity and $\mathcal{M}_b$ is the mass matrix about the interaction between the base and the manipulator.

3. The process of Single Legged Hopping Robot

The hopping simulation is started by giving an inertial height and an inertial horizontal velocity to the robot. With the effect of the gravity the robot accelerates towards the ground. As soon as the foot touches the ground, the spring at the leg joint is pressed until the vertical speed of the robot goes down to zero. After this, the spring starts to discharge and the robot starts to accelerate on the opposite direction. The robot continues to move up until the vertical velocity goes down to zero. Then the cycle is completed. In every cycle, a part of the energy disappears to accelerate the unsprung part of the leg. Thus, to reduce the energy loss, the mass of the leg's unsprung part should be less than the total mass. To maintain the hopping height, when leg is in contact with the ground, the leg motor starts to press the leg spring to provide extra energy to the system. To maintain the forward movement of the robot, the hip motor applies a torque in the flight phase so that it can adjust the position of the leg. The functioning schematic of the hopping robot is shown in figure 4.

4. Control System

The design of the control system that has been used in this study, is inspired from Raibert's single legged hopping robot controller [5, 6]. Raibert divided controlling into three steps:

- Hopping height control
- Forward speed control
- Body attitude control

4.1. Hopping Height Control

The hopping height of the single legged robot is obtained by a thrust applied to the leg joint in every cycle. The compression length of the spring is calculated by comparing the maximum height of the previous cycle with the height that is planned for the next cycle. Spring compression length is calculated proportional to this difference [7]. This proportional control rule can be written as:

$$\Delta l = \Delta l_{old} + k_p (h_d - h_{max}), \tag{20}$$

Here, Δl is the amount of spring compression, k_p is the proportional gain, h_{max} is the maximum height of the robot in the previous cycle, Δl_{old} the amount of spring compression in the previous cycle and h_d is the hopping height that is wanted to achieve.

4.2. Forward Speed Control

The forward speed is obtained by adjusting the position of the foot in every cycle before it touches the ground. Raibert observed that there is one foot position for every forward speed. To reach a reference speed, first of all, the control system will estimate the position of an indifferent spot which will not produce a speed change on the forward speed according to the present time. Then it places the foot before or after that indifferent spot to regulate the deceleration or acceleration. One of the important parameters to calculate the indifferent spot is the time the robot spends on the ground. And this time does not change as long as the mass and the spring rating stays the same. According to this, foot angle can be calculated with the equation below [7].

$$\theta_d = -\phi + sin^{-1}\left(\frac{\dot{x}T_s}{2r} + \frac{k_{\dot{x}}(\dot{x} - \dot{x}_d)}{r}\right), \tag{21}$$

Here, $k_{\dot{x}}$ is the velocity gain, $\dot{x}$ is the horizontal speed of the robot, r is the length of the leg, T_s is the time spent on ground and ϕ is the body attitude according to reference coordinate system. After calculating θ_d, we need to calculate the torque that has to be applied to the hip joint by using a PD controller, to achieve the angle we want. The control rules are as follows:

$$\tau = -k_p(\theta - \theta_d) - k_v\dot{\theta}, \tag{22}$$

Here, τ is the torque calculated, k_p is the proportional gain, k_v is the differential gain and θ is the hip angle.

4.3. Body Attitude Control

The purpose of the third step of the control system is to control the body attitude by applying a torque on the hip joint while the robot is on the ground. While doing this timing should be watched carefully. Because the robot will start to slip if the force used on the robot and the friction that is formed according to this is smaller than the torque used on the hip joint. When these are considered the control rule [7]:

$$\tau = +k_p(\phi - \phi_d) + k_v\dot{\phi}, \tag{23}$$

here, k_p is the proportional gain, k_v is the differential gain, ϕ is the robot base angle and ϕ_d the robot base angle which we want to achieve.

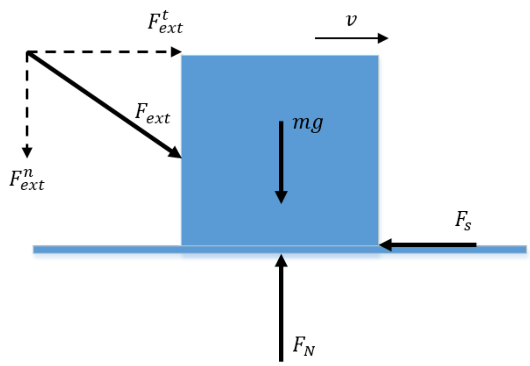

Figure 5. The force diagram of a block on the ground

5. Ground Contact Model

In general,when contact has occurred, the interaction between the robot and the ground can be studied in two topics: tangential and the normal forces which act on the robot. Figure 5 displays the force diagram originated from a block contacting with the ground.

5.1. Collision Model

Generally, contact models are divided in two topics: solid contact and compliant contact. Solid contacts is also called collision. The collision between two solid substances happens in a very short time, very high forces are included, sudden energy losses occur and high accelerations take place. In this study, the collision model is based on the advanced state of spring-damper collision model which was used in MSC Adams software. According to that, the collision model applied is as below [8]:

$$F = k(x_1 - x)^e - c_{max}\dot{x} \cdot STEP(x, x_1 - d, 1, x_1, 0), \tag{24}$$

Here, x is the distance between two bodies, x_1 is the distance where the friction is going to be started to calculate, k is the spring coefficient, e is the positive real variable which indicates the force deformation characteristic, c_{max} is the maximum damper coefficient, d is the penetration depth, and $STEP$ is the cubic step function.

One of the differences of the model from the classic spring - damper model is using the spring force as an exponential function. If x value is smaller than x_1 value, the relative velocity of two bodies will be different from zero and the linear damper force becomes discontinuous. For this reason, unlike the classic linear spring-damper model, in this method, cubic step function was used to increase the damper force along the penetration depth. Cubic step function is given below:

$$STEP = \begin{cases} h_0, & x \leq x_0 \\ h_0 + a\Delta^2(3 - 2\Delta), & x_0 < x < x_1 \\ h_1, & x \geq x_1 \end{cases}$$

$$a = h_1 - h_0$$
$$\Delta = (x - x_0)/(x_1 - x_0)$$

Here, x is the independent variable, x_0 is the starting point of the step function, x_1 is the finishing point of the step function, h_0 is the starting value of the step and h_1 is the last value of the step.

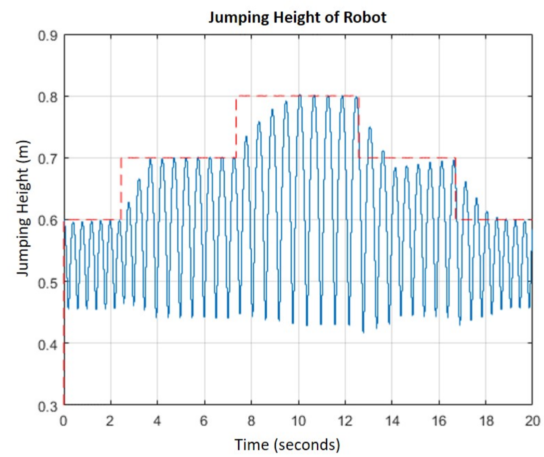

Figure 6. The graphic to track the robot's hopping height according to the given reference

5.2. Friction Model

Friction is the tangential surface reaction formed between two bodies in contact [9]. In this study, since it is easy to execute, classic Coulomb model was used. The main idea of Coulomb friction is that the friction responds to movement and its intensity is independent from the speed and the contact area. For this reason friction can be written as:

$$F_s = \mu F_n \tag{25}$$

Here F_s is the friction force, F_n is the normal force that effects the body and μ is the friction coefficient.

6. Simulation Results and Evaluation

The dynamic model of the single-legged hopping robot with 8 degrees of freedom is realised, the control system to control its hopping height and forward speed is designed and the simulation of the system is achieved. For dynamic evaluations Spatial Operator Algebra, that was suggested by Abhinandan Jain and Guillermo Rodriguez is used. During the controls of the single-legged hopping robot, the hopping height control, the forward speed control and the body attitude control was completed successfully by using Raibert's three-step control system. For the ground model, advanced spring-damper collision model and Coulomb friction force was applied. As the result of the tests, the coefficients for the control system and the ground model was determined and was added to the system to get a simulation as realistic as possible.

For position control of the leg joint, while robot is in contact with the ground a desired value was given to the system as input. At other phases, since the oscillation of the spring is not desired, zero is given as input. Different control entries for different phases of the hip joint has been applied. Hip joint is responsible for the angle of the foot when contacting the ground and arranging the basic attitude. The graphic outputs after applying the control is shown on figures 6 and 7.

In the hopping height graphics, steady state errors can be observed. That is the hopping height can be a little less or more than the desired height.

In the graphics obtained from the forward speed control, two things can be seen clearly. The first one is, right after the

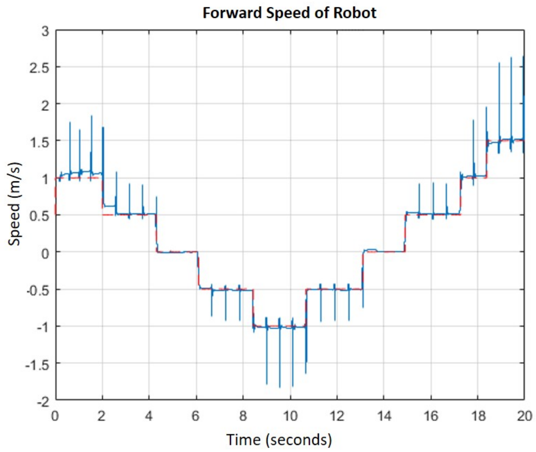

Figure 7. The graphic to track the robot's forward speed according to the given reference

start of the flight phase, there may be sudden increases at the forward speed. The main reason for that is the sudden change in the signal of the control system of the hip that produces a kick of the foot in the flight phase. And the second is, permanent attitude errors can also be seen at the forward speed.

7. Result

Because of their versatility legged robots is an important branch of robotic. Most of the studies that are made in this area are about the designing of the robots. Simulation is one of the important tools to get a successful design. It is important to use models that reflect the fact correctly to obtain designs which will be implemented to life successfully.

Gathering and bunching a large system with the help of smaller subsystems makes it quite easier to analyze, test and understand the unexpected behaviours of the system. As a result of this, it becomes easier to make modifications on the system and to observe their results.

One of the best ways to see the errors in a complicated system is, to visualise the movements of the system and view their behaviour in an virtual reality platform. The visualisations made on VRML, which is an virtual reality platform, was very helpful in understanding and solving the problems during this study.

Listed below are, the forward studies that can be added to develop and advance the studies which has been done in this area:

- To develop the system for two or multi-legged system.
- To advance the hopping simulation from 2D to 3D.
- To develop the system by using more realistic ground models.
- To add advanced control systems to the model.

8. References

[1] Yazar, Musa Nurullah, and S. Murat Yesiloglu. ”Path defined directed graph vector (Pgraph) method for multibody dynamics.” Multibody System Dynamics (2017): 1-19.

[2] Yan, Wang, et al. ”Development and application of wall-climbing robots.” Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on. Vol. 2. IEEE, 1999.

[3] Siciliano, Bruno, and Oussama Khatib, eds. Springer handbook of robotics. Springer, 2016.

[4] Rodriguez, Guillermo, Abhinandan Jain, and K. Kreutz-Delgado. ”Spatial operator algebra for multibody system dynamics.” Journal of the Astronautical Sciences 40.1 (1992): 27-50.

[5] Raibert, Marc H. Legged robots that balance. MIT press, 1986.

[6] Raibert, Marc H., H. Benjamin Brown Jr, and Michael Chepponis. ”Experiments in balance with a 3D one-legged hopping machine.” The International Journal of Robotics Research 3.2 (1984): 75-92.

[7] Bibalan, Pouyan Taghipour. ”Running Robots: Modeling, Control and Energy Efficiency.” (2009).

[8] ”About Adams/Solver” simcompanion.mscsoftware.com. MSC Software, 20 April. 2011. Web. 5 May 2017.

[9] Olsson, Henrik, et al. ”Friction models and friction compensation.” European journal of control 4.3 (1998): 176-195.