Forward and Inverse Dynamics and Quasi-Static Analysis of Mechanizes with MATLAB®

E. Corral, J. Meneses and J.C. García-Prada

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/63372

Abstract

There are many potential advantages of direct and inverse dynamic and quasi-static analysis of mechanisms, namely control the risk of slippage, improve stability, better adaptation to the environment, obtaining smooth movements and optimizing energy consumption. This chapter proposes new analysis methods and algorithms to bring new solutions to the mechanics of the machines under consideration. The methodology has been developed in modular programs thanks to the flexibility of MATLAB®.

In this chapter, a methodology for the complete kinematic and dynamic study of mechanisms is provided.

The programs have been designed so that all parameters can be modified. It was possible to automate these calculations creating an algorithm implemented in a programming language to easily find the solutions and the results of the analysis.

To test the interest of the methodology, in this chapter, this has been applied to the field of robots, especially the design of the biped robot PASIBOT. The inverse and forward dynamics, accounting for support foot slippage, are encoded in MATLAB®.

In addition, the methodology was applied to another machine, an unmanned ground vehicle (UGV), obtaining navigation optimization results using a numerical program based on a quasi-static half vehicle model.

Keywords: mechanism, dynamics, quasi-static, robot, vehicle

1. Introduction

Nowadays, walking robots, service robotics, and unmanned ground vehicles are considered one of the main areas of research. The employment of robots for dangerous tasks makes its design a crucial point. The interest in the development of humanoid robotics (personal assistance, social task, etc.) is rising, and it is being studied by a great number of research groups. The main issue is current mobile robots are not adapted to be used in domestic environments due to their lack of maneuverability but also to their large volume and/or weight.

These days humanoid biped have a high number of actuators that are used to control the high degrees of freedom (DOF) they possess. Nonetheless, one of the biggest drawbacks in humanoids is that both the weight and the power consumption still make this technology not suitable for many tasks and/or environments. In the majority of cases, around 30% of the total weight is related to the actuators and wires, and more than 25% to the reduction systems. That is why our work is focused on finding a new dynamics analysis to simplify the design of new mechanisms and kinematic chains which, maintaining the robot functionality, does not require such a high number of actuators. This would reduce the robot mass and hence its power consumption and total cost [1–4].

In recent years, different research groups have developed robots based on passive walking techniques. Biped locomotion has been studied from several perspectives. Much effort is devoted to the design of optimal trajectories and stabilized walking cycles via control programs. However, researchers have not focused enough their efforts in solving the slip problem, which has been largely ignored [5, 6].

In this chapter, the kinematics and dynamics analysis of PASIBOT (a biped walking robot designed and built by the Maqlab Groups of the Universidad Carlos III de Madrid, shown in **Figure 2**) are presented. The methodology to do the completed study from a theoretical point of view is explained. The study objective is to calculate all the forces and torques between links, as well as the linear and angular position coordinates, velocities, and accelerations for all links, for any time. The equations have been implemented in MATLAB® code, and the corresponding results have been contrasted [7–12].

The program accepts the initial kinematic state and the motor torque (as a function of time) as inputs and returns the bipedal movement, including the sliding of the supporting foot. The sliding is taken into account by adding one degree of freedom. Thus, we focus on the kinematics and dynamics of the sliding supporting foot [13].

In addition, there is no doubt about all advantages unmanned vehicles have. For this reason, the kinematics and dynamics analysis is one of the principal research lines in robotics. The Tallinn University of Technology is researching in the design and development of the unmanned ground vehicle (UGV) shown in **Figure 20**. This UGV is an all-terrain vehicle equipped with an engine for each of its wheels. The novel aspect of this vehicle is that each wheel is attached to the body by a leg so that the angle between the latter and the body may vary thanks to the use of an attached actuator [14–16].

In this chapter, a parametric quasi-static half vehicle model is implemented on MATLAB® following the explained methodology. The program calculates the variation in configuration angles that optimized desired criteria as the vehicle passes on a particular track profile. This algorithm makes it possible to find the variation of configuration angles along the track profile that keeps the applied torque constant and/or minimized and/or satisfying certain criterion.

2. Methodology

The sketch of the methodology is shown in **Figure 1**.

The first step is to define the mechanism. It is basically naming variables and parameters (with the correct nomenclature the implementation in MATLAB® code is made easier) and classifying the type of movement (degrees of freedom, class of joints, etc.)

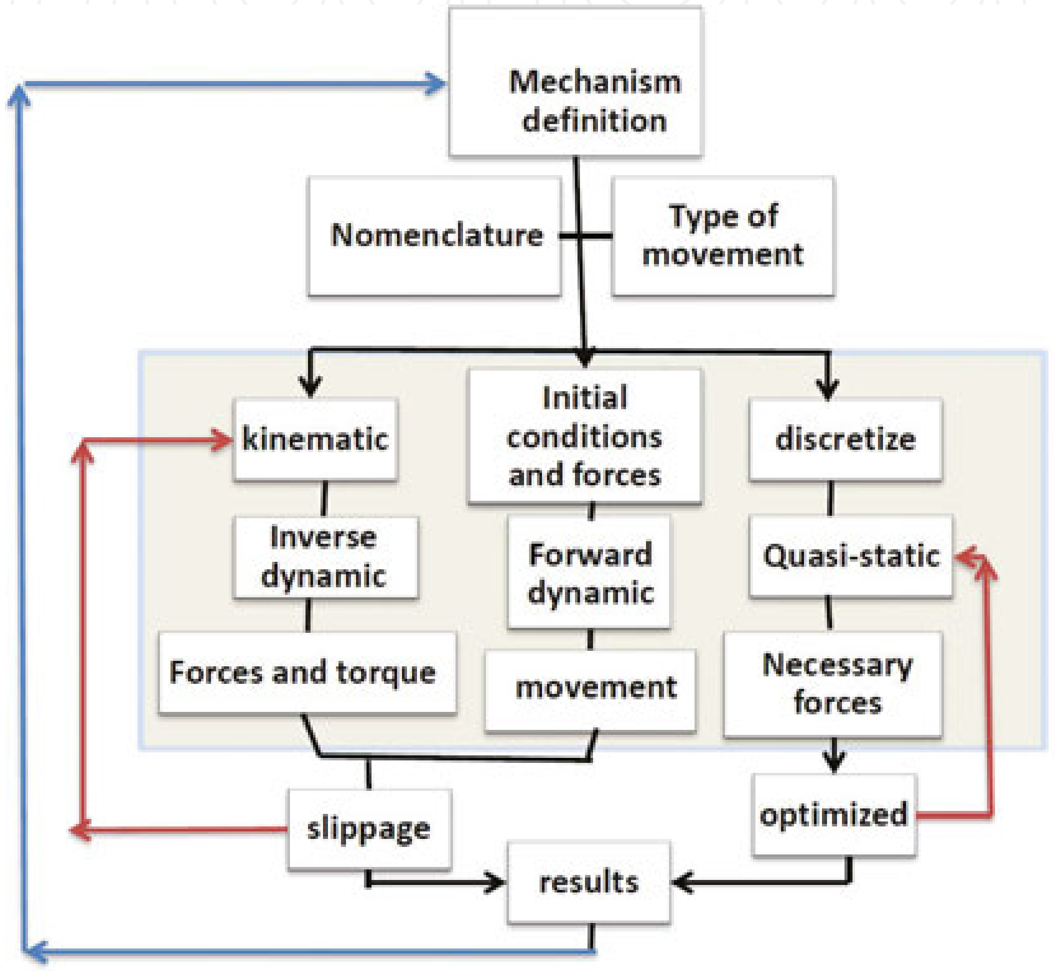


Figure 1. Sketch of the methodology.

It is possible to solve the kinematics and dynamics through the inverse dynamics or the forward dynamics algorithm depending on the type of mechanisms, the known input, and the desired output. Also, if the dynamics are complicated, the quasi-static approach can be useful. The biggest advantage of this approach is that it is easily optimized. This is a valuable tool to improve design and control with mathematics software.

In this chapter, the methodology of inverse and forward dynamics is applied to the biped walking robot PASIBOT and the quasi-static approach to an unmanned ground vehicle.

Note that the input and output of each approach are different and the equations are also different. However, the methodology can be simplified in the **Figure 1** for both, and the correctly chosen nomenclature helps to generate a MATLAB® code. The developed MATLAB® code is totally parametric, giving the possibility of changing the values, adding new analysis and new degrees of freedom (like slippage), or looking for analysis of sensibility.

Another advantage of doing the whole analysis with MATLAB® code is that the result can be used as an input for redesigning or as an optimizing function.

3. Kinematics of the biped robot PASIBOT

Nowadays, certain commercial software of mechanical simulation provides the dynamics of a mechanism with a small error. But, in some cases, like a biped robot actuated by small number of actuators, it is also possible to obtain the kinematics and apply this methodology and obtain the dynamics with a low degree of error. In this chapter, the kinematics of the biped PASIBOT is developed. In next chapters, the inverse and forward dynamics are addressed using the relations obtained in the kinematics.

The biped PASIBOT is a one-degree-of-freedom mechanical system based on a combination of classical mechanisms that emulates human walking. PASIBOT is shown in **Figure 2**.

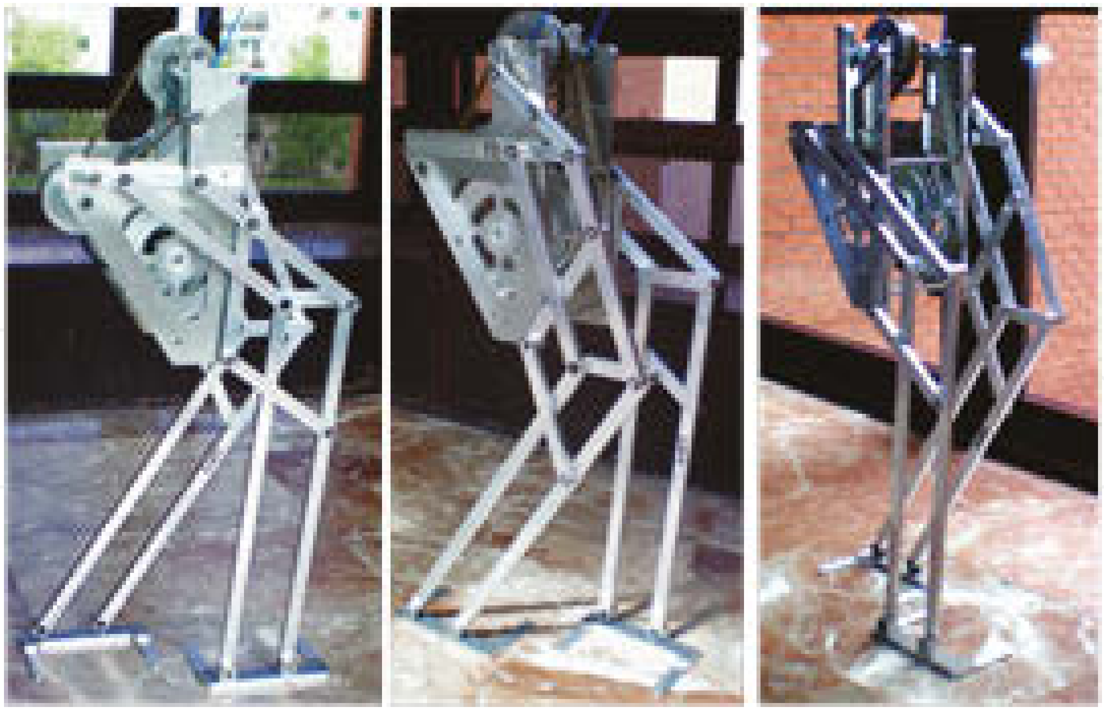


Figure 2. PASIBOT.

The biped PASIBOT (see **Figures 2–4**) is a mechanism composed of three sub-mechanisms of conventional type, each of which is designed to perform a different function:

1. The mechanism "Chebyshev" is responsible for generating a quasi-straight line.
2. A pantograph handles extend the "Chebyshev" coupling curve.
3. The stability of the biped is achieved by parallel extensions.

In **Figure 3**, the sub-mechanisms Chebyshev and pantograph as well as the trajectories for the most relevant points are shown.

The only engine of the biped conveys a full rotation to the crank of the Chebyshev mechanism, so that the end of its connecting rod (point C in **Figure 3**) makes a cyclical movement, one of its tracts being a quasi-straight line. This point is connected to one end of the pantograph, so that its other end (point E in **Figure 3**) carries an inverted and amplified from the previous path. The corresponding amplification ratio depends on the lengths of the links of the pantograph. The amplification ratio is two for the design of PASIBOT presented here.

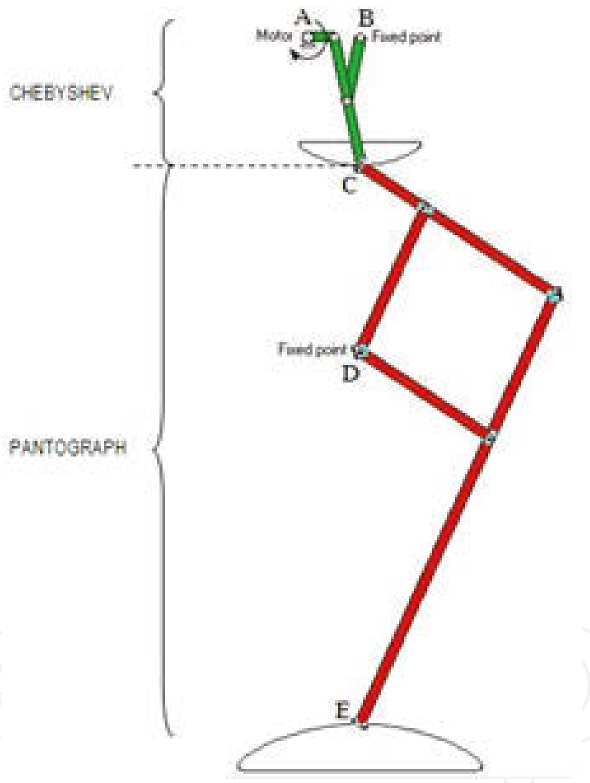


Figure 3. The sub-mechanisms Chebyshev and pantograph with the trajectories of their notable points.

Points A, B, and D in **Figure 3** are attached to the hip, as can be seen in **Figure 4**. The stabilization system consists of a series of articulated parallelograms which are based on the two longest links of the pantograph. These parallelograms guarantee the parallelism between the foot in contact with the floor and the stabilizing link, the end of which slides on a slot at the hip. Since that slot is aligned with the linear segment of the Chebyshev trajectory, the supporting foot remains parallel to the slot during the whole period of support. In order to provide the opposite leg with the appropriate movement, the corresponding crank is phased out π rad (see **Figure 4**) in the same motor axis. In fact, both cranks are part of the same rigid element.

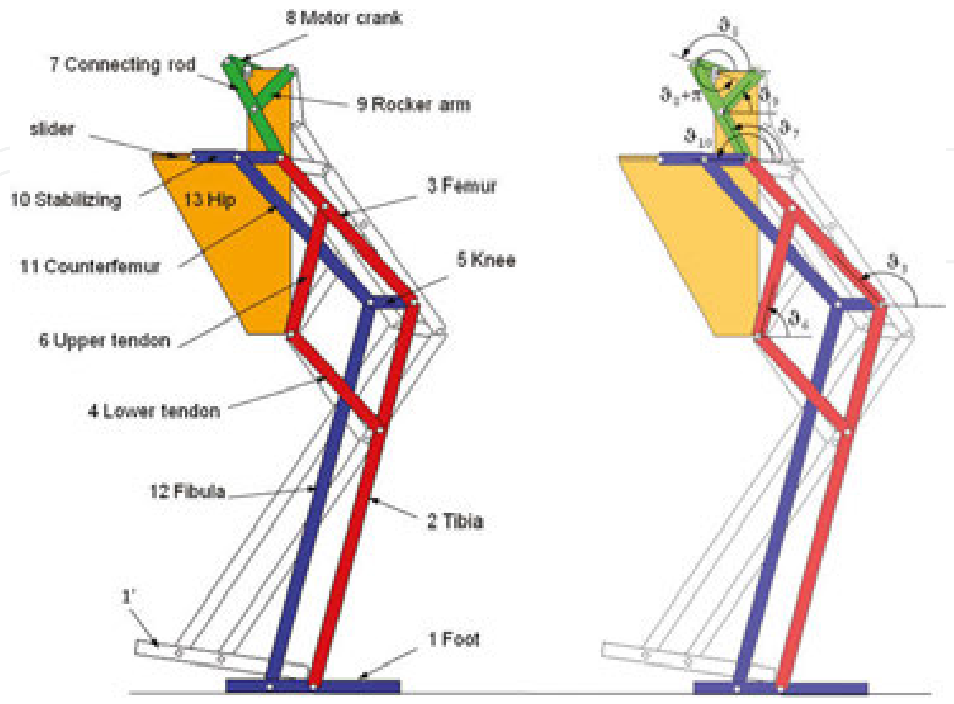


Figure 4. Sub-mechanisms of PASIBOT; nomenclature and numeration for the supporting leg and angular positions for the links.

As shown in **Figure 4**, any link belonging to the flying leg has the same name and number as the corresponding link from the supporting leg, but with a prime to distinguish between them. Each leg comprises 12 links, apart from the engine crank (link number 8), which is shared with both legs (no link number 8′ exists), so the biped PASIBOT has a total of 24 links, including the single hip (link number 13).

Listed below are the parameters and variables that describe the kinematics and dynamics of the biped:

l_i: length of the link i (mm)

m_i: mass of the link i (kg)

$_i$: angle between the link i and the hip (rad)

ω_i: rotational velocity of link i (rad/s)

α_i: rotational acceleration of link i (rad/s^2)

I_i: inertia moment for the link i (kg mm^2)

r_{ij}: position vector of the ij joint from the link i center of mass (mm)

r_{ijx}: x projection of the position vector (mm)

r_{ijy}: y projection of the position vector (mm)

f_{ij}: force exerted by the link i on the link j (N)

f_{ijx}: x projection of the f_{ij} (N)

f_{ijy}: y projection of the f_{ij} (N)

In **Figure 5**, a sequence for one step of PASIBOT is presented, as simulated with a commercial program. Note that one step corresponds to a half rotation (π rad) of the motor crank.

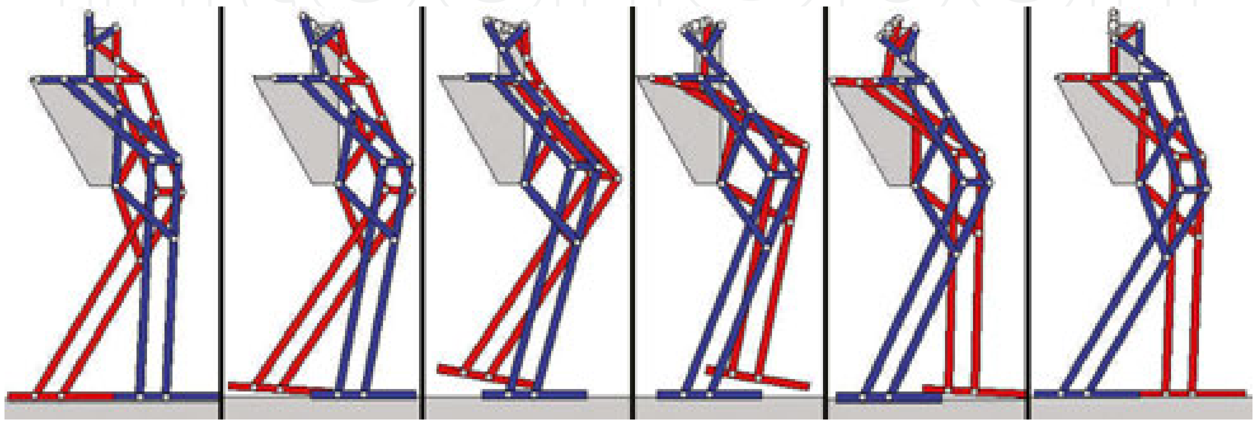


Figure 5. PASIBOT gait along one step (from $_8$= π to 3π/2 rad).

After defining the whole type of movement and the nomenclature, the kinematical study of one PASIBOT step is presented here. The kinematics is developed for the phase of "simple support," in which the supporting foot is in contact with the horizontal ground, whereas the other leg is flying. First, no sliding between the supporting foot and the ground is considered, so it can be considered part of the ground. Hence, the biped PASIBOT is a one DOF planar mechanism, and we can refer the angular positions of any link to the angular position of the motor crank (ω8):

$$\theta_i = \theta_i(\theta_8), i = 1,2,\ldots 1',2', \tag{1}$$

Then, the x,y coordinates for its center of mass can be easily expressed with respect to that angle:

$$x_i = x_i(\theta_8); y_i = y_i(\theta_8), i = 1,2,\ldots 1',2', \tag{2}$$

The angular velocities, accelerations and the center of mass linear velocities and accelerations are obtained by taking the first and second derivatives in Eqs. (1) and (2).

Actually, the biped kinematics is divided into three closed-loop kinematic chains:

1. Chebyshev chain (links number 7, 8, 9, and 13)

The distance between motor crank and rocker fixed points (A and B in **Figure 3**, respectively) in a Chebyshev mechanism is $l_{AB} = 2l_8$, the rocker arm length is $l_9 = 2.5l_8$, the connecting rod length is $l_7 = 5l_8$, and the rocker arm and connecting rod are joined at the middle point of the latter. The link lengths have been particularized for the designed biped, and normalized to the crank length, $l_8 = 1$. Taking into account these lengths, the Chebyshev closed-loop kinematic chain provides (see **Figure 6**):

$$2.5e^{j\vartheta_7} - 2.5e^{j\vartheta_9} - e^{j\vartheta_8} + 2 \tag{3}$$

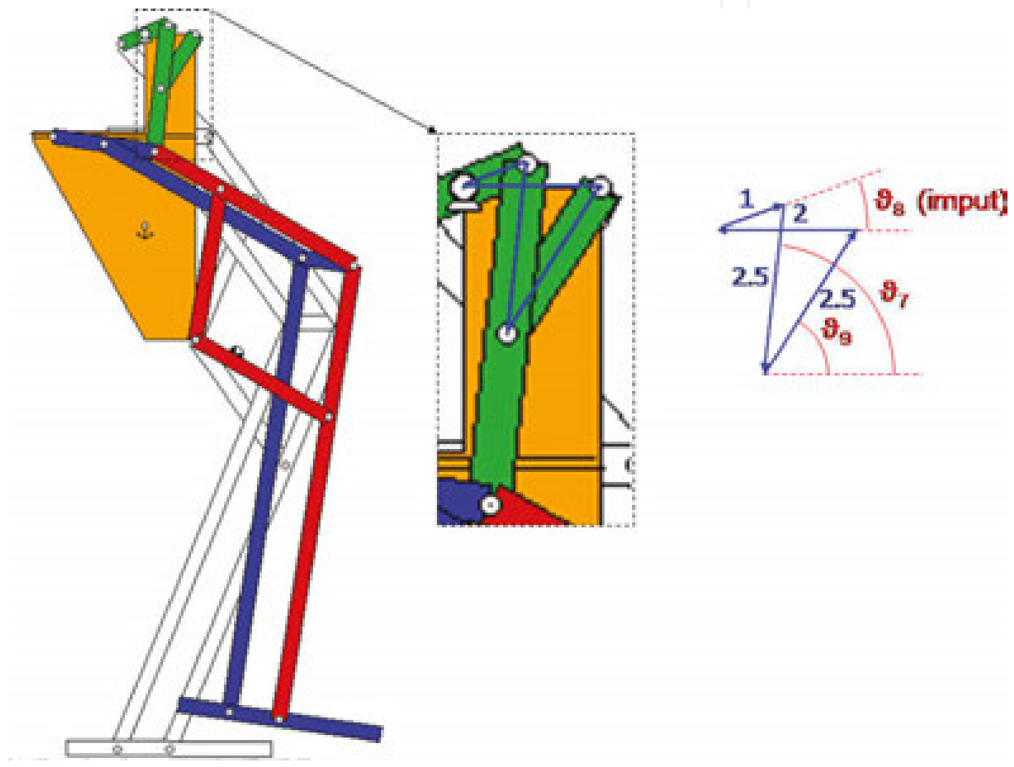


Figure 6. Chebyshev chain (lengths in units of l_8).

In Eqs. (3)–(5), both projections (vertical and horizontal) for each closed-loop equation are written in a compact form following the Euler's formula, where *j* is the imaginary unit.

2. Pantograph chain (links number 9, 7, 3, 6, and 13)

The tendons length is $l_4 = l_6 = 6l_8$, whereas the distance between the connecting rod-femur and upper tendon-femur joints (points C and F, respectively) is $l_{CF} = 3l_8$, and the distance between rocker arm-hip and upper tendon-hip joints (points B and D, respectively) is $l_{BD} = 12l_8$, so the pantograph closed-loop kinematic chain provides (see **Figure 7**):

$$6e^{j\vartheta_6} + 3e^{j\vartheta_3} + 2.5\left(e^{j\vartheta_7} + e^{j\vartheta_9}\right) - 12j = 0 \tag{4}$$

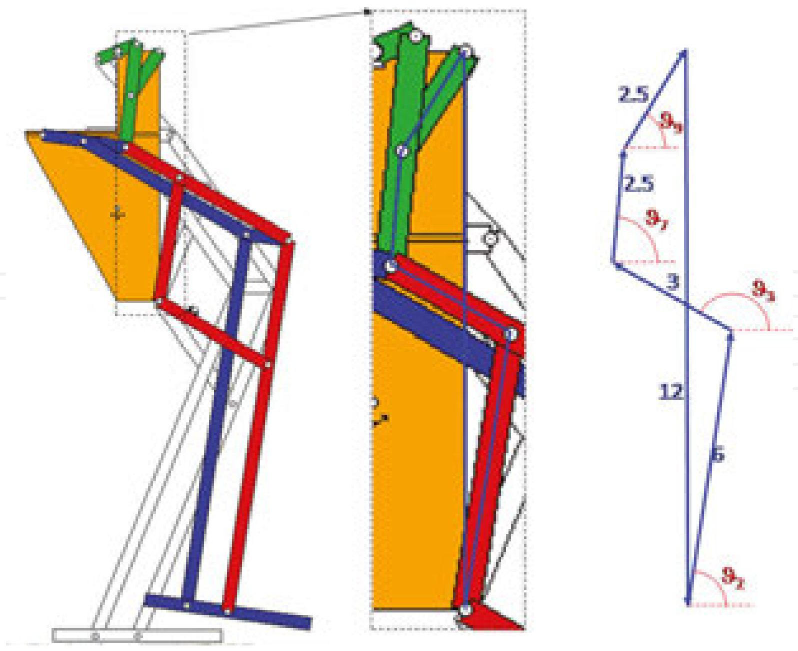


Figure 7. Pantograph chain (lengths in units of l_8).

3. Stability chain (links number 8, 7, 10, and 13)

In our model, the stabilizing link length is $l_{10} = 4.2l_8$. The vertical distance between the motor crank joint and the slot at the hip is $4l_8$. The horizontal projection distance between the motor crank joint and the end of the stabilizing link is called x. The stability closed-loop kinematic chain is as follows (see **Figure 8**):

$$24.e^{j\vartheta_{10}} - 5e^{j\vartheta_7} + e^{j\vartheta_8} - x + 4j = 0 \tag{5}$$

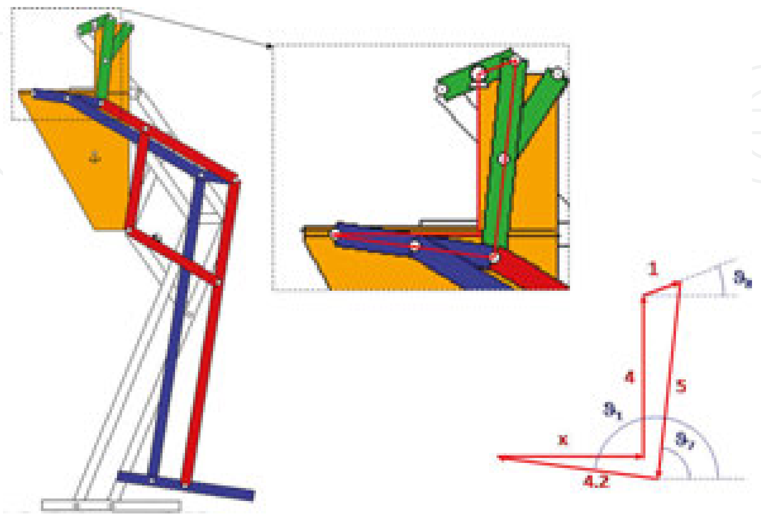


Figure 8. Stabilization chain (lengths in units of l_8).

As stated below, these equations determine the angles for all the links as functions of that for the motor crank, ϑ_8, which is also a function of time. Solving Eq. (3), the following expressions for the connecting rod and rocker arm angles are found:

$$\begin{cases} \vartheta_7 = a\cos\left[\dfrac{-4\cdot\cos^2\vartheta_8 + 13\cdot\cos\vartheta_8 - 10 + \sin\vartheta_8\cdot\sqrt{-16\cdot\cos^2\vartheta_8 - 60\cdot\cos\vartheta_8 + 100}}{25 - 20\cdot\cos\vartheta_8}\right] \\ \vartheta_9 = a\cos\left[\dfrac{-4\cdot\cos^2\vartheta_8 + 13\cdot\cos\vartheta_8 - 10 - \sin\vartheta_8\cdot\sqrt{-16\cdot\cos^2\vartheta_8 - 60\cdot\cos\vartheta_8 + 100}}{-25 + 20\cdot\cos\vartheta_8}\right] \end{cases} \tag{6}$$

From Eq. (4), the femur and tibia angles are found as functions of the previous ones:

$$\begin{cases} \vartheta_6 = a\cos\left[\dfrac{(2.5\cdot(\cos\vartheta_7 + \cos\vartheta_9)\cdot(-27-A) - 2.5\cdot(\sin\vartheta_7 + \sin\vartheta_9 - 12)\cdot\sqrt{144\cdot A - (-27-A)^2}}{12\cdot A}\right] \\ \vartheta_3 = a\cos\left[\dfrac{(2.5\cdot\cos\vartheta_7 + \cos\vartheta_9)\cdot(27-A) + 2.5\cdot(\sin\vartheta_7 + \sin\vartheta_9) - 12)\cdot\sqrt{36\cdot A - (27-A)^2}}{6\cdot A}\right] \end{cases} \tag{7}$$

where $A = (2.5\cdot(\cos\vartheta_7 + \cos\vartheta_9))^2 + (2.5\cdot(\sin\vartheta_7 + \sin\vartheta_9) - 12)^2$

Finally, Eq. (5) gives the solution for the stabilizing angle:

$$\vartheta_{10} = \operatorname{asin}\left(\frac{5\cdot\sin\vartheta_7 - \sin\vartheta_8 - 4}{4.2}\right) \tag{8}$$

As can be seen in **Figure 3**, the rest of the angles involved are identical to one of the given ones in Eqs. (6)–(8), in particular:

$$\begin{aligned} \vartheta_1 &= \vartheta_5 = \vartheta_{10} \\ \vartheta_{12} &= \vartheta_4 = \vartheta_3 \\ \vartheta_2 &= \vartheta_{13} = \vartheta_6 \end{aligned} \tag{9}$$

For the links belonging to the opposite leg, we apply a phase out of π radians on ϑ_8:

$$\vartheta_{i'}(\vartheta_8) = \vartheta_i(\vartheta_8 + \pi) \tag{10}$$

In order to reference all values to the ground, this corresponding base change must be applied:

$$\vartheta_i^{\text{ground}} = \vartheta_i - \vartheta_1, \tag{11}$$

where $\vartheta_i^{\text{ground}}\vartheta$ is the angle related to the ground system, and ϑ_i is the corresponding one related to the reference system fixed at link 14 (hip).

The positions of the center of mass for every link are obtained using trigonometric relations (e.g. x2=L2cosϑ2/2, y2=L2sinϑ2/2; x3=L2cosϑ2+L3cosϑ3/2, y3=L2sinϑ2+L3sinϑ3/2; etc.). Then, by time differentiating once and then twice, the angular velocity and acceleration as well as linear velocity and acceleration, respectively, for any link are calculated.

Thanks to apply this equations on MATLAB program, the kinematics of the biped robot PASIBOT can be solved for one step, considering a motor crank constant angular velocity,

$$\omega 8 : \vartheta 8(t) \;= \omega 8 \cdot t \tag{12}$$

The PASIBOT possessed a single DOF, so the positions of the center of mass of link *i* can be referred to the angular position of the motor crank (ϑ8):

$$\vartheta_i = \vartheta_i\left(\vartheta_8\right), i = 2,3\ldots 1', 2', \ldots (i \neq 8) \tag{13}$$

$$X_i = X_i\left(\vartheta_8\right), i \neq 1 \tag{14}$$

$$y_i = y_i\left(\vartheta_8\right), i \neq 1 \tag{15}$$

4. Slipping kinematics of the biped robot PASIBOT

If the supporting foot is allowed to slip, the PASIBOT becomes a 2-DOF mechanism (the biped moves across a plane, and the supporting foot supposedly remains horizontal). Eqs. (13) and (15) remain valid, while Eq. (14) increases by the value of the supporting foot slippage x_1 as follows:

$$\mathrm{x}_i = \mathrm{x}_1 + \mathrm{X}_i\left(\vartheta_8\right); \mathrm{i} \neq 1 \tag{16}$$

The first and second time derivatives of Eq. (16) are as follows:

$$\dot{x}_i = \frac{dx_1}{dt} + \frac{dX_i}{d\vartheta_8}\frac{d\vartheta_8}{dt} = \dot{x}_1 + X_i'(\vartheta_8)\dot{\vartheta}_8 \\ \ddot{x}_i = \ddot{x}_1 + X_i''(\vartheta_8)\dot{\vartheta}_8^2 + X_i'(\vartheta_8)\ddot{\vartheta}_8 \tag{17}$$

where a prime denotes explicit derivative with respect to ϑ_8, and dots denote time derivatives.

5. Non-slipping inverse dynamics of the biped robot PASIBOT

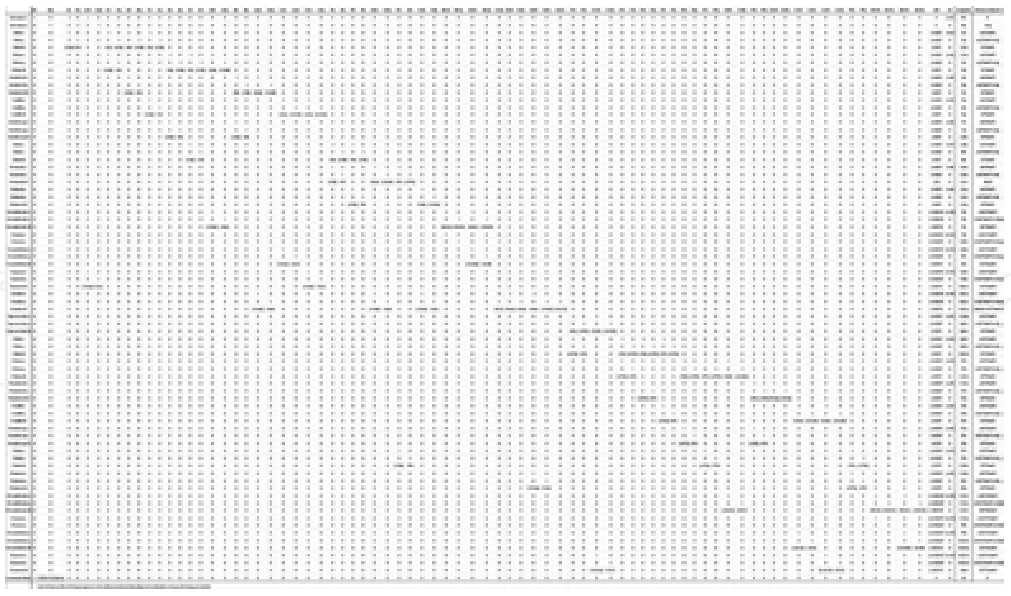
The inputs for the dynamical problem are the kinematic magnitudes (angular acceleration, α_i, and its center of mass acceleration, (a_{ix}, a_{iy})). The dynamical equations for the motion of every link, using Newton action-reaction law, are exposed in Eq. (18):

$$\left.\begin{array}{c} \sum_i \vec{F}_i = m_i \vec{a}_i \\ \vec{f}_{ij} = -\vec{f}_{ji} \\ \sum_i T_{on\ i} = I_i \alpha \end{array}\right\} \Rightarrow \left\{\begin{array}{c} \sum_{j<i} f_{ji_x} - \sum_{k>i} f_{ik_x} = m_i \ddot{x}_i \\ \sum_{j<i} f_{ji_y} - \sum_{k>i} f_{ik_y} = m_i g + m_i \ddot{y}_i \\ T_i + \sum_{j<i}\left(r_{ij_x} f_{ji_y} - r_{ij_y} f_{ji_x}\right) - \sum_{k>i}\left(r_{ik_x} f_{ik_y} - r_{ik_y} f_{ik_x}\right) = I_i \ddot{\theta}_i \end{array}\right. \\ i = 2,\ 3,\ \ldots,\ 13,\ 1', 2', \ldots, 7', 9', \ldots,\ 12 \tag{18}$$

There are three equations for each link (there are 23 links, excluding the supporting foot), and the system describing the dynamics of the whole mechanism consists of 69 linear equations. The system (Eq. 18) is expressed in a matrix form (Eq. 19), and then solved with a MATLAB® via matrix inversion:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & \\ a_{21} & a_{22} & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ & & \cdots & \end{bmatrix} \cdot \begin{bmatrix} f_{12_x} \\ f_{12_y} \\ \vdots \\ T_8 \\ \vdots \end{bmatrix} = \begin{bmatrix} m_2 \ddot{x}_2 \\ m_2 g + m_2 \ddot{y}_2 \\ I_2 \ddot{\theta}_2 \\ \vdots \end{bmatrix} \\ [A(\text{coefficient})] \cdot [F(\text{force})] = [I(\text{inertia})] \\ [F] = [A]^{-1} \cdot [I] \tag{19}$$

The code of the matrix that must be written in MATLAB® is shown in **Figure 9**.



[illegible]

Figure 9. The matrix A (coefficient).

Using MATLAB® code the kinematical and dynamical equations have been implemented in order to obtain solutions depending on a set of parameters (link dimensions, masses and densities, motor angular velocity) entered by the user. In **Figure 10**, the MATLAB flow chart of the kinematics and dynamics algorithm is shown.

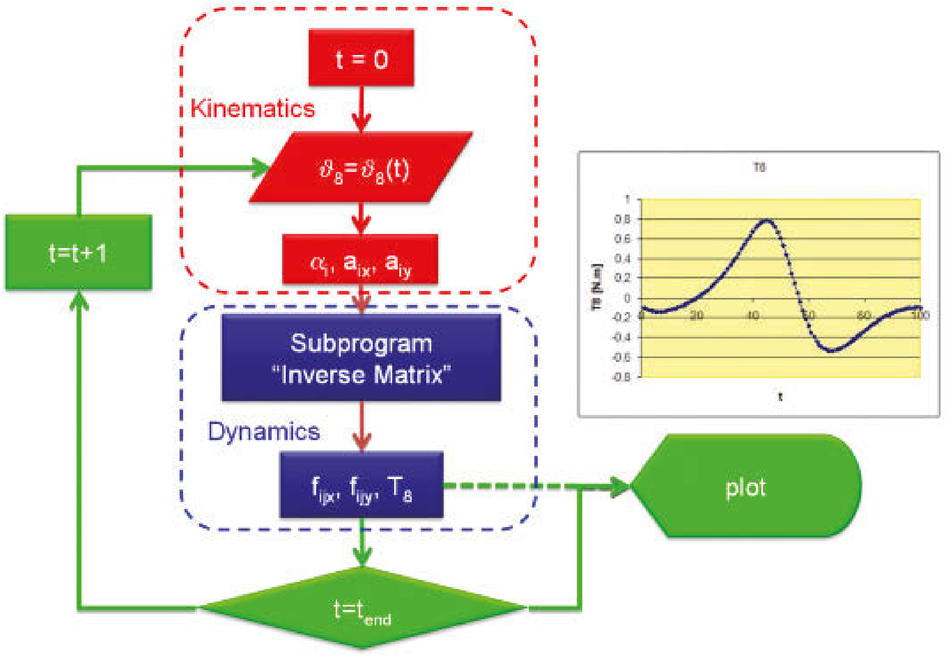

t = 0
Kinematics
$\vartheta_8=\vartheta_8(t)$
α_i, a_{ix}, a_{iy}
t=t+1
Subprogram "Inverse Matrix"
Dynamics
f_{ijx}, f_{ijy}, T_8
plot
t=t$_{end}$
T8
T8 [N.m]
t

Figure 10. MATLAB® flow chart for the PASIBOT kinematics and dynamics calculus code.

The MATLAB® program first finds the corresponding value of $\vartheta 8(t)$, and using Eqs. (6) to (11), it obtains the corresponding values of the rest of the angles and the positions of the centers of mass. Then, it calculates the kinematics. These data, which define the state of the biped at the time t, form the inertia matrix, (I), in Eq (19). Finally the MATLAB® program inverts the coefficient matrix, (A), by means of a matrix inversion subroutine and multiplies both matrices to provide the forces and torques between links at this time step. These values are stored to be plotted and the calculations are restarted for the next time step.

The results have been validated by comparison with others programs (working model and ADAMS code). The main advantage of the program developed via MATLAB® is that it lets us perform fast modifications, making the final robot design easier by changing parameters.

As the first result, the program implemented in MATLAB® has calculated the motor torque required to perform the movement. **Figure 11** shows the actuator torque in the crank (link number 8) related to time, for different values of the motor angular velocity, $\omega 8$.

In **Figure 11**, the same shape for each case can be appreciated apart from the torque obtained from the highest velocity value ($\omega 8$ = 5 rad/s). With this velocity, the dynamical effect of the inertia forces becomes important. However, for low speeds (below 3 rad/s) torque graphs hardly differ from one to another.

In **Figure 12**, the torque is represented again but for different loads (5, 10, and 15 kg) added to the hip. It is an interesting result. It shows that the required motor torque depends slightly on the added load. This is because the hip remains at almost the same level in a course of a step.

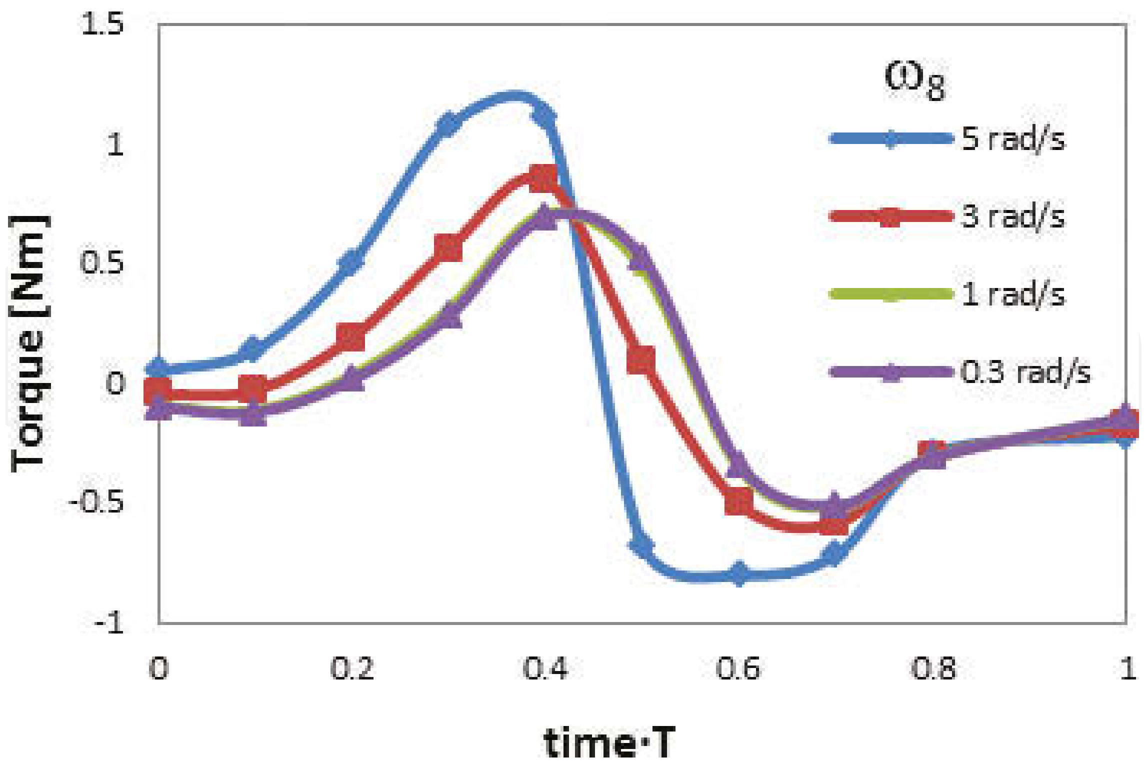


Figure 11. Torque for different crank velocities. *T* is the period for one step.

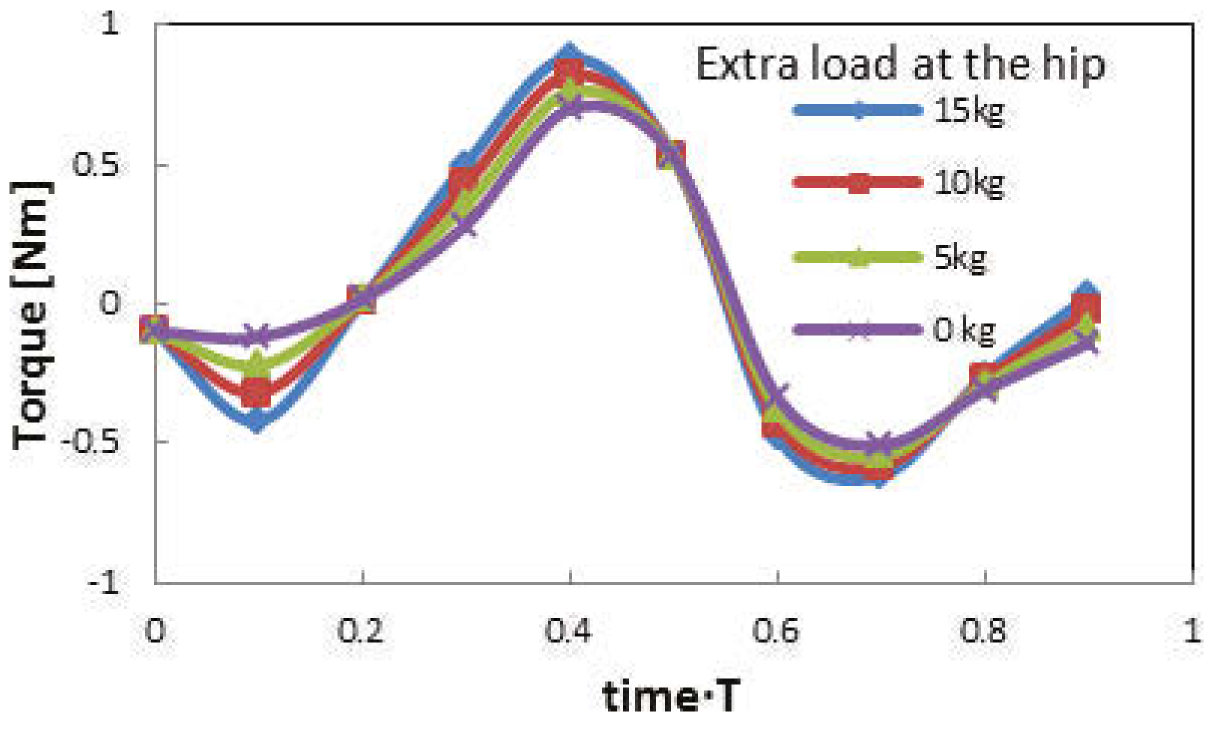


Figure 12. Actuator torque for different hip extra loads.

6. Forward dynamics for biped robot PASIBOT

The dynamics of mechanical systems can be modeled in two ways: inverse dynamics, which calculates the forces and torques that produce kinematics (movement), and forward dynamics, which computes the movement from known applied forces and torques.

When addressing forward dynamics, the kinematics is unknown. However, the angular position of the motor crank, ϑ_8, defines the position of the remaining links by Eqs. (20)–(22). These functions were defined in Eqs. (6)–(12). The corresponding angular velocities and accelerations as well as the center of mass linear velocities and accelerations are obtained from the corresponding first and second time derivatives:

$$\dot{\vartheta}_i = \frac{d\vartheta_i}{d\vartheta_8}\frac{d\vartheta_8}{dt} = \vartheta_i'(\vartheta_8)\dot{\vartheta}_8$$
$$\ddot{\vartheta}_i = \vartheta_i''(\vartheta_8)\dot{\vartheta}_8^2 + \vartheta_i'(\vartheta_8)\ddot{\vartheta}_8 \tag{20}$$

$$\dot{X}_i = X_i'(\vartheta_8)\dot{\vartheta}_8$$
$$\ddot{X}_i = X_i''(\vartheta_8)\vartheta_8^2 + X_i'(\vartheta_8)\ddot{\vartheta}_8 \tag{21}$$

$$\begin{gathered}\dot{y}_i = y_i'(\vartheta_8)\dot{\vartheta}_8 \\ \ddot{y}_i = y_i''(\vartheta_8)\vartheta_8^2 + y_i'(\vartheta_8)\ddot{\vartheta}_8\end{gathered} \tag{22}$$

When the kinematics is unknown, Eq. (19) becomes a system of second-order differential equations. To solve it numerically, in addition to time discretization, a motor crank angle ϑ_8 discretization is proposed. In this way, the derivatives of the known functions, $\vartheta_i = \vartheta_i(\vartheta_8)$, $X_i = X_i(\vartheta_8)$ and $y_i = y_i(\vartheta_8)$, are computed with respect to ϑ_8 as follows:

$$\begin{cases}\vartheta_i'(\vartheta_8) \cong \dfrac{1}{\Delta\vartheta_8}\left[\vartheta_i(\vartheta_8 + \Delta\vartheta_8) - \vartheta_i(\vartheta_8)\right] \\ \vartheta_i''(\vartheta_8) \cong \dfrac{1}{(\Delta\vartheta_8)^2}\left[\vartheta_i(\vartheta_8 + 2\Delta\vartheta_8) - 2\vartheta_i(\vartheta_8) + \vartheta_i(\vartheta_8)\right]\end{cases} \tag{23}$$

$$\begin{cases}X_i'(\vartheta_8) \cong \dfrac{1}{\Delta\vartheta_8}\left[X_i(\vartheta_8 + \Delta\vartheta_8) - X_i(\vartheta_8)\right] \\ X_i''(\vartheta_8) \cong \dfrac{1}{(\Delta\vartheta_8)^2}\left[X_i(\vartheta_8 + 2\Delta\vartheta_8) - 2X_i(\vartheta_8) + X_i(\vartheta_8)\right]\end{cases} \tag{24}$$

$$\begin{cases}y_i'(\vartheta_8) \cong \dfrac{1}{\Delta\vartheta_8}\left[y_i(\vartheta_8 + \Delta\vartheta_8) - y_i(\vartheta_8)\right] \\ y_i''(\vartheta_8) \cong \dfrac{1}{(\Delta\vartheta_8)^2}\left[y_i(\vartheta_8 + 2\Delta\vartheta_8) - 2y_i(\vartheta_8) + y_i(\vartheta_8)\right]\end{cases} \tag{25}$$

In fact, to implement the forward dynamic problem, Eqs. (23)–(25) are inserted into Eqs. (20)–(22) and then into Eq. (18). Thus, we obtain a system of equations in which the first and second time derivatives of ϑ_8 are unknowns, while the torque is now a known function of time, $T_8 = T_8(t)$. The resulting system of equations is as follows:

$$\begin{cases}\sum\limits_{j<i} f_{ji_x} - \sum\limits_{k>i} f_{ik_x} - m_i X_i'(\vartheta_8)\ddot{\vartheta}_8 = m_i X_i''(\vartheta_8)\dot{\vartheta}_8^2 \\ \sum\limits_{j<i} f_{ji_y} - \sum\limits_{k>i} f_{ik_y} - m_i y_i'(\vartheta_8)\ddot{\vartheta}_8 = m_i g + m_i y_i''(\vartheta_8)\dot{\vartheta}_8^2 \\ \sum\limits_{j<i} T_{ji} - \sum\limits_{k>i} T_{ik} + \sum\limits_{j<i}\left(r_{ij_x} f_{ji_y} - r_{ij_y} f_{ji_x}\right) - \\ -\sum\limits_{k>i}\left(r_{ik_x} f_{ik_y} - r_{ik_y} f_{ik_x}\right) - I_i\vartheta_i'(\vartheta_8)\ddot{\vartheta}_8 = I_i\vartheta_i''(\vartheta_8)\dot{\vartheta}_8^2\end{cases} \tag{26}$$

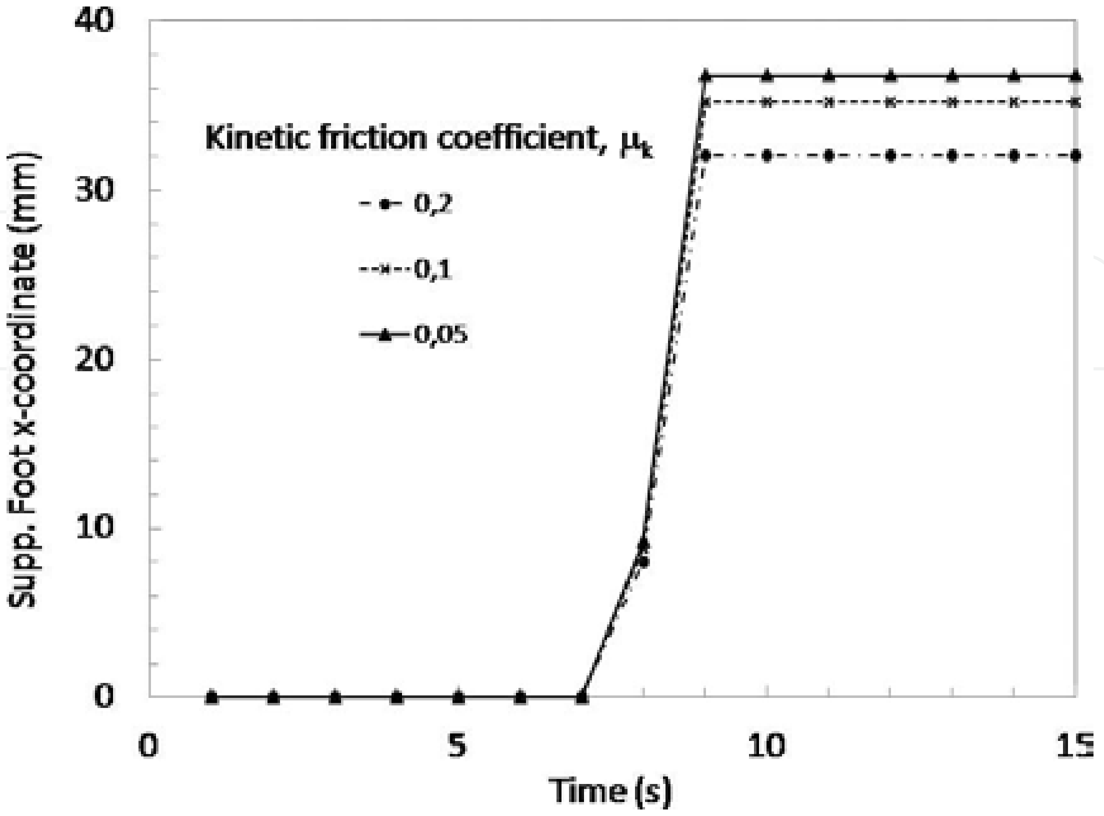


Figure 13. Evolution of the motor crank angle for different motor torques, when the biped starts walking from rest, according to the program described in this chapter (a) and according to Working Model 2D simulations (b).

To solve this system of differential equations, Eq. (26), a time discretization is used. For each time step, a linear inhomogeneous system is calculated, where the forces between links, and the angular acceleration of the motor crank, $\ddot{\vartheta}_8$, are unknowns, while the angular velocity of the motor crank, ϑ_8, is known. In fact, ϑ_8 is assigned an initial input, $\vartheta_8(t=0)$, which is updated after solving Eq. (26) in the previous time step, regarding the determined angular acceleration as constant during Δt. Then the updated angular velocity is as follows:

$$\dot{\vartheta}_8\left[n\Delta t\right] = \dot{\vartheta}_8\left[\left(n-1\right)\Delta t\right] + \ddot{\vartheta}_8\Delta t \tag{27}$$

Thus, the coefficient matrix is obtained from Eq. (19) by eliminating the column corresponding to the torque coefficients T_8 (previously, it was unknown), and adding a column representing the coefficients of the motor crank angular acceleration, $\ddot{\vartheta}_8$. The torque T_8 must now be included in the right-hand side (RHS) column vector. The forward dynamics system is obtained as

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,71} & 0 \\ a_{21} & a_{22} & \cdots & a_{2,71} & 0 \\ a_{31} & a_{32} & \cdots & a_{3,71} & -m_2 x_2{}'(\vartheta_8) \\ a_{41} & a_{42} & \cdots & a_{4,71} & -m_2 y_2{}'(\vartheta_8) \\ a_{51} & a_{52} & \cdots & a_{5,71} & -I_2 \vartheta_2{}'(\vartheta_8) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{71,1} & a_{71,2} & \cdots & a_{71,71} & -I_{12}\vartheta_{12}{}'(\vartheta_8) \end{bmatrix};\mathrm{U}=\begin{bmatrix} f_{01x} \\ f_{01y} \\ f_{12x} \\ f_{12y} \\ \vdots \\ . \\ f_{10'12'y} \\ \ddot{\vartheta}_8 \end{bmatrix};C=\begin{bmatrix} 0 \\ m_1 g \\ m_2 X_2{}''(\vartheta_8)\dot{\vartheta}_8^{\,2} \\ m_2 g + m_2 y_2{}''(\vartheta_8)\dot{\vartheta}_8^{\,2} \\ I_2\vartheta_2{}''(\vartheta_8)\dot{\vartheta}_8^{\,2} \\ \vdots \\ T_8 \end{bmatrix} \tag{28}$$

$$[A(\text{coeff. })]\cdot\left[U(\text{forces},\ddot{\vartheta}_8)\right]=[C(\text{cts})]\Rightarrow[U]=[A]^{-1}\cdot[C]$$

Note that the torque T_8 now appears in the RHS column vector (constants column). Now some results from the developed forward dynamics model are presented.

As shown in **Figure 13(a)**, the torque above which the biped can start walking is 0.84 Nm. The initial value of ϑ_8 is $\pi/2$ (both feet are on the floor).

These results were compared with those obtained with the software Working Model 2D. Comparing **Figure 13(a)** and **(b)**, we can say that the results are similar enough to each other to validate the program described in this chapter.

The MATLAB® program allows changing the density. In **Figure 14**, the motor crank angle is plotted for a constant torque, T_8 = 1 Nm and varying total weight (obtained by varying the density of all links).

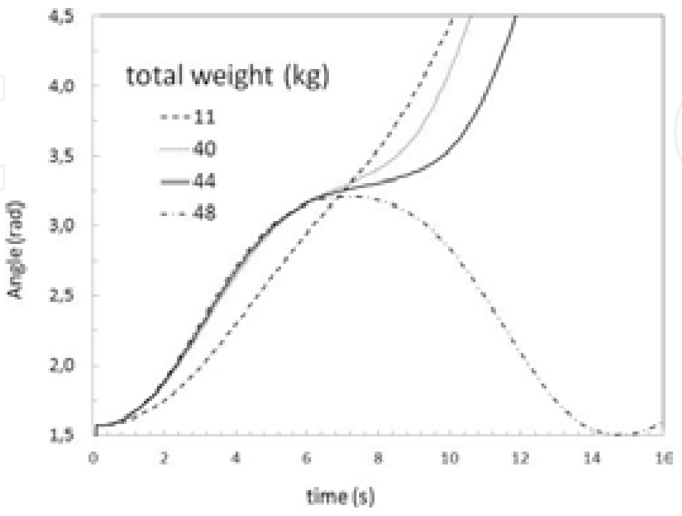


Figure 14. Evolution of the motor crank angle, for a constant motor torque, T_8 = 1 Nm, and for different total weights.

7. Dynamics of the biped robot PASIBOT including slippage between the ground and the supporting foot

7.1. Inverse dynamics with slippage

In the previous sections, we have applied inverse dynamics to parametrically calculate the required torque at the sole motor for PASIBOT to walk at a steady state (constant speed) with no sliding between the supporting foot and the floor. However, when sliding between the supported foot and the floor is allowed the kinematics is unknown and other approaches must be applied. In fact, three more equations regarding the supporting foot dynamics must be considered, as well as the kinematics-statics friction condition.

The forces acting on the supporting foot are

$$f_{01_x} - f_{12_x} - f_{112_x} = m_1 \ddot{x}_1 \, ; f_{12y} - f_{12y} - f_{112_y} = m_1 g \tag{29}$$

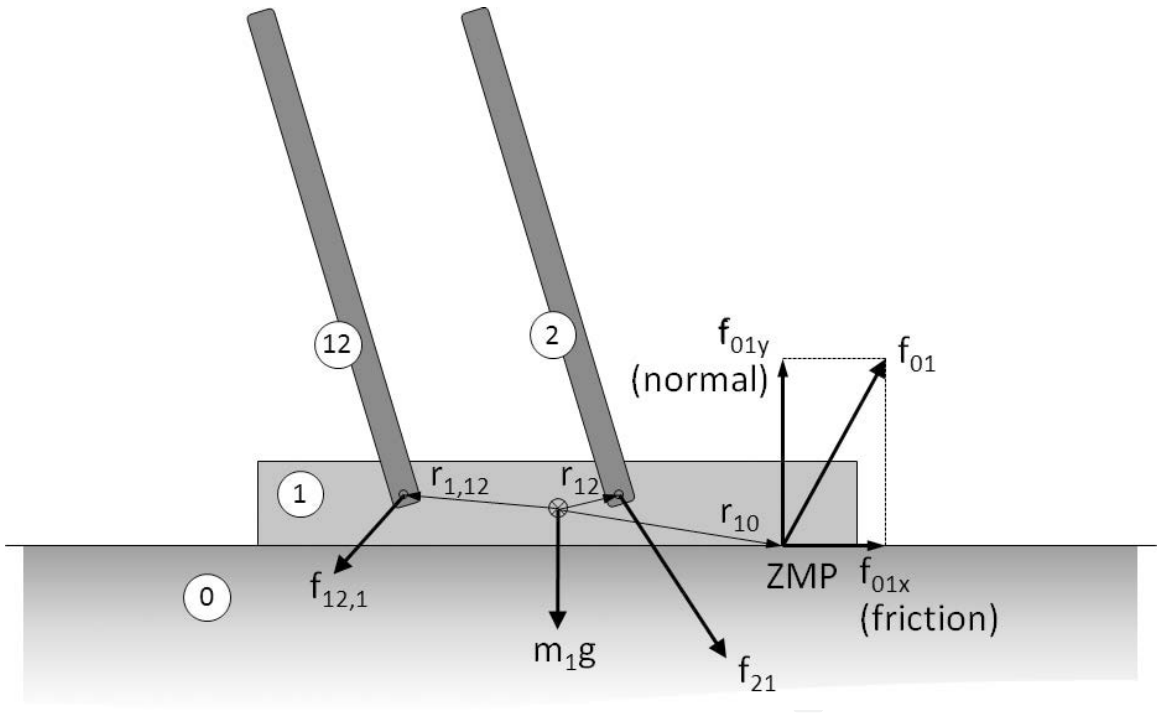


Figure 15. Forces acting on the supporting foot (link 1). Link 1 is connected to links 2 and 12. The floor is considered as link 0.

Since r_{10_x} and $f_{01_y} r$ are both unknown, Eq. (30) is the non-linear torque equation for the supporting foot (link 1; see **Figure 15**):

$$r_{10_x} f_{01_y} - r_{10_y} f_{01_x} - \left(r_{12_x} f_{12_y} - r_{12_y} f_{12_x} \right) - \left(r_{1,12_x} f_{1,12_y} - r_{1,12_y} f_{1,12_x} \right) = 0 \tag{30}$$

Eq. (30) is non-linear. If Eq. (18) is solved without Eq. (30), the latter can be used to obtain the instantaneous zero moment point (ZMP) relative to the center of mass, ZMPx = r_{10_x} r10x, which determines whether the biped topples.

In summary, the "inverse dynamic static friction equation" is obtained as follows in a matrix form:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,71} \\ a_{21} & a_{22} & \cdots & a_{2,71} \\ \vdots & \vdots & \ddots & \vdots \\ a_{71,1} & a_{71,2} & \cdots & a_{71,71} \end{bmatrix} \cdot \begin{bmatrix} f_{01_x} \\ f_{01_y} \\ f_{12_x} \\ f_{12_y} \\ \vdots \\ T_8 \\ \vdots \\ \end{bmatrix} = \begin{bmatrix} 0 \\ m_1 g \\ m_2 \ddot{X}_2 \\ m_2 g + m_2 \ddot{y}_2 \\ I_2 \ddot{\vartheta}_2 \\ \vdots \\ I_{12'} \ddot{\vartheta}_{12'} \end{bmatrix} \tag{31}$$

$$[A(\text{coefficients})] \cdot [F(\text{force})] = [I(\text{inertia})] \Rightarrow$$
$$\Rightarrow [F] = [A]^{-1} \cdot [I]$$

The forces and the motor torque in each time step are computed by solving Eq. (31) via matrix inversion encoded in MATLAB®.

When the supporting foot is allowed to slide, x_1 becomes an independent variable (in general, $x_1 \neq 0$, $\dot{x}_1 \neq 0$, $\ddot{x}_1 \neq 0$). To consider sliding between two bodies involves three possible scenarios: (1) no sliding (static friction), (2) imminent sliding, and (3) actual sliding. To determine the sliding status at each time step, conditional branching was incorporated into the code.

Initially, no sliding is assumed and the state is static friction ($x_1 = 0; \dot{x}_1 = 0; \ddot{x}_1 = 0$). Thus, solving Eq. (31), the values of friction force (f_{01_x}) and normal force (f_{01_y}) are calculated. Then, the static friction condition

$$\left| f_{01_x} \right| \leq \mu_s \left| f_{01_y} \right| \tag{32}$$

is evaluated for a specified static friction coefficient, μ_s. If Eq. (31) is satisfied, the time is incremented by Δt and Eq. (31) is recalculated using the updated values of r_{ij}, $\ddot{x}_i$, $\ddot{y}_i$, $\ddot{\vartheta}_i$. This step closes the "static friction" conditional loop.

If Eq. (32) is false, the PASIBOT enters the state of imminent slipping. This system has the following conditions: First, since the acceleration of the supporting foot, $\ddot{x}_1$, is no longer zero but unknown, it must appear in the column vector of unknowns. Second, the kinetic frictional relationship between normal and tangential components of the floor-foot force must be considered:

$$\left|f_{01_x}\right| = \mu_{\mathrm{k}}\left|f_{01_y}\right|, \tag{33}$$

where μ_{k} is the kinetic friction coefficient.

The new matrix of coefficients is then obtained from its predecessor by adding the following:

- A column of mi elements in positions corresponding to the x components of Newton's equations, with zeros elsewhere.
- An additional row incorporating Eq. (34).

Therefore, the final matrix form of the "inverse dynamic sliding friction equation" is as follows:

$$\left[\begin{array}{cccc:c} a_{11} & a_{12} & \cdots & a_{1,71} & -m_1 \\ a_{21} & a_{22} & \cdots & a_{2,71} & 0 \\ a_{31} & a_{32} & \cdots & a_{3,71} & -m_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{71,1} & a_{71,2} & \cdots & a_{71,71} & 0 \\ \hdashline 1 & \pm\mu_k & 0 & \cdots & 0 \end{array}\right] \cdot \begin{bmatrix} f_{01_x} \\ f_{01_y} \\ f_{12_x} \\ f_{12_y} \\ \vdots \\ T_8 \\ \vdots \\ f_{10',12'_y} \\ \ddot{x}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ m_1 g \\ m_2\ddot{X}_2 \\ m_2 g + m_2\ddot{y}_2 \\ I_2\ddot{\vartheta}_2 \\ \vdots \\ I_{12'}\ddot{\vartheta}_{12'} \\ 0 \end{bmatrix} \tag{34}$$

$$[A(\text{Coefficients})]\cdot[U(\text{Unknowns})] = [C(\text{Constants})] \Rightarrow [U] = [A]^{-1}\cdot[C]$$

IntechOpen

The sign of the friction force is depending on the sliding status in the previous calculation. The friction force opposes the horizontal component of the other forces acting on the foot when the previous state was imminent sliding. The friction force opposes the velocity of the supporting foot when the previous state was actual sliding.

With Eq. (34), the MATLAB® program calculates the acceleration with which the supporting foot has begun sliding, $\ddot{x}_1$. In the current time interval $((n-1)\Delta T - n\Delta t)$, the supporting foot is assumed to move with the calculated uniform acceleration. The velocity and position of the supporting foot are then updated by Eq. (35):

$$\begin{gathered}\ddot{x}_1 = \text{constant in the interval } \Delta t \\ \dot{x}_1[n\Delta t] = \dot{x}_1[(n-1)\Delta t] + \ddot{x}_1\Delta t \\ x_1[n\Delta t] = x_1[(n-1)\Delta t] + \dot{x}_1[(n-1)\Delta t]\cdot\Delta t + \frac{1}{2}\ddot{x}_1(\Delta t)^2\end{gathered} \tag{35}$$

The MATLAB® program obtain the kinematics and dynamics data (torque and force data) for all links, and then it increments the time by Δ*t* and re-solves Eq. (34). Note that the square brackets show the dependence.

When the PASIBOT is having a slippage, the friction force is against the supporting foot movement. This force is considered constant during this time interval, and it can stop the sliding of the PASIBOT but not change the direction of the movement. This is obtained with the stopping time, t_s, and compares it with the time increment, Δ*t*, as shown in Eq. (36):

$$t_s = -\frac{\dot{x}_1}{\ddot{x}_1} \tag{36}$$

If t_s is positive and less than Δ*t*, then friction stop the PASIBOT sliding before the end of the time interval. Else, if it becomes negative or exceeds Δ*t*, the PASIBOT continues sliding in the time interval. After that, the MATLAB® program updates the results using t_s instead of Δ*t* and returns to the beginning to solve the case of static friction, as provided in Eq. (31).

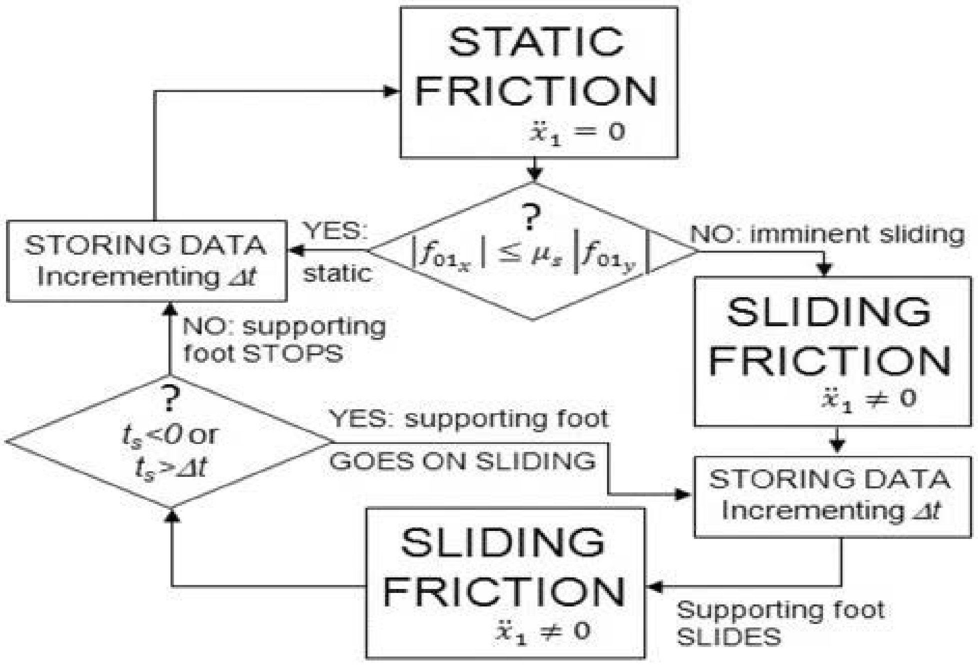

STATIC FRICTION $\ddot{x}_1 = 0$
?
$|f_{01_x}| \leq \mu_s |f_{01_y}|$
YES: static
STORING DATA Incrementing Δt
NO: imminent sliding
SLIDING FRICTION $\ddot{x}_1 \neq 0$
NO: supporting foot STOPS
?
$t_s<0$ or $t_s>\Delta t$
YES: supporting foot GOES ON SLIDING
STORING DATA Incrementing Δt
SLIDING FRICTION $\ddot{x}_1 \neq 0$
Supporting foot SLIDES

Figure 16. MATLAB® program flowchart with slippage.

In **Figure 16**, it is shown the MATLAB® program inverse dynamics flowchart with slippage. The forces between links and torque (dynamical variables) are calculated during the "STORING DATA" task. The position, velocity, and acceleration (kinematic unknowns) for the supporting foot are updated according to Eq. (34).

7.2. Forward dynamics with slippage

In the previous sections, we have applied this methodology to design the MATLAB® program of inverse dynamics with slippage. To obtain the forward dynamics program, the slippage is treated as in 30, while the dynamics is formulated as in 34. The MATLAB® flowchart showed in **Figure 16** is mostly maintained, but the systems of equations are different:

– The equation system to be calculated in the state of "STATIC FRICTION" is the forward dynamics system of Eq. (28): static system (ST).

– The equation system that describes the slippage of PASIBOT (Eq. 35) in the state of "SLIDING FRICTION," is added to Eq. (28). Also, the motor torque appears in the constants' column and the motor acceleration and the slippage acceleration become the penultimate and final elements of the unknowns' column. Using Eqs. (20)–(25), the first and second derivatives with respect to ϑ_8 of the position, velocity, and acceleration of every link are calculated, with a sufficiently fine discretization of ϑ_8. The resulting system of equations (36) is referred to as the "sliding system (SL)."

$$\left[\begin{array}{ccccc:c} a_{11} & a_{12} & \cdots & a_{1,71} & 0 & -m_1 \\ a_{21} & a_{22} & \cdots & a_{2,71} & 0 & 0 \\ a_{31} & a_{32} & \cdots & a_{3,71} & -m_2 x_2^{pf}{}'(\vartheta_8) & -m_2 \\ a_{41} & a_{42} & \cdots & a_{4,71} & -m_2 y_2{}'(\vartheta_8) & 0 \\ a_{51} & a_{52} & \cdots & a_{5,71} & -I_2 \vartheta_2{}'(\vartheta_8) & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & -I_8 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{71,1} & a_{71,2} & \cdots & a_{71,71} & -I_{12'} \vartheta_{12'}(\vartheta_8) & 0 \\ \hdashline 1 & \pm\mu_k & 0 & \cdots & 0 & 0 \end{array}\right] \cdot \begin{bmatrix} f_{01x} \\ f_{01y} \\ f_{12x} \\ f_{12y} \\ \vdots \\ . \\ f_{10',12'y} \\ \ddot{\vartheta}_8 \\ \ddot{x}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ m_1 g \\ m_2 X_2{}''(\vartheta_8)\dot{\vartheta}_8{}^2 \\ m_2 g + m_2 y_2{}''(\vartheta_8)\dot{\vartheta}_8{}^2 \\ I_2 \vartheta_2{}''(\vartheta_8)\dot{\vartheta}_8{}^2 \\ \vdots \\ T_8 \\ \vdots \\ . \end{bmatrix} \tag{37}$$

$$[A(\text{coeff. })]\cdot\left[U(\text{forces}, \ddot{\vartheta}_8, \ddot{x}_1)\right] = [C(\text{cts})] \Rightarrow [U] = [A]^{-1}\cdot[C]$$

– In "STORING DATA" mode, the kinematics of the supporting foot is updated by Eq. (37). The motor crank position and velocity is updated according to Eq. (35).

Some results are presented in the following figures, in which different friction coefficients and motor crank velocities have been considered. **Figure 17** plots the horizontal supporting foot position as a function of time for a constant motor crank angular velocity, ϑ_8=3rad/s, and varying friction coefficient (here $\mu_s = \mu_k \equiv \mu$). From this plot, we can deduce the time course of the supporting foot sliding.

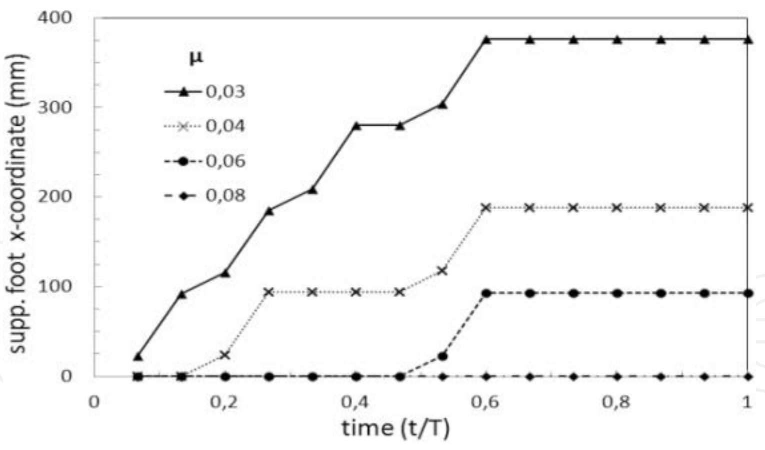


Figure 17. Supporting foot versus time (in units of one period, T) for constant $\vartheta_8 = 3$ rad/s, for a set of different friction coefficient.

In **Figure 17**, it is shown that the minimum friction coefficient that prevents the slippage is 0.08. Also, it can be observed that the slippage occurs during preferred phases. The sliding starts at mid-step until when the swinging leg has reached its highest point. If two slippages occur, one of them is again invoked at mid-step, while the other occurs at the first quarter step. For $\mu = 0.03$, slippage occurs repeatedly at various phases.

Figure 18 shows the sliding characteristics for constant friction coefficient $\mu = 0.1$ but different motor crank angular velocities.

Figure 18. Supporting foot versus time (in units of one period, *T*) for $\mu = 0.1$, varying ϑ_8.

Because the program is parametric it is easy to set different values for static and kinetic friction coefficients (kinetic friction coefficient is smaller than static friction coefficient). **Figure 19** shows the supporting foot of the PASIBOT with slippage for a static friction coefficient, μ_s = 0.2, and three different kinetic friction coefficient, μ_k = 0.2, 0.1, and 0.05.

Note that the bigger the kinetic friction coefficient, the smaller the sliding distance slippage occurs. Also, if there is slippage, it occurs at the same point for all three cases.

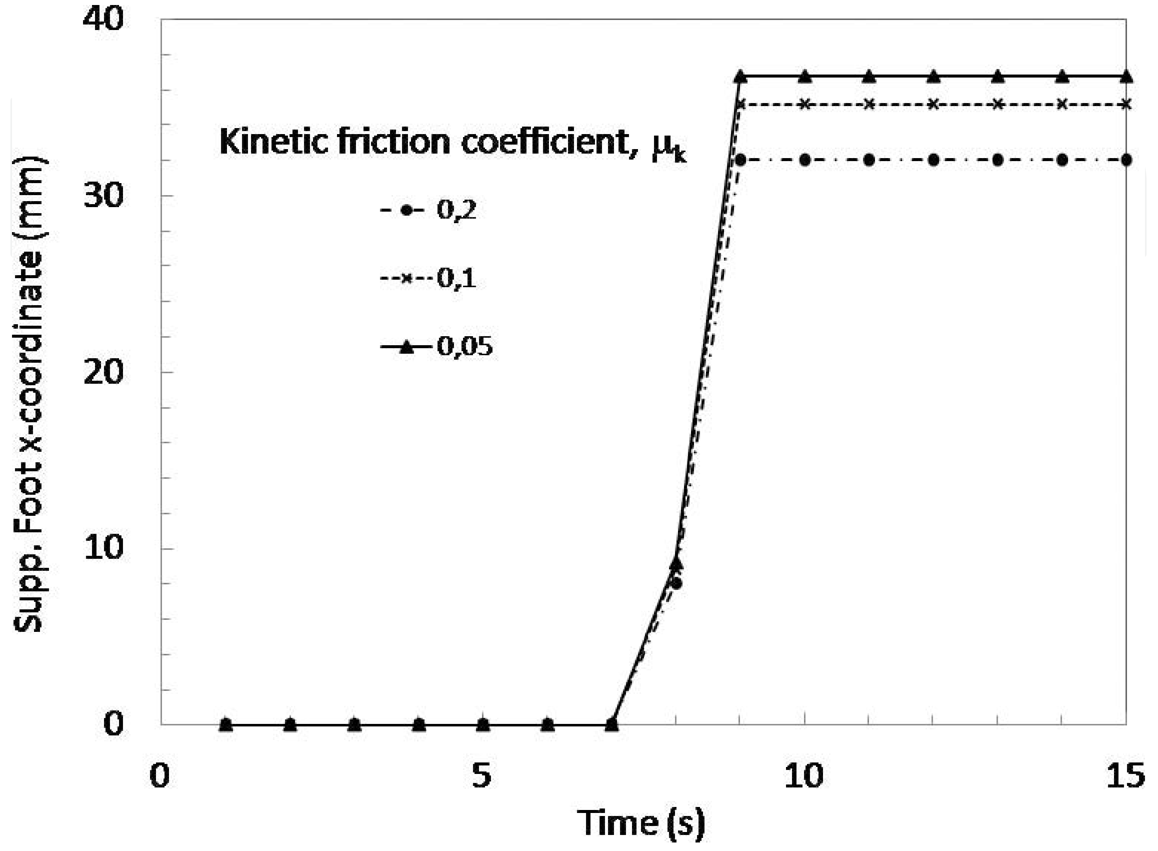


Figure 19. Supporting foot versus time for constant $\dot{\vartheta}_8$ = 5 rad/s, for the same static friction coefficient and for three different kinetic friction coefficients.

8. Applied quasi-static approach methodology to UGV

The quasi-static methodology is applied in this chapter. The main advantage of this approach is that it is easily optimized. It is applied to a vehicle in order to optimize the navigation capabilities. This methodology is applied to an UGV shown in **Figure 20**, which was designed and developed by the Tallinn University of Technology. This unmanned ground vehicle can change the angle between the body and the legs to improve the capabilities of passing obstacles or navigation. It changes the position of the center of mass (CoM) relative to the ground-wheel contacts, as well as the distance between the ground and the body [14].

In order to explain how to apply the methodology to implement this analysis in MATLAB® code, the nomenclature and geometry of the vehicle are presented. The position and/or trajectories of centers of mass, joints, and ground-wheel contact points are defined. Then the quasi-static model is developed and the equations to calculate the forces and torques involved are implemented in MATLAB®. The algorithm with the quasi-static equations obtains the position along the track for any configuration angles, and then calculates the optimal values of those angles that satisfy a given condition. If the vehicle slips or overturns at any point of the track, it is also calculated by the program [15, 16].

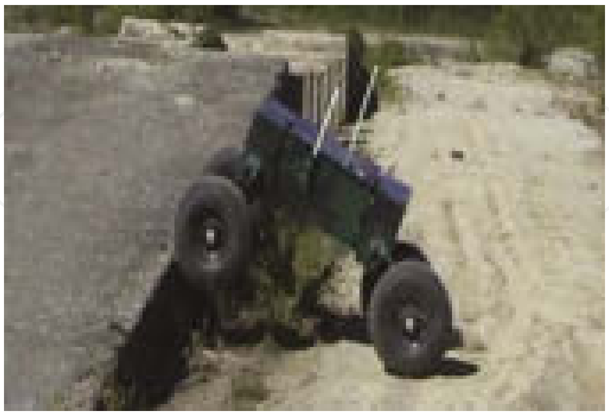


Figure 20. The unmanned ground vehicle (UGV) of Tallinn University of Technology.

The nomenclature is shown in the following list:

F_{lw}: force on the rear wheel exerted by the rear leg

F_{wl}: force on the front leg exerted by the front wheel

F_{r}: friction force exerted by the ground on the rear wheel

F_{f}: friction force exerted by the ground on the front wheel

N_{r}: normal force exerted by the ground on the rear wheel

N_{f}: normal force exerted by the ground on the front wheel

M: torque on the wheels (supposedly the same on front and rear wheels)

D: ground-front wheel contact point; T: ground-back wheel contact point

$L = 0.83$ m: body length; $l = 0.35$m: leg length; C: Center of mass

φ_{r}: angle between rear leg and body; φ_{f}: angle between front leg and body

β_{r},β_{f}: rear wheel contact angle, front wheel contact angle

$f(x)$: function defining the track profile

$g(x)$: function defining the trajectory for the centers of the wheels

$m_{\mathrm{w}} = 50$ kg: wheel mass; $m_{\mathrm{b}} = 300$ kg: body mass; $m_{\mathrm{l}} = 20$ kg: leg mass

First, the problem of positions is resolved. For a given ground function, with the front wheel position, x_{f} and the configuration angles, φ_{r} and φ_{f}, the back wheel can be located. The slope of the ground at x is obtained as: $\beta x = \tan\text{-}1[f'(x)]$. After the locus of the centers of the wheels, g(s) are calculated as follows (see **Figure 21**):

$$\begin{gathered} s = x - r\sin\beta_x \\ f(x) \rightarrow g(s) = f(x) + r\cos\beta_x \end{gathered} \tag{38}$$

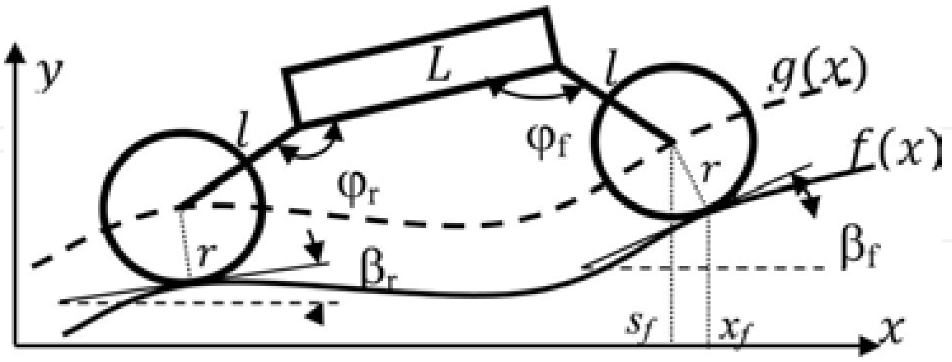


Figure 21. Position problem.

The front wheel center position (sf,g(sf)) is obtained from Eq. (38) and the wheelbase is solved as following:

$$R = \sqrt{\left(L - l\cos\cos\varphi_{\mathrm{r}} - l\cos\cos\varphi_{\mathrm{f}}\right)^2 + l^2\left(\sin\sin\varphi_{\mathrm{r}} - \sin\sin\varphi_{\mathrm{f}}\right)^2} \tag{39}$$

The intersection between the function g(s) and the circumference of radius *R* centered on the front wheel $(x_{\mathrm{f}} - r\sin\sin\beta_{\mathrm{f}},\ f(x_{\mathrm{f}}) + r\cos\cos\beta_{\mathrm{f}})$ locates the back wheel center (see **Figure 22**).

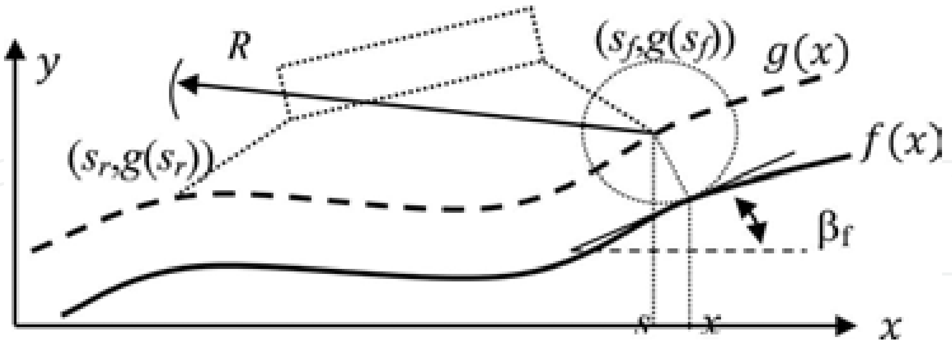


Figure 22. Scheme for locating the back wheel center.

Once the position of the vehicle is established, the locations of the CoMs are solved. The quasi-static approach can be calculated with three subsystems: (see **Figure 23**): (1) back wheel, (2) back wheel, both legs and body, and (3) the whole vehicle.

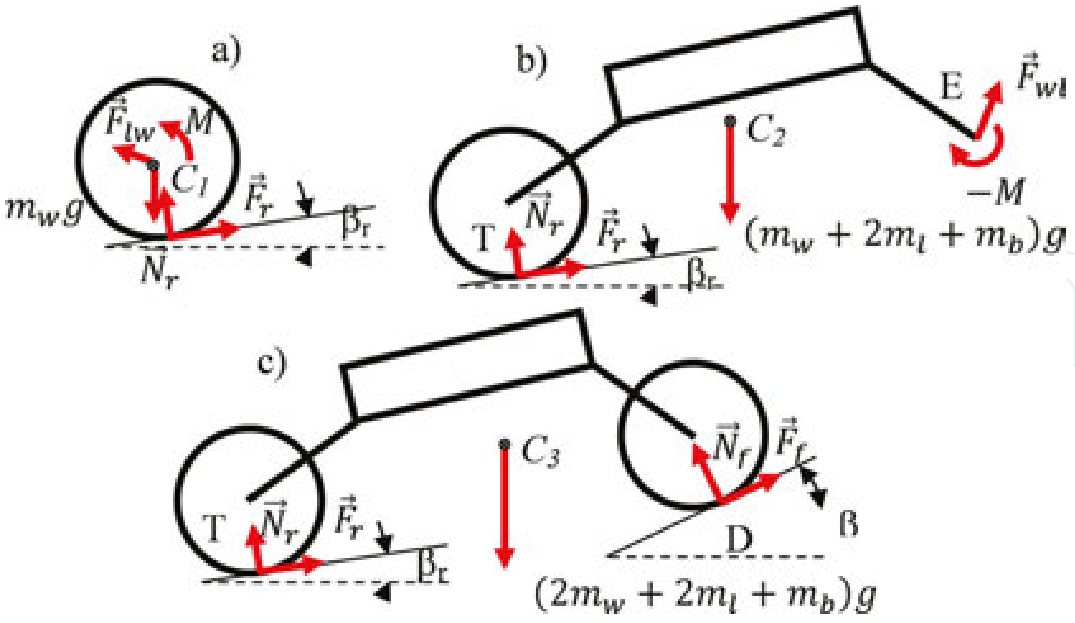


Figure 23. Vehicle subsystems.

For each subsystem, equilibrium requires two force equations and one moment equation:

$$\text{a)}\begin{cases} F_r \cos\cos\beta_r - N_r \sin\sin\beta_r + F_{\mathrm{lw}_x} = 0 \\ F_r \sin\sin\beta_r + N_r \cos\cos\beta_r + F_{\mathrm{lw}\,y} = m_w g \\ F_r r + M = 0 \end{cases} \tag{40}$$

$$\text{b)}\begin{cases} F_r Fr\cos\cos\beta_r \mathrm{Br} - N_r Nr\sin\sin\beta_r \mathrm{Br} + F_{\mathrm{wl}_x} = 0 \\ F_r Frf\mathrm{sin}f\mathrm{sin}\beta_r \mathrm{Br} + N_r Nr\cos\cos\beta_r \mathrm{Br} + F_{\mathrm{wl}_y} = (m_w + 2m_l + m_b)g \\ (C_2T)_x (F_r \sin\sin\beta_r + N_r \cos\cos\beta_r) - (C_2T)_y (F_r \cos\cos\beta_r \cos\cos\beta_r - N_r \sin\sin\beta_r \sin\sin\beta_r) + \\ +(C_2E)_y F_{\mathrm{wl}_x} - (C_2E)_x F_{\mathrm{wl}_y} - M = 0 \end{cases} \tag{41}$$

$$c)\begin{cases} F_r \cos\cos\beta_r - N_r \sin\sin\beta_r + F_f \cos\cos\beta_f - N_f \sin\sin\beta_f = 0 \\ F_r \sin\sin\beta_r + N_r \cos\cos\beta_r + F_f \sin\sin\beta_f + N_f \cos\cos\beta_f = (2m_w + 2m_l + m_b)g \\ (C_3T)_x (F_r \sin\sin\beta_r + N_r \cos\cos\beta_r) - (C_3T)_y (F_r \cos\cos\beta_r - N_r \sin\sin\beta_r) + \\ +(C_3D)_x \left(F_f \sin\sin\beta_f + N_f \cos\cos\beta_f\right) - (C_3D)_y \left(F_f \cos\cos\beta_f - N_f \sin\sin\beta_f\right) = 0 \end{cases} \tag{42}$$

where the unknowns are: F_r, N_r, F_f, N_f, F_{lw_x} Flwx, $F_{\mathrm{lw}y}$ Flwy, F_{lw_x} Fwlx, $F_{\mathrm{lw}y}$ Fwly and M. The torque is the same on front and back wheels. This system of nine equations can be simplified into a new system of five equations, where the unknowns are the torque, normal forces, and friction forces. And in a matrix form: A·F = C ⇒ F = A-1 C, where

$$A = \begin{array}{ccccc} \cos\beta_r & -\sin\beta_r & \cos\beta_f & -\sin\beta_f & 0 \\ \sin\beta_r & \cos\beta_r & \sin\beta_f & \cos\beta_f & 0 \\ r & 0 & 0 & 0 & 1 \\ C_2T\ _x \sin\beta_r - \ C_2T\ _y \cos\beta_r) & C_2T\ _x \cos\beta_r + \ C_2T\ _y \sin\beta_r & C_2E\ _x \sin\beta_f - \ C_2E\ _y \cos\beta_f & C_2E\ _x \cos\beta_f + \ C_2E\ _y \sin\beta_f & -1 \\ C_3T\ _x \sin\beta_r - \ C_3T\ _y \cos\beta_r & C_3T\ _x \cos\beta_r + \ C_3T\ _y \sin\beta_r & C_3D\ _x \sin\beta_f - \ C_3D\ _y \cos\beta_f & C_3D\ _x \cos\beta_f + \ C_3D\ _y \sin\beta_f & 0 \end{array}$$

$$F = \begin{bmatrix} F_{\mathrm{r}} \\ N_{\mathrm{r}} \\ F_{\mathrm{f}} \\ N_{\mathrm{f}} \\ M \end{bmatrix} C = \begin{bmatrix} 0 \\ (2m_{\mathrm{w}} + 2m_{\mathrm{l}} + m_{\mathrm{b}})\mathrm{g} \\ 0 \\ \left(\mathrm{C_2E}\right)_x \left(m_{\mathrm{b}}\mathrm{g}\right) \\ 0 \end{bmatrix} \tag{43}$$

MATLAB® code is used to solve the system of five equations. The program calculate for a set of discretized values of the front wheel position, the needed torque for any combination of a set of discretized values of the angles ϕ_{r} and ϕ_{d}. Thus, different criterions can be applied: minimize the energy to be supplied to the wheels, minimize the instantaneous torque or maximization the grip, the ground-wheel normal force, etc.

Some results are presented for a soft bump: square exponential profile, $f(x)=0.1e^{-(x-4)^2}$ (in meters). **Figure 24** shows the torque function that must be applied for any static configurations angles passing the soft bump.

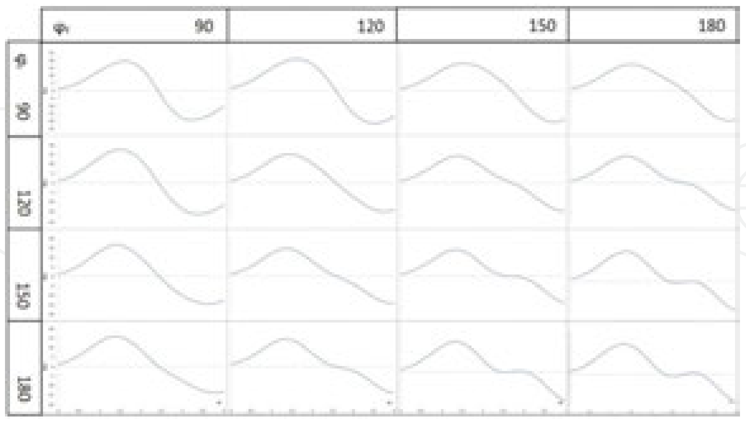


Figure 24. Torque needed passing a soft bump for different static configuration angles.

Figure 25(a) shows configuration angles variation needed to pass over the soft bump, with the minimum variation of torque and the corresponding torque function. **Figure 25(b)** shows the sequence of the UGV passing through the soft bump.

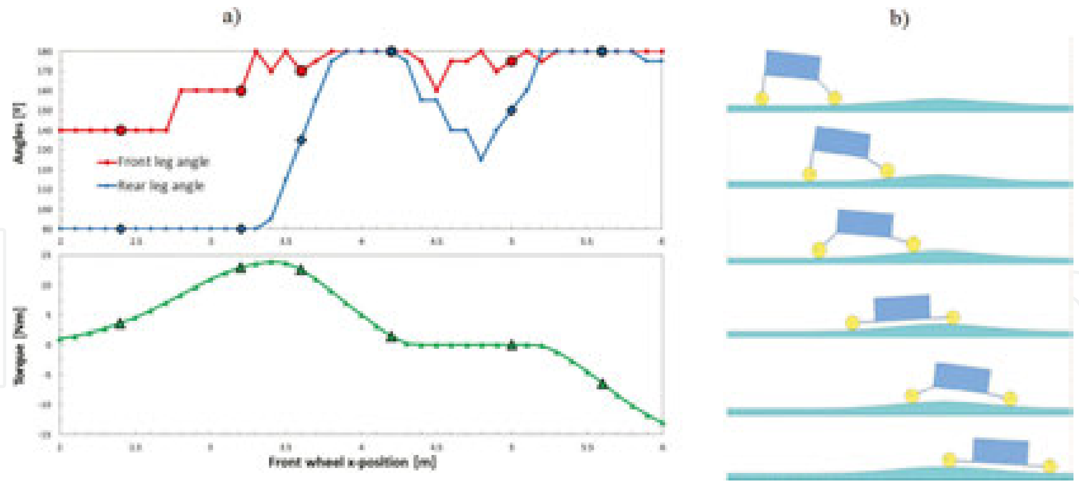


Figure 25. (a) Configuration angles and torque and (b) sequence of the UGV.

9. Conclusions

The methodology provided in this chapter can be applied to mechanisms, vehicles, or robots for the complete mechanical study. The kinematics and dynamics are solved using Newton-Euler equations, from the movement of the actuator to iterating during the time of initial condition as well as external forces, and with the quasi-static approach.

The programs have been designed so that all parameters can be modified. It was possible to automate these calculations creating an algorithm implemented in a programming language to simply find the solutions and the results of the analysis.

The methodology has been applied to design the biped robot PASIBOT. The kinematics and dynamics (both forward and inverse) of the biped robot "PASIBOT," taking into account for support foot slippage are encoded in MATLAB® code.

The great advantage of creating a parametric MATLAB® code following this methodology is that the algorithm can be modify to obtain the results in a parametric way or even changing the conditions easily. For example, it can calculate the motion of the biped from the torque function given by the biped's sole motor or the torques required for starting and braking as well as defining the conditions that prevent or control slippage.

Because the program remains parametric the lengths, densities and masses, motor velocities and torque, friction coefficients and other parameters can be modified by the user.

The methodology was also applied to another machine, a UGV vehicle, obtaining navigation optimization results. A numerical program based on a quasi-static half vehicle model is presented. For a given profile that could be read by sensors the program calculates how the angles between the body and the legs must vary, in order to fulfill the criteria like maintain as constant as possible the torque for example. The program created with this methodology in

MATLAB® code also can calculate the values of normal and friction forces, checking if the UGV rolls over or slip at any point.

In conclusion, this methodology can help to generate MATLAB® programs that will be valuable tools to optimize some navigation capabilities, dynamics analysis, quasi-static analysis, and slippage control among other.

Following this link the reader can find some examples of MATLAB codes done with the methodology of the chapter: http://www.mathworks.com/MATLABcentral/profile/authors/7854464-eduardo-corral

The code that calculates the inverse dynamic of the biped PASIBOT with slippage (using the methodology that we explain in the chapter) and the code that calculates the torque of the UGV and that optimized the best route are in the previous links.

Author details

E. Corral*, J. Meneses and J.C. García-Prada

*Address all correspondence to: eduardocorralabad@gmail.com

MAQLAB group, Universidad Carlos III de Madrid, Spain

References

[1] Corral, E., Meneses, J., Castejón, C., García-Prada, J.C. Forward and inverse dynamics of the biped PASIBOT. Int J Adv Robot Syst, 2014, 11:109. doi: 10.5772/58537

[2] Meneses, J., Castejón, C., Corral, E., Rubio, H., García-Prada, J.C. Kinematics and dynamics of the quasi-passive biped "PASIBOT". Strojniški vestnik J Mech Eng, 2011, 57, 12:879–887.

[3] Corral, E., Meneses, J., Rubio, H., Castejón, C., García-Prada, J.C. A configuration optimization algorithm based on quasi-static approach for a UGV. 17th International conference on climbing and walking robots, CLAWARS 2014, University of Technology, Poznan, Poland, 2014.

[4] Corral, E., Meneses, J., Garcia-Prada, J.C. Inverse and forward dynamics of the biped PASIBOT. International Symposium on Multibody Systems and Mechatronics, MUSME 2011, Valencia, España, 2011.

[5] Fujimoto, Y. Minimum Energy Trajectory Planning for Biped Robots, Humanoid Robots: New Developments, Armando Carlos de Pina Filho (Ed.), InTech, Rijeka, Croatia, 2007, ISBN: 978-3-902613-00-4, Available from: http://www.intechopen.com/

books/humanoid_robots_new_developments/minimum_energy_trajectory_planning_for_biped_robots

[6] Kappaganthu, L., Nataraj, C. Optimal Biped Design Using a Moving Torso: Theory and Experiments, Biped Robots, Prof. Armando Carlos Pina Filho (Ed.), InTech, 2011, ISBN: 978-953-307-216-6, Available from: http://www.intechopen.com/books/biped-robots/optimal-biped-design-using-a-moving-torsotheory-and-experiments

[7] Garcia de Jalon, J., Bayo, E. Kinematic and Dynamic Simulation of Multibody Systems – The Real-Time Challenge," Springer-Verlag, New York, 1993.

[8] Shabana, A.A., Computational Dynamics, Wiley, New York, United States, 2001.

[9] Shabana, A.A., Dynamics of Multibody Systems, 2nd ed., Cambridge University Press, New York, United States, 1998.

[10] Castejón, C., Carbone, G., García-Prada, J.C., Cecarelli, M. A methodology to design robotic arms for service tasks since early design stage, Int J Mech Control, 13, 02:73–83, 2012. ISSN: 1590-8844

[11] Castejón, C., Carbone, G., García-Prada, J.C., Ceccarelli, M. A multi-objective optimization of a robotic arm for service tasks. Strojniški vestnik – J Mech Eng, 2010, 56, 5:316–329.

[12] Gómez, M.J., Castejón, C., García-Prada. J.C., (2010). Evaluación de la adaptabilidad mecánica de los robots en entornos humanos (*Evaluation of Mechanical Adaptability of Robots in Human Environments*), in language spanish. Anales de Ingeniería Mecánica. Vol 01, Pages 187. ISSN: 0212-5072.

[13] Tokiwadai, Hodogayaku, Yokohama. Maintaining floor-foot contact of a biped robot by force constraint position control. Proceedings of the 2011 IEEE International Conference on Mechatronics, 2011.

[14] Universal Ground Vehicle , Research Project L523. Tallinn University of Technology , Department of Mechatronics , 2005–2008.

[15] Corral, E., Meneses, J., Aryassov, G. A quasi-static approach to optimize the motion of a UGV depending of the track profile. 9th International Conference, MSM, Vilnius, Lithuania, 2013.

[16] Corral, E., Aryassov, G., Meneses, J. A quasi-static approach to optimize the motion of an UGV depending on the track profile. Solid State Phenomena, 2015, 220–221:774–780. doi:10.4028/www.scientific.net/SSP.220-221.774