Employing a Novel Gait Pattern Generator on a Social Humanoid Robot

Ali Meghdari\textsuperscript{a,}* , Saeed Behzadipour\textsuperscript{b}, Majid Abedi\textsuperscript{c}

\textsuperscript{a}. Social & Cognitive Robotics Laboratory, Center of Excellence in Design, Robotics, and Automation (CEDRA), School of Mechanical Engineering, Sharif University of Technology, Tehran, IRAN, *Corresponding Author’s Email: meghdari@sharif.edu
\textsuperscript{b}. School of Mechanical Engineering, Sharif University of Technology, Tehran, Iran, Email: Behzadipour@sharif.ir
\textsuperscript{c}. School of Mechanical Engineering, Sharif University of Technology, Tehran, Iran, Email: majid.abedi92@yahoo.com

Abstract

This paper presents a novel Gait Pattern Generator developed for the “Alice” social humanoid robot which up to now lacked an appropriate walking pattern. Due to the limitations of this robot, the proposed gate pattern generator was formulated based on a nine-mass model to decrease the modeling errors; and the inverse kinematics of the whole lower-body was solved in such a way that the robot remains statically stable during the movements. The main challenge of this work was to solve the inverse kinematics of a 7-link chain with 12 degrees-of-freedom. For this purpose, a new graphical-numerical technique has been provided using the definition of the kinematic equations of the robot joints’ Cartesian coordinates. This method resulted in a significant increase in the calculations’ solution rate. Finally, a novel algorithm was developed for step-by-step displacement of the robot towards a desired destination in a two-dimensional space. Performance of the proposed gate pattern generator was evaluated both with a model of the robot in a MATLAB Simulink environment and in real experiments with the Alice humanoid robot.

Key Words

Social robots, bipedal robots, gait pattern generating, inverse kinematics, static stability condition
1. Introduction

In recent years, social humanoid robots have emerged with the goal to improve interaction between humans and environment [1-3]. Despite the complexities of manufacturing and controlling bipedal and humanoid robots, research on this class of robots is ongoing and growing [4-6]. Gait Pattern Generation (GPG) is one of the challenges of working with bipedal robots. High degrees of freedom, robot stability and the unique characteristics of each robot are some of the difficulties of GPG planning. Some important characteristics of the robot which affects the GPG are size and weight of the robot, number of degrees-of-freedom (DOF), sensors and actuators, data processing performance, and goals expected for the robot. However, gait pattern generating generally includes modelling, kinematic and dynamic analysis, stability condition, path planning, and controller design [7]. The final objective of a GPG is to generate robot joint trajectories that result in stable robot walking.

Stability is an important problem in GPG planning since ground contact forces must be compressive during the robot movement. In general, there are two types of stability conditions: static and dynamic [8-11]. In static stability, the Ground Projection of the Center of Mass (GCOM) of the robot is kept within the stability area (the convex hull of all contact points between the feet and the ground). Of course, static criterion is the oldest stability condition that was first suggested for a walking robot with massless legs [12]. However, it has been proven that it is necessary to consider the mass of the legs to determine the robot stability when the leg mass is great enough [13]. This condition is easier to implement, but the walking produced is usually too slow and not human-like because of excessive movement in the frontal plane. On the other hand, keeping the Zero-Moment-Point (ZMP) in the stability area is one of the dynamic conditions. Although ZMP is a normal stability criterion, it is not comprehensive. So other criteria, such as Foot Rotation Indicator (FRI) and Centroidal Moment Pivot (CMP), have been derived to improve the ZMP method [14]. Also, due to the complex dynamics of a humanoid robot, it is impossible to generate a stable walking pattern in real-time using these conditions. These conditions are usually used on
a simple model of the robot. For this purpose, a one-mass model (linear inverted pendulum mode [15]), a
two-mass model (gravity-compensated inverted pendulum mode [16]) and a three-mass model have been
suggested [17-18]. Although some analytic gait planning is achieved using these simple models, it is also
necessary to compensate for the modelling error. To this end, force sensors applied on the soles of the
robot are necessary to calculate the real ZMP position. However, if a more precise model (i.e. the multiple
masses inverted pendulum mode [19]) is used the ZMP equation becomes highly nonlinear and complex.
In this case, the robot joint trajectories are calculated by a time-consuming iterative procedure which
makes it inappropriate for generating a real-time walking pattern.

Solving the Inverse Kinematics (IK) of humanoid robots is also challenging because of their high
number of DOFs. Most commercial bipedal robots have 12 DOF in their lower-body (6 DOF in each leg)
[20-21]. There is no analytic IK solution for a whole lower-body with such a structure. But by using the
simpler models (one-, two- or three-mass model), it is possible to solve the IK of each leg (two six-DOF
chain) separately. Using these simplifications, the trajectory of the robot's body can be calculated
analytically. While it is possible to provide a closed-form IK solution for each leg for most commercial
biped robots (with 6 DOF in each leg) [22], this solution exists only if the three axes of the hip intersect
each other at one point (in which case, the hip is assumed as a spherical joint).

One should note that it is possible to generate a gait pattern analytically fast enough to use in real-time
applications, only if there is no offset at the hip of the robot, and the mass of the legs is light enough to use
the simple models. Otherwise, iterative methods are probably the only solutions. However, these methods
are generally not effective because of the expensive computation and large amount of time required to
achieve the desired accuracy. Therefore, it is necessary to find a solution to reduce the iterations. For
example, Limón et. al. [23] provided a two-step approach for a six-DOF leg with a non-negligible offset at
the hip. At First, an approximate IK solution was derived analytically regardless of the offset at the hip.
Then, this result was used as the initial condition of a numerical refining procedure.
In this research, a GPG is developed for the humanoid robot “Alice” with the given Iranian name “Mina” [24] as shown in Figure 1. This robot has been used as a therapy assistant in autism treatment and English as a Second Language education [25-28]. Although this robot has the capability for walking, no walking pattern has been designed for it thus far. To resolve this limitation, we found a walking pattern for Alice and implemented it.

According to the unique characteristic of this robot (especially the non-negligible offset at the hip, sensors and actuators of this robot, and the mass of the legs) discussed in section 2, a static stability with high accuracy has been used in the proposed GPG. To this end, the GPG has been designed based on a nine-mass model. Using this highly accurate model makes it necessary to solve the IK of the whole lower-body. Hence the main objective of this work was to solve the IK of a 12-DOF chain. This is not common in previously proposed GPG, because there is no analytical method for this problem. Previous iterative methods for this case would be very time consuming because forward kinematics (common in the numerical process of joint angles calculation) are very complicated for this 12-DOF chain. Also, it is not possible to find an approximate analytical solution as the initial condition for this chain by ignoring the offset at the hip (similar to the approach in [23]). Accordingly, a new graphical-numerical method has been provided in this paper to reduce the calculation time. A set of kinematic equations based on the robot geometry have been derived for this purpose. Finally, The GPG was completed with an algorithm for a step by step displacement of the robot towards a desired destination in a two-dimensional space [29].

2. The Alice “Mina” Humanoid Robot

Alice (renamed as Mina for a Persian dialogues) is a R50 model poorly designed and constructed by the Robokind Company [24, 30]. It has a height of 69 cm and the mass of its components are presented in Figure 1. Also, it has 32 DOF. This robot, similar to most commercial bipedal robots, has 6 DOF in each leg (3 in the hip, 1 in the knee and 2 in the ankle). So it is capable of walking on straight- and curved-paths,
walking up and down stairs and on sloped ground. These joints are actuated by Dynamixel RX-64. These servos are controlled through the internal computer of the robot. Java script is utilized to send the goal position of each joint to the internal computer. This is the only way of joints position control that is suggested for this robot. Due to this method, joint acceleration cannot be controlled and slow movements ensure that all joints will follow the desired trajectories simultaneously. Also, due to the mass of the robot legs (which is great enough to be considered) and the lack of force sensors in the robot soles (to compensate for the modelling error), a nine-mass model was used in the proposed GPG. These factors make it difficult to use a dynamic stability criterion. Fortunately, according to the goals expected for this robot [25-29], it does not need to walk too fast, so static stability has been used in this paper. Review of the construction of this robot (see Fig. 1) shows a 6.09 cm offset at the hip. Due to the length of the robot's leg an approximate gait planning based on spherical hip joints, which neglects this offset, results in a significant error in the robot walk. Since there are no force sensors in its soles, there is no analytic IK solution for each leg, and it is also not possible to compensate for the modelling error using a feedback control. In this paper, the proposed GPG is designed with regard to the mentioned important characteristics of this robot.

3. Gait Pattern Generating for the Alice Robot

3.1 Robot modelling

Kinematic and dynamic modelling is the first step in GPG planning. A model of the robot, with 7 links and 8 joint, is shown in Figure 2.a. The dimensions corresponding to Figure 3 can be seen in Table 1. In this model, it is assumed that the body and shoes are located in vertical and horizontal directions, respectively. The first assumption allows the robot to walk like a human. Also based on this assumption, the upper body is modelled by a single link (link 4). The second assumption ignores the height of the shoes. In this model, \( q_2, q_3, q_6 \) and \( q_7 \) each have one degree-of-freedom, and the other joints have two DOF. Also, the dynamic of the robot was modelled by 9 lumped masses. The weight and location of each lumped mass
(see Fig. 3) are presented in Table 1. In the next sections, the kinematics and dynamics of the robot have been analyzed according to this model.

A simulation environment was needed to evaluate the proposed GPG before implementing it on the robot. For this purpose, a more accurate model of the robot was designed in SolidWorks (see Fig. 2.b). Then, it was imported in MATLAB Simulink to run the GPG [29].

3.2 Inverse kinematics

The purpose of this section is to calculate the robot joint angles in a desired posture. The desired posture is determined by conditions such as coordinates of the shoes and the GCOM. Due to the high number of DOF, an infinite number of solutions will be achieved.

In this study, inverse kinematics were solved by a numerical method. Because the three axes of the hip don’t intersect each other in one point in this robot, it is not possible to provide a closed form solution for the inverse kinematics of each leg. Alternatively, the stability condition is an important criterion to define the desired postures of the robot. In this study, the static method was used as the stability condition. So, the inverse kinematics of the Lower-body, i.e. a 12-DOF kinematic chain, was solved to keep the GCOM of the nine lumped mass in the stability area.

3.2.1. Derivation of the Kinematics Equations

The derivation of kinematic equations of the robot is the first step in the numerical solution of inverse kinematics. To this end, forward kinematic equations can be used. First, the coordinate frames (see Fig. 4) were selected based on the commonly used convention Denavit-Hartenberg (DH) [31]. Thus, the base frame \( (X_0 Y_0 Z_0) \) and the final frame \( (X_{12} Y_{12} Z_{12}) \) were attached to the right and left shoes, respectively.

Then, \( H_{12}^0 \), the homogeneous transformation matrix that expresses the position and orientation of left shoe with respect to the right shoe was obtained as follows:
$$H^{0}_{12} = \prod_{i=1}^{12} H^{i-1}_{i}$$ (1)

Where, the homogeneous transformation $H^{i-1}_{i}$ is represented as a product of four basic transformations based on the DH parameters in Figure 4:

$$H^{i-1}_{i} = \text{Rot}_{z_{i-1}} \text{Trans}_{z_{i}} \text{Trans}_{x_{i-1}} \text{Rot}_{x_{i}}$$ (2)

Now it is possible to solve equation 1 numerically to obtain the joint angles. But because these equations are very complicated, motion generation would be very time-consuming. In this paper, the kinematic equations were derived in the joints’ Cartesian coordination ($\vec{q}_i = [x_i, y_i, z_i]^T$, $i=1:8$). These equations were much easier to solve numerically because they were shorter and without trigonometric functions. First, the distance between two joints was defined as follows:

$$\vec{P}_j = [X_j, Y_j, Z_j]^T = \vec{q}_j - \vec{q}_i$$ (3)

Then, the following seven equations were easily obtained:

$$|\vec{P}_j| = l_i^2, \quad j = i+1, \quad i = 1:7$$ (4)

But it is difficult to derive equations in Cartesian coordinates that indicate how the links are connected to each other by the joints. According to the robot structure (see Fig. 5), links 1, 2 and 3 always remain coplanar (plan $A_1$). Similarly, links 3, 4 and 5 and links 5, 6 and 7 constitute planes $B$ and $A_2$, respectively. The following equations indicate these conditions:

$$(\vec{P}_{12} \times \vec{P}_{23}) \cdot \vec{P}_{34} = 0$$ (5)

$$(\vec{P}_{34} \times \vec{P}_{45}) \cdot \vec{P}_{56} = 0$$ (6)

$$(\vec{P}_{56} \times \vec{P}_{67}) \cdot \vec{P}_{78} = 0$$ (7)

Robot kinematics was fully expressed by relations 4 to 7. But the system still had 8 degrees of redundancy. So, an infinite number of solutions will be achieved for a desired posture. But, only some of
the solutions are acceptable since the acceptable solution must satisfy the modelling assumptions and stability condition. According to the modelling assumptions, the body and shoes must be in vertical and horizontal directions, respectively. Also, the GCOM of the nine lumped mass should be kept in the stability area to keep the robot stable during the movement. In this study, an acceptable solution was selected by defining eight auxiliary equations.

At First, the solutions associated with robot stability should be selected. For this purpose, the static method was used as the stability condition, regardless of effects of velocity and acceleration in a slow walk. Suppose that \( \overrightarrow{comd} = [X_{comd} \ Y_{comd} \ 0]^T \) is the desired location of the GCOM. Relations 8 and 9 result in robot stability:

\[
\sum_{i=1}^{9} m_iX_{com_i} - X_{comd} \sum_{i=1}^{9} m_i = 0 \tag{8}
\]

\[
\sum_{i=1}^{9} m_iY_{com_i} - Y_{comd} \sum_{i=1}^{9} m_i = 0 \tag{9}
\]

Also, the body of the robot should be in a vertical direction. So, first plane B should be perpendicular to the ground, and second joints \( q_4 \) and \( q_5 \) must be at the same height. Equations 10 and 11 keep the robot body upright:

\[
Nb_z = 0 \tag{10}
\]

\[
z_4 = z_5 = H \tag{11}
\]

\( Nb_z \) is the Z-component of the vector orthogonal to the plane B \( \overrightarrow{n_B} \) which is obtained as follows:

\[
\overrightarrow{n_B} = \begin{bmatrix} Nb_x & Nb_y & Nb_z \end{bmatrix}^T = \overrightarrow{P_{43}} \times \overrightarrow{P_{45}} \tag{12}
\]

\( H \) is the desired height of the body. This has a major impact on the torque, temperature and energy consumption of the motors.

The orientation of the planes \( A_1, B \) and \( A_2 \) is another important point. Planes \( A_1 \) and \( A_2 \) are perpendicular to plane \( B \) when the robot walks in a straight line. But when the robot simultaneously walks
and rotates about the Z-axis, the orientation of the planes must be controlled.

The orientation of the body was determined by vector $\overrightarrow{P_{45}}$. To this target, vector $\overrightarrow{n_B} = [N_{b_x}, N_{b_y}, 0]^T$ was defined in the X-Y plane. The angle $\theta_B$ between $\overrightarrow{n_B}$ and the Y-axis was set as the rotation of the body about Z-axis. Then, equation 13 was used to make $\overrightarrow{P_{45}}$ parallel to $\overrightarrow{n_B}$:

$$N_{b_x}(y_5 - y_4) - N_{b_y}(x_5 - x_4) = 0 \quad (13)$$

Similarly, orientation of planes $A_1$ and $A_2$ was determined by the projection of vectors $\overrightarrow{n_{A_1}}$ and $\overrightarrow{n_{A_2}}$ in the X-Y plane. $\overrightarrow{n_{A_1}}$ and $\overrightarrow{n_{A_2}}$ are the vectors orthogonal to planes $A_1$ and $A_2$, respectively. Thus, $\theta_{A_1}$ and $\theta_{A_2}$ (the angles between the Y-axis and the projection of the vectors $\overrightarrow{n_{A_1}}$ and $\overrightarrow{n_{A_2}}$, respectively, in the X-Y plane) will be adjusted.

Definition of the eight auxiliary conditions decreased the degrees of redundancy of the system to zero, significantly reducing the number of solutions while allowing the desired posture for the robot to be achieved.

3.2.2. Calculation of the Joints' Coordinates:

In this stage, the kinematics and auxiliary equations are utilized to achieve the desired posture for the robot. To this goal, the unknown variables (joints’ coordinates) must be calculated. The specified variables are $q_1$ and $q_8$ because they are two inputs of the inverse kinematics process. Also, due to equation 9, $z_4$ and $z_5$ are specified by the body height. So, 16 other equations are utilized to calculate 16 other unknown variables. In this study, the Newton-Raphson Method was applied to solve this system of equations. So, if the parameters $q_1$, $q_8$, $\text{com}d$, $\theta_{A_1}$, $\theta_{A_2}$ and $\theta_B$ (input parameters of the IK process) are determined, starting from an initial condition the joints’ coordinates will be achieved.

3.2.3. Calculation of the Joint Angles
In the next stage, the joint angles which correspond to the joints' coordinates, must be determined. The following relations are derived according to model geometry (see Fig. 6):

\[
\theta_3 = \cos^{-1}\left(\frac{P_{21} \cdot P_{32}}{P_{21} \| P_{32}}\right)
\]

\[
\theta_4 = w_1 \cos^{-1}\left(\frac{P_{43} \cdot P_{32}}{P_{43} \| P_{32}}\right)
\]

\[
\theta_5 = \frac{\pi}{2} - \cos^{-1}\left(\frac{n_A \cdot n_B}{n_A \| n_B}\right)
\]

\[
\theta_6 = \cos^{-1}\left(\frac{P_{43} \cdot P_{45}}{P_{43} \| P_{45}}\right) - \frac{\pi}{2}
\]

\[
\theta_7 = \frac{\pi}{2} - \cos^{-1}\left(\frac{P_{54} \cdot P_{56}}{P_{54} \| P_{56}}\right)
\]

\[
\theta_8 = \frac{\pi}{2} - \cos^{-1}\left(\frac{n_A \cdot n_B}{n_A \| n_B}\right)
\]

\[
\theta_9 = w_2 \cos^{-1}\left(\frac{P_{56} \cdot P_{67}}{P_{56} \| P_{67}}\right)
\]

\[
\theta_{10} = \cos^{-1}\left(\frac{P_{78} \cdot P_{67}}{P_{78} \| P_{67}}\right)
\]

In the above relations, due to the range of motion of the robot’s joints, the \( \cos^{-1} \) function was used to calculate the angles. The sign of \( \theta_4 \) and \( \theta_9 \) were corrected by \( w_1 \) and \( w_2 \) as follows:

\[
w_1 = \text{sgn}(N_{b_x} \cdot x_{42} + N_{b_y} \cdot y_{42})
\]
\[ w_2 = \text{sgn}(Nb_x \cdot x_{47} + Nb_y \cdot y_{47}) \] (23)

To calculate the angles \( \theta_1 \) and \( \theta_2 \), three vectors \( \vec{f}_1 \), \( \vec{t}_1 \), and \( \vec{n}_1 \) were defined as shown in Figure 7. \( \vec{f}_1 \) is perpendicular to \( \vec{n}_1 \) and \( \vec{P}_{12} \cdot \vec{t}_1 \) is the vector along the common line of \( A_1 \) and the ground. \( \vec{n}_1 \) is perpendicular to \( \vec{Z} \) (the vector perpendicular to the ground) and \( \vec{t}_1 \). Thus, these three vectors are calculated according to the following equations:

\[
\vec{f}_1 = [f_{x_1} \quad f_{y_1} \quad f_{z_1}]^T = \vec{P}_{12} \times \vec{n}_1
\] (24)

\[
\vec{t}_1 = \vec{Z} \times \vec{n}_1
\] (25)

\[
\vec{n}_1 = [n_{x_1} \quad n_{y_1} \quad n_{z_1}]^T = \vec{t}_1 \times \vec{Z}
\] (26)

When \( \theta_1 \) and \( \theta_2 \) are equal to zero, \( \vec{Z}_1 \) and \( \vec{Z}_2 \) are parallel to \( \vec{f}_1 \) and \( \vec{n}_1 \), respectively. So, the right shoe is moved to a horizontal position by two successive rotations with the angles \( \theta_1 \) and \( \theta_2 \) as follows:

\[
\theta_1 = \text{sgn}(n_{z_1}) \cos^{-1} \left( \frac{\vec{n}_1 \cdot \vec{n}_1}{|\vec{n}_1||\vec{n}_1|} \right)
\] (27)

\[
\theta_2 = \text{sgn}(f_{z_1}) \cos^{-1} \left( \frac{\vec{f}_1 \cdot \vec{t}_1}{|\vec{f}_1||\vec{t}_1|} \right)
\] (28)

In the same way, \( \theta_{11} \) and \( \theta_{12} \) are calculated to move the left shoe to a horizontal position. So, it was found that if the parameters \( \vec{q}_1 \), \( \vec{q}_8 \), \( \text{comd} \), \( \theta_A \), \( \theta_A \), and \( \theta_B \) are determined (the inputs of inverse kinematics), joint angles will be achieved (the outputs of inverse kinematics). Using this ability, the GPG is designed in the next section.

4. GPG Planning

The goal of the GPG is to move the Alice robot to a desired position in the horizontal plane. So, the robot
displacements consist of moving along the $X$- and $Y$-axis ($\vec{g} = X\hat{I} + Y\hat{J}$) and rotating about the $Z$-axis ($\theta$).

To achieve this goal, the GPG sends appropriate commands to the Alice robot at any moment.

Since the outputs of the GPG are an appropriate control method for the Alice robot, it must send the desired joint angles to the robot (see Fig. 8). To this aim, the GPG first discretizes the whole displacement to several points. Then, the joint angles are calculated for all points and sent to the robot successively. This means that after positioning the servos in the desired angles at a point, the joint angles of the next point will be sent to the robot.

According to Figure 8, planning a path between the initial position and the desired position is the first step to generate the points. In this study, the shortest path to the desired position is selected (see Fig. 9.a). It consists of two rotations in the starting and ending position ($\alpha_1$ and $\alpha_2$), and a straight-line motion between them ($\vec{L}$). For this purpose, according to Figure 9.b, displacement vectors of the right and the left shoes ($\vec{g}_1$ and $\vec{g}_2$) were derived:

$$\vec{g}_1 = x_{g_1}\hat{I} + y_{g_1}\hat{J} = u_{g_1} + g + s_1 = \frac{l_4}{2}J + (X\hat{I} + Y\hat{J}) + \frac{l_4}{2}\sin(\alpha)\hat{I} - \frac{l_4}{2}\cos(\alpha)J$$

$$\vec{g}_2 = x_{g_2}\hat{I} + y_{g_2}\hat{J} = u_{g_2} + g + s_2 = -\frac{l_4}{2}J + (X\hat{I} + Y\hat{J}) - \frac{l_4}{2}\sin(\alpha)\hat{I} + \frac{l_4}{2}\cos(\alpha)J$$

Then, $\vec{L}$, $\alpha_1$ and $\alpha_2$ are calculated based on the sign of $Y$ as follows:

$$\vec{L} = g_2, \alpha_1 = \tan^{-1}\left(\frac{y_{g_2}}{x_{g_2}}\right), \alpha_2 = \alpha - \alpha_1 \quad \text{(if } Y > 0\text{)}$$

$$\vec{L} = g_1, \alpha_1 = \tan^{-1}\left(\frac{y_{g_1}}{x_{g_1}}\right), \alpha_2 = \alpha - \alpha_1 \quad \text{(if } Y < 0\text{)}$$
Because traveling along a path is not a continuous procedure for a bipedal robot, it is hard to provide a single algorithm to discretize the path to these positions. Bipedal robots move toward a destination by taking several steps with a maximum length limit. Each step includes several phases, such as the swing of each leg, switching a leg to swing mode, rotation of the robot body and rotation of each leg. As can be seen in Figure 8, the path is discretized to points in three stages. First, locations of feet placements on the ground are determined. This is done in the "path to steps" block so that the robot reaches the final position with minimum steps. Then, some phases combine to create the steps in the "steps to phases" block. Finally, each phase can be defined by a variation of one or more input parameters of the IK process (e.g. coordinates of swing foot) as a continuous function of time. This definition is done in the "phases to points" block to determine the points as robot positions at specific time intervals. For this purpose, polynomial functions were utilized. For example, in the straight line motion, coordinates of the swing foot ($q_1$ or $q_8$) have been defined as follows:

$$x = \frac{6L}{T^5}t^5 - \frac{15L}{T^4}t^4 + \frac{10L}{T^3}t^3$$

$$z = -\frac{64h}{T^6}t^6 + \frac{192h}{T^5}t^5 - \frac{192h}{T^4}t^4 + \frac{64L}{T^3}t^3$$

$i$ is 1 for swing of the right foot, and 8 for swing of the left foot. $l$, $h$ and $T$ are, respectively, the length, height and duration of a step. For smooth take-off and landing, coefficients were calculated in such a way that the velocity and acceleration of the swing foot are equal to zero at the beginning and end of the step.

Also, the coordinates of the GCOM (\([X_{com} \quad Y_{com} \quad 0]^T\)) that lead to the static stability of the robot during the movement were defined as depicted in Figure 10. a. GCOM is moved with a polynomial function of time to switch a leg to swing mode, Figure 10. b. Similarly, definition of $\theta_1$, $\theta_2$ and $\theta_3$ resulted in the starting and ending rotation.

In the next stage, the robot joint angles are calculated for all points. To this end, the inverse kinematics
mentioned in the last section is utilized. In this process, each point is used as the initial condition to calculate the next point. The calculated angles are saved in a matrix; then they are sent to the robot successively. The process mentioned in this section is evaluated in the next sections.

5. Results and Discussion

5.1 Simulation Results

The proposed GPG was run in the simulation environment to evaluate and correct it before implementing it on the robot. For this purpose, the GPG was coded in MATLAB. An example of the joint trajectories generated by the GPG is depicted in Figure 11. According to Figure 12, all the robot movements were correctly done so that the robot reached the desired position. Consequently, correctness of the algorithms of the path planning, point generating and the inverse kinematics was proven. Also, the robot stability during the movement indicates that the stability condition was applied properly.

Then this model was run in different average gait velocities ($V_{ave}$) to determine the maximum $V_{ave}$ where the robot remains stable. To this end, the center of pressure (COP) of the ground contact forces was determined during the movements. Also, the stability margin was defined as the smallest of distances from the COP to the edges of the stability area. The stability margin of the robot for walking with three different $V_{ave}$ is depicted in Figure 13. According to this evaluation, the stability margin reaches zero and the robot becomes unstable for $V_{ave}$ of more than 1 cm/s. Of course, the robot’s walk was very slow due to using the static stability condition.

5.2 Implementation of the GPG on the Robot

Finally, the GPG was coded in Java in NetBeans IDE and implemented on the robot. The robot movements indicated an appropriate performance of the IK, and acceptable accuracy of the robot kinematics modelling (see Fig. 14). Unfortunately, the robot frequently fell down while walking. Factors, such as the robot dynamics modelling and performance of the servos, were investigated as possible
reasons responsible for the robot falls.

Weighing the robot indicated a 55% error in the mass of the model. It seems that the masses reported in previous [29-30] (that were used in section 3.1) are wrong or incomplete. So the model was modified by trial and error. The mass of the components of the model were changed to improve the stability of the robot. Although, the robot falls decreased the stability problem was not completely solved.

Inappropriate contact between the shoes and the ground at the end of the swing phases was another reason for the robot’s instability. The investigation revealed that the shoes were not in the horizontal position while the robot was walking because of improper joint position tracking. Figure 15 shows the position tracking error of the 12 lower-body joints. It seems that the torques generated in the servos were not proper for tracking the desired positions. So, default PID coefficients should be tuned to achieve a better tracking performance. However, there was no way to resolve this problem because it was not possible to access to the controllers of the servos.

6. Conclusion
The purpose of this research was to generate a Gait pattern for the Alice humanoid robot. This pattern was specified according to the characteristics of the robot. First, a kinematic and dynamic model of the robot was developed. Then, inverse kinematics of the lower-body was solved using the Newton-Raphson numerical method. To this end, the kinematic equations of the robot were derived in the joints’ Cartesian coordination, and the static method was used as the stability condition. Finally, the GPG was developed for a step by step displacement of the robot towards a desired destination in a two-dimensional space.

One of the strengths of this method was using a multi-mass model that generates fewer errors than the one-, two- or three-mass models [15-18]. This method is more appropriate for robots that’s mass of their leg parts are not negligible in comparison with their body mass. In addition, the static method and the kinematic equations that were used in this paper resulted in a significant increase in the calculations’
solution rate compared to the algorithm suggested in [19]. This result was achieved even for a model with more complex kinematics (a three-dimensional model with non-spherical hip joints), although the produced walking was too slow. However, this problem did not have a serious conflict with the goals expected for this robot [25-28].

Simulations showed a good performance of the proposed displacement’s algorithms, inverse kinematics and stability condition. Finally, the GPG was implemented on the Alice robot. The robot’s acceptable movements indicated the suitability of the kinematic modelling. However, the stability of the robot was not acceptable. Factors disrupting the robot stability were identified and the walking pattern was reformed as much as possible. As a result of this research one can also conclude that the Alice humanoid robot manufactured by RoboKind Company [24 and 30] shall not be recommend as is for education and research due to its poor design, performance, and customer support.

Acknowledgement

This research was supported in part by a grant from the National Science Foundation of Iran-INSF (http://en.insf.org/) under contract #FY1395-1397 received by the corresponding author Ali Meghdari.

References


Biographies

Ali Meghdari received his Ph.D. in Mechanical Engineering from the University of New Mexico (UNM) in 1987. He then joined the robotics group of Los Alamos National Laboratory (LANL) as a Post-Doctoral research collaborator. In 1988, he accepted the position of Assistant Professor of Mechanical Engineering at Sharif University of Technology (SUT) in Tehran. From 1993-94, he was a visiting research faculty at the AHMCT center of the University of California-Davis, and during 1999-2000 he served the IBDMS research center at the Colorado School of Mines, and the Rocky Mountain Musculoskeletal Research Laboratory (RMMRL) as a visiting research professor. Professor Meghdari has performed extensive research in the areas of robotics, social and cognitive robotics, mechatronics, bio-robotics, nano-manipulation, and modelling of biomechanical systems. He has been the recipient of various scholarships and awards, the latest being: the 2012 Allameh Tabatabaei distinguished professorship award by the National Elites Foundation of Iran (BMN), the 2001 Mechanical Engineering Distinguished Professorship Award from the Ministry of Science, Research & Technology (MSRT) in Iran, and the 1997 ISESCO Award in Technology from Morocco. He was nominated and elected as a Fellow of the American Society of Mechanical Engineers (ASME) in 2001. He is currently the Director of the Centre of Excellence in Design, Robotics and Automation (CEDRA). Since 2005 he has been elected as an affiliate member of the Iranian Academy of Sciences (IAS).

Saeed Behzadipour received his BSc and MSc in Mechanical Engineering from Sharif University of Technology, Iran in 1998 and 2000, respectively. He received his PhD in Mechanical engineering from University of Waterloo, Canada in 2005. He was an assistant professor of Mechanical Engineering at the University of Alberta, Canada from 2005 to 2011. He is currently an associate professor of Mechanical Engineering at Sharif University of Technology. His research interests include medical robotics and rehabilitation technologies.

Majid Abedi received his MSc in Mechanical Engineering from Sharif University of Technology, Iran, Tehran in 2016. He is currently a PhD candidate in the School of Mechanical Engineering at Sharif University of Technology. His main research interests are robotics and rehabilitation.
list of captions:

Figure 1. Alice “Mina” robot and its components (dimensions are in centimetres) [24].

Figure 2. a) A 7-link model of the robot with 9 concentrated masses, b) Model of the robot designed in SolidWorks.

Figure 3. Masses and lengths of the model presented in Table 1.

Figure 4. DH coordinate frame assignment and DH parameters for the lower body of the Alice robot.

Figure 5. Connection of the model links.

Figure 6. The angles $\theta_3$, $\theta_4$, $\theta_6$, $\theta_7$, $\theta_8$ and $\theta_{10}$.

Figure 7. Definition of the vectors $\vec{f}_1$, $\vec{t}_1$ and $\vec{n}_1$.

Figure 8. The gait pattern generator (GPG).

Figure 9. a) Path planning for Alice robot. b) The shoes displacement vectors ($\vec{g}_1$ and $\vec{g}_2$).

Figure 10. a) GCOM trajectory in X-Y plane. b) GCOM displacement with a polynomial function of time.

Figure 11. Joint trajectories generated by the GPG.

Figure 12. Snapshots of the robot walking in the simulation environment.

Figure 13. Robot stability margins for walking with AGV of 1 cm/s, 1.2 cm/s and 1.5 cm/s.

Figure 14. Snapshots of the robot walking during the experiment.

Figure 15. Position tracking error of the 12 lower-body joints.

Table 1. Kinematic and dynamic characteristics of the 7-link model.
Figure 3:

![Figure 3 Image]

<table>
<thead>
<tr>
<th>Robot components</th>
<th>Mass (g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>shoes</td>
<td>505.14</td>
</tr>
<tr>
<td>Ankle</td>
<td>324.08</td>
</tr>
<tr>
<td>Shin</td>
<td>108.53</td>
</tr>
<tr>
<td>Thigh</td>
<td>352.02</td>
</tr>
<tr>
<td>Hip Yaw</td>
<td>62.10</td>
</tr>
<tr>
<td>Hip Roll</td>
<td>124.17</td>
</tr>
<tr>
<td>Waist</td>
<td>475.09</td>
</tr>
<tr>
<td>Torso</td>
<td>924.65</td>
</tr>
</tbody>
</table>
Figure 3:

![Diagram of a mechanical system with labeled dimensions and masses.]

Figure 4:

![Diagram of a mechanical system with labeled coordinate systems and angles.]

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\alpha_i$</th>
<th>$\alpha_i$</th>
<th>$d_i$</th>
<th>$\theta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>$\theta_1$</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>$l_{Sh}$</td>
<td>0</td>
<td>$\theta_2$</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>$l_{Th}$</td>
<td>0</td>
<td>$\theta_3$</td>
</tr>
<tr>
<td>4</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>$\theta_4$</td>
</tr>
<tr>
<td>5</td>
<td>-90</td>
<td>0</td>
<td>0</td>
<td>$\theta_5$</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>$l_g$</td>
<td>0</td>
<td>$\theta_6$</td>
</tr>
<tr>
<td>7</td>
<td>-90</td>
<td>0</td>
<td>0</td>
<td>$\theta_7$</td>
</tr>
<tr>
<td>8</td>
<td>-90</td>
<td>0</td>
<td>$-l_{Hy}$</td>
<td>$\theta_8$</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>$l_{Th}$</td>
<td>0</td>
<td>$\theta_9$</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>$l_{Sh}$</td>
<td>0</td>
<td>$\theta_{10}$</td>
</tr>
<tr>
<td>11</td>
<td>-90</td>
<td>0</td>
<td>0</td>
<td>$\theta_{11}$</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\theta_{12}$</td>
</tr>
</tbody>
</table>
Figure 5:

Figure 6:
Figure 7:

Figure 8:
Figure 11:

Figure 12:
Figure 13:

![Graph showing stability margin against time for different speeds.](image)

Figure 14:

![Sequence of images showing a robot moving on a cable.](image)
Figure 15:
Table 1:

<table>
<thead>
<tr>
<th>Model components</th>
<th>Length(cm)</th>
<th>Mass(kg)</th>
<th>Location of the center of mass(cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right shoe</td>
<td>0</td>
<td>$m_1 = 0.82922$</td>
<td>$\overline{\text{com}}_1 = \overline{q}_1$</td>
</tr>
<tr>
<td>Link 1</td>
<td>$l_1 = 9.57$</td>
<td>$m_2 = 0.10853$</td>
<td>$\overline{\text{com}}_2 = \overline{q}_2 + \frac{4.5}{l_1} + (\overline{q}_1 - \overline{q}_2)$</td>
</tr>
<tr>
<td>Link 2</td>
<td>$l_2 = 9.67$</td>
<td>$m_3 = 0.35202$</td>
<td>$\overline{\text{com}}_3 = \overline{q}_3 + \frac{4.63}{l_1} + (\overline{q}_2 - \overline{q}_3)$</td>
</tr>
<tr>
<td>Link 3</td>
<td>$l_3 = 6.09$</td>
<td>$m_4 = 0.0621$</td>
<td>$\overline{\text{com}}_4 = \frac{(\overline{q}_3 + \overline{q}_4)}{2}$</td>
</tr>
<tr>
<td>Link 4</td>
<td>$l_4 = 8.98$</td>
<td>$m_5 = 1.6481$</td>
<td>$\overline{\text{com}}_5 = \frac{(\overline{q}_4 + \overline{q}_5)}{2}$</td>
</tr>
<tr>
<td>Link 5</td>
<td>$l_5 = l_3$</td>
<td>$m_6 = m_4$</td>
<td>$\overline{\text{com}}_6 = \frac{(\overline{q}_5 + \overline{q}_6)}{2}$</td>
</tr>
<tr>
<td>Link 6</td>
<td>$l_6 = l_4$</td>
<td>$m_7 = m_3$</td>
<td>$\overline{\text{com}}_7 = \overline{q}_6 + \frac{4.63}{l_1} + (\overline{q}_7 - \overline{q}_6)$</td>
</tr>
<tr>
<td>Link 7</td>
<td>$l_7 = l_1$</td>
<td>$m_8 = m_2$</td>
<td>$\overline{\text{com}}_8 = \overline{q}_7 + \frac{4.5}{l_1} + (\overline{q}_8 - \overline{q}_7)$</td>
</tr>
<tr>
<td>Left shoe</td>
<td>0</td>
<td>$m_9 = m_1$</td>
<td>$\overline{\text{com}}_9 = \overline{q}_8$</td>
</tr>
</tbody>
</table>