POLYNOMIAL TRAJECTORY ALGORITHM FOR A BIPED ROBOT

Erik Cuevas,∗,** Daniel Zaldivar,∗,** Marco Pérez-Cisneros,∗ and Marte Ramírez-Ortegón**

Abstract

Building trajectories for biped robot walking is a complex task considering all degrees of freedom (DOFs) commonly bound within the mechanical structure. A typical problem for such robots is the instability produced by violent transitions between walking phases in particular when a swinging leg impacts the surface. Although extensive research on novel efficient walking algorithms has been conducted, falls commonly appear as the walking speed increases or as the terrain condition changes. This paper presents a polynomial trajectory generation algorithm (PTA) to implement the walking on biped robots following the cubic Hermitian polynomial interpolation between initial and final conditions. The proposed algorithm allows smooth transitions between walking phases, significantly reducing the possibility of falling. The algorithm has been successfully tested by generating walking trajectories under different terrain conditions on a biped robot of 10 DOFs. PTA has shown to be simple and suitable to generate real-time walking trajectories, despite reduced computing resources of a commercial embedded microcontroller. Experimental evidence and comparisons to other state-of-the-art methods demonstrates a better performance of the proposed method in generating walking trajectories under different ground conditions.

Key Words

Biped robots, trajectory generation, dynamic walking

1. Introduction

Robots must successfully deal with complicated environments such as rugged terrain, sloped surfaces, and steep stairs. Although it is assumed that biped robots can walk in almost any type of terrain surpassing some of the wheeled robots capabilities [1–3], they are complex nonlinear systems with many degrees of freedom (DOFs) that may fall down easily while walking due to its relatively small feet and other important design constraints.

A popular technique to implement biped walking is to keep the zero moment point (ZMP) constraint within a supporting feet polygon to ensure stable walking gaits [4]. Several methods have been proposed to generate walking trajectories satisfying this condition such as those referred in [5–17]. They fall into two groups: time-dependent and time-invariant algorithms. By far, the most popular algorithms are time-dependent which involve the tracking of pre-calculated trajectories. The second group requires the precise knowledge of biped dynamics to solve complex nonlinear models to generate walking patterns. This paper focuses only on the time-dependent algorithms.

Several techniques for generating walking motion for biped robots may be found in the literature. For instance, Ohishi et al. [11] approximated the biped movement as a three-dimensional inverted pendulum model (IPM), using the resulting points as tracking reference in Cartesian space. Some other algorithms with small variations of IPM can also be found. In [17] the IPM is transformed into a cart-table model, with the cart movements corresponding to the trajectory of the centre of mass (CoM). In [12] and [13], Kajita et al. demonstrated the use of a length-varying inverted pendulum as reference point to generate trajectories. The pendulum’s length varies as to keep the biped’s CoM at a constant height above the walking surface (see also [8]). Grishin et al. [14] used a pre-computed trajectory with online adjustment to improve stability of the biped robot. Other works such as those in [5–7, 10], use a different IPM to generate the walking trajectory. Most of the works on three-dimensional walking movements compute decoupled trajectories from the frontal and sagittal planes. Katoh and Mori [15] demonstrated that using a Van der Pol oscillator as generator of the tracking reference would induce walking trajectories for a biped robot. Moreover, Furusho and Masubuchi [16] presented the walking control algorithms by tracking a piecewise-linear joint reference trajectory. Another method for trajectory generation is to mimic the human rhythmic function by means of a central pattern generator, just as it is reported in [9]. In this method, one self-oscillating system is designed to generate synchronized periodic motions for each joint. Although extensive research on novel efficient walking algorithms has been conducted, the reported results still show a trend for falling down as walking speed increases or when terrain conditions change. Slow down the walking pace will be a temporary solution as the problem remains unsolved as walking deterioration is inflicted on subsequent
walking phases. Some previous works \[18\] have shown the negative effect of a violent impact between the feet and the ground which yields a reaction force \(F_R\) and increases the possibility of falls. The effect can be even worse, when the robots’ velocity is increased or the terrain conditions change. Although some IPM algorithms have attempted to solve the problem, they require extensive computation unsuitable for real-time applications.

This paper presents the polynomial trajectory algorithm (PTA) which is a simple and effective walking trajectory algorithm based on cubic Hermite polynomial interpolation of the kinematics positions of the robot. A Hermite spline (HS) is a cubic polynomial interpolation in segments with adjustable derivatives at each control point that allows decreasing link’s velocities when they do reach the target point. Therefore, the impact on the leg caused by the ground contact or by violent transitions among different walking phases may also be reduced. The walking trajectory is generated for each joint, adjusting some intermediate positions to assure the best ZMP trajectory. The resulting approach generates suitable bipedal walking gaits in real time considering only modest computing resources, such as a microcontroller embedded platform. Experimental evidence shows the effectiveness of the method to generate walking trajectories under different ground conditions. The proposed algorithm was successfully tested on a biped robot of 10 DOFs \[19\] under different walking conditions. This paper presents a comparison between several state-of-the-art methods and it demonstrates a better relationship between the link velocities and the reaction forces produced by the proposed method. It reduces the ZMP instability created by violent transitions between walking phases such as the swinging leg impacting the surface.

This paper is organized as follows: Section 2 introduces the biped robot model and the walking gait. Section 3 discusses on the impact model and its velocity discontinuity. Section 4 describes PTA and its parameters while Section 5 features the walking motions generated by the PTA, despite terrain changes. This section also discusses on comparing the PTA performance to other related methods. Finally, Section 6 draws some conclusions.

2. Robot Model and Bipedal Walking

2.1 Robot Description

The dynamics of a biped robot is closely related to its structure \[20\]. This work employs the “Dany Walker” robot \[19\], built on 10 low-density aluminium links. Each link consists of a structure which has been carefully designed to allow an effective torque transmission and low deformation. All links are connected within the biped robot structure of 10 DOFs as shown in Fig. 1. The motors allow movements within the frontal and sagittal plane. Figure 1(a) shows the frontal plane of the robot while the Fig. 1(b) shows the sagittal plane. The embedded control computer system is based on a PIC18F4550 microcontroller executing concurrent tasks such as trajectory generation, servo-motor control, and sensor signal collection. The ZMP is estimated through data integration from pressure sensors (Flexiforce type) located on each foot. The centre of pressure (CoP) matches the ZMP if and only if the latter falls inside the supporting foot polygon (SIP). In the “Dany Walker” biped robot, the ZMP was found by using feedback from triangular force sensor arrangement as it is described in \[19, 21\]. Each mass is assumed to be located on the midpoint of its corresponding link. The parameters of the robot model according to the convention presented in \[22\] are summarized in Table 1 following locations drawn in Fig. 2.

The foot’s length and width are 10cm and 8cm, respectively. Let position \((x_s, y_s, z_s)\) be middle point of the supporting foot. Thus, the robot motion may be expressed with respect to the reference frame whose centre is \((x_s, y_s, z_s)\). The \(x\)-directional ZMP \((x_{ZMP})\) and the \(y\)-directional ZMP \((y_{ZMP})\) should be located inside the region defined on \(-5cm < x_{ZMP} < 5cm\) and \(-4cm < y_{ZMP} < 4cm\). For the single support phase, such region is a convex hull for all contact points between the foot and the ground \[10, 23\]. Thus, if the ZMP falls within this area, the biped robot can walk without falling down \[24\]. However, it is difficult to calculate the ZMP from a 3D robot model as in Fig. 1 because of the coupling motion between the frontal plane \((y-z\) plane) and sagittal plane \((x-z\) plane). Therefore, the walking motion may be generated from two 2D models including the sagittal plane model with 8 segments and 7 DOFs as shown in Fig. 2(b). The frontal plane model has 6 segments and 5 DOFs as shown by Fig. 2(c). The ZMP equations of the sagittal plane robot model (Fig. 2(b)) and the frontal plane robot model (Fig. 2(c)) are:

\[
x_{ZMP} = \frac{\sum_{i=1}^{7} m_i(\ddot{z}_i + g)x_i - \sum_{i=1}^{7} m_i \ddot{x}_i z_i}{\sum_{i=1}^{7} m_i(\ddot{z}_i + g)}
\]

\[
y_{ZMP} = \frac{\sum_{i=1}^{7} m_i(\ddot{z}_i + g)y_i - \sum_{i=1}^{7} m_i \ddot{y}_i z_i}{\sum_{i=1}^{7} m_i(\ddot{z}_i + g)}
\]

with \(g\) being the gravity, \((x_i, y_i, z_i)\) and \(m_i\) being the position and mass of the \(i\)-th point mass \((i = 1, \ldots, 7)\) \[7\]. In (1), the inertia factors can be ignored assuming that
Table 1
Parameters of the Biped Robot

<table>
<thead>
<tr>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$m_3$</th>
<th>$m_4$</th>
<th>$m_5$</th>
<th>$m_6$</th>
<th>$m_7$</th>
<th>$l_1$</th>
<th>$l_2$</th>
<th>$l_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2 Kg</td>
<td>0.3 Kg</td>
<td>0.3 Kg</td>
<td>0.455 Kg</td>
<td>0.3 Kg</td>
<td>0.2 Kg</td>
<td>10 cm</td>
<td>11 cm</td>
<td>11 cm</td>
<td></td>
</tr>
<tr>
<td>$l_4$</td>
<td>$l_5$</td>
<td>$l_6$</td>
<td>$l_7$</td>
<td>$l_w$</td>
<td>$l_{fl}$</td>
<td>$l_{fw}$</td>
<td>$l_{f1}$</td>
<td>$l_{f2}$</td>
<td></td>
</tr>
<tr>
<td>10 cm</td>
<td>11 cm</td>
<td>11 cm</td>
<td>10 cm</td>
<td>11.2 cm</td>
<td>10 cm</td>
<td>8 cm</td>
<td>5 cm</td>
<td>4 cm</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. (a) Foot parameters, (b) sagittal plane, and (c) frontal plane configuration.

the mass of link $i$ is uniformly distributed about the CoM [11, 25].

3. Impact Model and Velocity Discontinuity

This section describes the effects in terminal velocities in the foot–ground impact. This condition occurs when the swing leg touches the walking surface. Let $Q_s$ be the $N$-dimensional configuration space of the robot when the stance leg end is acting as a pivot and let $q_s = (q_1, ..., q_N) \in Q_s$ be a set of generalized coordinates. The swing phase model corresponds to an open kinematic chain. Applying the method of Lagrange (see [26]), the model is written in the form:

$$D_s(q_s)\ddot{q}_s + C_s(q_s, \dot{q}_s)\dot{q}_s + G_s(q_s) = B_s(q_s)u$$ (2)

The matrix $D_s$ being the inertia matrix, $C_s$ being the Coriolis matrix, $G_s$ representing the gravity vector, and $B_s$ mapping the joint torques to generalized forces. Expression $u = (u_1, ..., u_N) \in \mathbb{R}^N$ holds the torque being applied to each joint $i$ ($i \in (1, ..., N)$). It is clear so far that not all configurations of the model are physically compatible to the single support phase concept of walking as presented before. For example, all points of the robot should be above the walking surface excluding the end of the stance leg of course. In addition, there are some other kinematic constraints [27] that must also be considered.

The impact event is very short [28], so the ground reaction forces may be replaced by an impulse. The impact model therefore involves the reaction forces at the leg’s end and requires the model of the biped robot [18, 26]. Let $q_s$ be the generalized coordinates for the single support model and $c = (c_x, c_y, c_z)$ the Cartesian coordinates of some fixed point mass on the robot. By using the generalized coordinates $q_e = (q_s, c)$, the Lagrange method yields:

$$D_e(q_e)\ddot{q}_e + C_e(q_e, \dot{q}_e)\dot{q}_e + G_e(q_e) = B_e(q_e)u + \delta F_R$$ (3)

with $\delta F_R$ representing the reaction force acting on the robot due to the contact between the swing leg’s end and the ground. According to [18], that results in a discontinuity for the velocity components of the biped robot. This will therefore be a new initial condition from which the single support model would evolve until the next impact appears as follows:

$$\dot{q}_e^+ = \Delta(\dot{q}_e^-)$$ (4)

with $\dot{q}_e^+$ and $\dot{q}_e^-$ being the velocities values just after and just before the impact, respectively. Considering no rebounds, the impact can thus be modelled as in [18] as follows:

$$D_e(q_e^-)\dot{q}_e^- - D_e(q_e^+)\dot{q}_e^+ = F_R$$ (5)

According to (5), the value of the reaction force $F_R$ (impact) decreases when the difference among $\dot{q}_e^+$ and $\dot{q}_e^-$ is minimized. Such condition may be induced, according
to \((4)\), if the trajectory speed \(\dot{q}_c^-\) is minimized before impacting the ground \([18]\).

4. Polynomial Trajectory Algorithm

The position of the robot is controlled with respect to the frontal plane by motors M1, M6, M5, and M10 (See Fig. 1(a)). The walking sequence of a biped robot can thus be determined by computing the hip and swing foot trajectories in the sagittal and frontal plane \([29, 30]\). For the sagittal case, the servo control system drives motors M7, M8, and M9 for the left leg and M2, M3, and M4 for the right leg (Fig. 1(b)). In this work, the robot’s stability was achieved by applying the ZMP criteria, while HS interpolation is used to generate the walking trajectories as explained below.

4.1 Trajectory Generation

One walking motion can be considered as a repetition of one-step motion being repeated within a \(T_s\) period with the initial state representing the standing home position. To achieve robust walking, the impact with the walking surface at the end of the single support phase should be executed smoothly by reaching zero velocity at the very contact spot. The walking sequence can thus be determined by only computing the trajectory of the hip and the swing foot and using the inverse kinematics to generate the trajectory for each independent joint in the biped structure.

4.2 Hip Trajectory in Sagittal Plane

Hip trajectory can be generated using the HS polynomial algorithm, considering that the initial and final states (position and velocities) are known from the single support phase. Figure 3 shows the initial position defined by \([x_{hs}, z_{hs}]\) and the final position represented by \([x_{he}, z_{he}]\). The initial velocity \([v_{xhs}, v_{zhs}]\) (produced when the robot leaves the initial position) is also specified in the trajectory model. The same applies for the final velocity \([v_{xhe}, v_{zhe}]\), exhibited when the robot contacts the walking surface. The initial and final positions for the cubic trajectory in \(z\) (the \(z_h(t)\) direction) can be expressed as follows:

\[
x_h(t) = \begin{cases} 
  x_{hs} & t = kT \\
  x_{hs} + v_{xhs}(t-kT) & \frac{kT}{2} < t \leq kT + T_s \\
  x_{hs} + v_{xhs}(t-kT) + \frac{1}{2}a_0(t-kT)^2 & kT < t \leq kT + T_s 
\end{cases}
\]

\[
x_h(t) = \frac{(3)(x_{he} - x_{hs}) - 2v_{zhs}(T_p - T_1)}{(T_p - T_1)^3} \quad kT < t \leq kT + T_s
\]

The same applies for the final velocity \([v_{xhe}, v_{zhe}]\), exhibited when the robot contacts the walking surface. The initial and final positions for the cubic trajectory in \(z\) (the \(z_h(t)\) direction) can be expressed as follows:

\[
z_h(t) = \begin{cases} 
  z_{hs} & t = kT \\
  z_{hs} + v_{zhs}(t-kT) & \frac{kT}{2} < t \leq kT + T_s \\
  z_{hs} + v_{zhs}(t-kT) + \frac{1}{2}a_0(t-kT)^2 & kT < t \leq kT + T_s 
\end{cases}
\]

\[
z_h(t) = \frac{(3)(z_{he} - z_{hs}) - 2v_{zhs}(T_p - T_1)}{(T_p - T_1)^3} \quad kT < t \leq kT + T_s
\]
4.3 Swing Foot Trajectory in Sagittal Plane

HS polynomial interpolation is used to generate the foot trajectory in the single support phase. To assure a smooth transition, velocities \((v_{xhs}, v_{zhs})\) and \((v_{xfe}, v_{zfe})\) should be defined near zero. The final foot position (Fig. 3) representing target positions and velocities can thus be obtained following:

\[
x_f(t) = \begin{cases} 
  x_f(t) = x_{fs} & t = kT \\
  x_f(t) = x_{fe} & t = kT + T_s \\
  x_f(t) = 0 & t = kT + T_s 
\end{cases} \\
\]

\[
z_f(t) = \begin{cases} 
  z_f(t) = z_{fs} & t = kT \\
  z_f(t) = z_{fe} & t = kT + T_s \\
  z_f(t) = 0 & t = kT + T_s 
\end{cases}
\]

(10)

From the initial and the final positions in \(x\)- and \(z\)-axis, a smooth trajectory can be generated by the HS interpolation yielding for \(x_f(t)\) and \(z_f(t)\):

\[
x_f(t) = x_{fs} + 3(x_{fs} - x_{fe}) \frac{(t-kT)^2}{T_s^2} - 2(x_{fs} - x_{fe}) \frac{(t-kT)^5}{T_s^5} \\
\]

\[
z_f(t) = \begin{cases} 
  z_f(t) = z_{fs} + 3(z_{fs} - z_{fe}) \frac{(t-kT)^2}{T_m^2} - 2(z_{fs} - z_{fe}) \frac{(t-kT)^3}{T_m^3} & kT < t < kT + T_m \\
  z_f(t) = z_{fe} + 3(z_{fe} - z_{fm}) \frac{(t-kT-T_m)^2}{T_m^2} - 2(z_{fe} - z_{fm}) \frac{(t-kT-T_m)^3}{T_m^3} & kT + T_m < t < kT + T_s 
\end{cases}
\]

(11)

The hip and knee positions required to produce appropriate movements for each leg can thus be calculated using the inverse kinematics of the robot’s structure.

4.4 Hip Trajectory in Frontal Plane

The hip trajectory can also be generated by the HS polynomial algorithm. However, all movements in the frontal plane should be performed within a cycle (see Fig. 4). The hip moves from the initial position \((y_{hs})\) to the maximum allowed displacement \((y_{he})\), with an initial velocity \((v_{yhs})\). Such movement must be completed within the half of the walking period \((T_s = T_v/2)\). From that position \((y_{he})\), the hip moves again to its initial position \((y_{hs})\), by executing a very smooth trajectory. It allows diminishing the speed while arriving to the final point by reducing the characteristic impact on the leg at ground contact. From initial and final positions in \(y\), a smooth trajectory can be generated by the HS interpolation yielding:

\[
y_h(t) = \begin{cases} 
  y_h + \frac{3(y_{hs} - y_{he})}{(T_s - T_v)}(t-kT)^2 - \frac{2(y_{hs} - y_{he})}{(T_s - T_v)^3}(t-kT)^3 & kT < t < T_2 \\
  y_{he} + \frac{3(y_{hs} - y_{he})}{(T_s - T_v)}(t-kT)^2 - \frac{2(y_{hs} - y_{he})}{(T_s - T_v)^3}(t-kT)^3 & T_2 < t < T_s 
\end{cases}
\]

(12)

4.5 Determination of the Algorithm Parameters

Parameters \(v_{xhs}, T_1, T_m, z_m,\) and \(v_{yhs}\) have a decisive impact on the trajectory smoothness with respect to velocity and acceleration. Considering that the link’s speed at the end of the single support phase should be close to zero to assure a smooth contact with the floor surface, an appropriate selection is relevant for the overall performance. If \(T_1 = T_m = T_v\) then only \(T_v\) needs to be fixed. If \(T_v\) is close to \(T_s/2\) then a robust walking trajectory with smooth impact transition is produced. In turn, the parameter-setting allows a better robustness-disturbances relationship either from non-modelled dynamics of the biped robot or from the walking surface.

5. Walking Motions

Motion can be thus generated according to the PTA algorithm following Equations (13)–(23). The trajectory for each point mass \((m_i)\) is shown in Fig. 5. A step period of period \(T_s = 2 s\) and step length \(L_s = 10 cm\) are considered. The evolution of the trajectories may be simulated considering the “Dany Walker” model according to Table 1. The test considers a flat walking surface. Figure 6 shows the trajectories for the ZMP as they are generated by PTA for real-time experiments in the robot. The values of the ZMP were calculated from a triangular arrangement of force sensors located at each foot of the “Dany Walker” robot [19, 21].
5.1 Experiments

To demonstrate the effectiveness of the PTA algorithm, the step motion is tested under the assumption that the contact between the swing foot and the ground happens on uneven terrain. Two experiments are prepared. First, the impact is simulated over a variable-height surface to test the PTA performance. The experiment includes a comparison between PTA and other trajectory generation methods. The second experiment is a real-time test of the resulting PTA’s trajectories. Small wooden pieces are arranged over the robot’s path. They are slightly bigger than the size of the robot’s foot, aiming to create a changing relationship between the swinging foot and the floor. To compare the results, the same experiment was proved with the broadly well-known approach (according to refs. [32–36] found in the literature) presented in [10].

5.1.1 Simulation of the Impact on the Floor

The first experiment employs the “Dany Walker” biped robot. The kinematic and dynamic models of the robot are presented in [20] and [19]. The impact model proposed in [18] is used to test the actual ability of PTA to achieve a stable walking within a variable-height terrain whose attitude varies from 0 to 1 cm at about 6 cm away from the starting point (see Figure 7). The simulation considers $T_s = 5\text{ s}$ and $L_s = 10\text{ cm}$. Figure 8 shows the trajectories produced by PTA and the zoom of the transitory produced over the mass 6, at the impact instant. After the simulation, it can be assumed that the speed of $m_7(|\dot{m}_7|)$ was 3.17 cm/s producing an equivalent force of $F_R = 0.11N$ at the impact.

Table 2 shows the performance of the PTA as it is compared to other IPM trajectory generation algorithms such as the Ohishi’s method [11], the Kajita’s algorithm [17], Park’s contribution in [10], and the Van der Pol oscillator as it is applied in [15]. The experiment employed the same parameters reported by the authors to ease the comparison to the “Dany Walker” robot.

The PTA reduces the speed and therefore produces a small reaction force $F_R$ at the impact (near to zero), compared to other algorithms based on IPM or on Van der Pol oscillators which hold the highest velocity at the impact with 13.71 cm/s. The IPM algorithms adjust the trajectories according to accelerations yielding higher velocities before the impact while the Van der Pol-based algorithm keeps constant speeds even when the surface’s height is increased. The value of the reaction force $F_R$ (impact) decreases when the velocity $\dot{m}_7$ is minimized.

5.1.2 Impacts on the Real-Time Walking

This experiment aims to set a new walking surface by scattering wooden pieces of same dimensions of the biped’s foot but being 2 cm taller than them at each side. Obstacles begin at about 15 cm away from the initial position and they end at about 50 cm at the robot’s front flank. Figure 8(a) shows the results of PTA, while Fig. 8(b) presents the performance of the IPM Park algorithm [10] over the same experimental setting. The ZMP trajectory presented in Fig. 8(a) demonstrates that the ZMP is always kept
Figure 7. Walking PTA trajectory in the sagittal plane under variable terrain conditions. (Upper right) A zoom view of the transient response of the mass \(m_6\).

Figure 8. Second experiment. The ZMP trajectories are shown while the robot walks over an uneven terrain. (a) PTA operation and (b) IPM Park’s algorithm.

Table 2
Results of the First Experiment: PTA versus Other Trajectory Generation Algorithms under Changing Terrain Conditions

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Velocity of (m_7) ((\dot{m}_7)) (cm/s)</th>
<th>(F_R) (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTA</td>
<td>3.17</td>
<td>0.11</td>
</tr>
<tr>
<td>IPM Ohishi version</td>
<td>9.51</td>
<td>0.821</td>
</tr>
<tr>
<td>IPM Kajita version</td>
<td>10.44</td>
<td>0.902</td>
</tr>
<tr>
<td>IPM Park version</td>
<td>8.82</td>
<td>0.711</td>
</tr>
<tr>
<td>Van der Pol oscillator</td>
<td>13.71</td>
<td>1.05</td>
</tr>
</tbody>
</table>

Table 3
Results of the Second Experiment: PTA versus IPM Park Algorithms under Changing Terrain Conditions

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>(F_R) (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPM Park version</td>
<td>0.717</td>
</tr>
<tr>
<td>PTA</td>
<td>0.201</td>
</tr>
</tbody>
</table>

To measure the reaction force, a force–voltage relationship [37] is used. Only left foot impacts have been considered.

5.2 Computational Cost

This section compares the computational cost of PTA to other generation algorithms such as Hermitian polynomials and B-Splines. The time required to calculate the solution and the required storage space are considered as comparison indexes. All systems are compiled using the CCS PIC-C®4.068 compiler aiming for the PIC18F4550 microcontroller platform.

The experiment considers a complete step trajectory and its computation. Table 4 shows results considering several methods such as PTA, IPM Ohishi method [11], IPM Kajita algorithm [17], IPM Park procedure [10], and
Table 4
PTA Computational Cost versus Other Trajectory Generation Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Calculus Time (s)</th>
<th>Storage Space (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTA</td>
<td>0.4</td>
<td>86</td>
</tr>
<tr>
<td>IPM Ohishi version</td>
<td>4</td>
<td>415</td>
</tr>
<tr>
<td>IPM Kajita version</td>
<td>4</td>
<td>415</td>
</tr>
<tr>
<td>IPM Park version 2</td>
<td>211</td>
<td></td>
</tr>
<tr>
<td>Van der Pol oscillator</td>
<td>12</td>
<td>518</td>
</tr>
</tbody>
</table>

the Van der Pol oscillator algorithm [15].

Table 3 shows that only the IPM Park version can actually generate acceptable gaits, considering the modest computer resources of the robot. Other methods cannot generate speeds smaller than 4s which poses a serious difficulty to perform control on the system by considering the signal sensors feedback and the calculus of the ZMP. Besides, to implement the IPM-based algorithms (Ohishi, and Kajita) and the Van der Pol oscillator, it was necessary to expand the memory of the host computer system because one numerical method is required to solve the pendulum-based system for IPM methods or to generate the oscillation trajectories for the Van der Pol method. The B-splines class is a well-known method which is comparable to the Hermitian polynomial models. They generate trajectories by simply approximating control points (positions and speeds). However, the B-spline allows a better control of the intermediate points within the trajectory, as they can be modified locally with no substantial change in the overall trajectory. In the case of biped-robots trajectory generation, the intermediate points are commonly not modified as the most important target is the trajectory’s ending point. Therefore, the use of B-splines do not represent any advantage in comparison to Hermitian polynomials, nonetheless its use implies a bigger computational cost and the use of more complex recursive computations [38].

A detailed comparative analysis can be found in [39].

6. Conclusions

In this paper, the PTA has been proposed as a simple algorithm to generate walking trajectories for biped robots. PTA employs equations from the interpolation of initial and final conditions of HS polynomials as trajectories. The algorithm is capable of generating smooth impacts and reducing the falling trend despite un-modelled dynamic conditions and changes on the walking terrain. To test the overall performance, several joint trajectories for walking motion have been tested over the bipedal robot “Dany walker,” drawing comparisons to other IPM methods. The results show that impact effects can be minimized by PTA. The paper also analyses the resulting ZMP trajectory after applying PTA and other similar methods. Results confirm that PTA is a simple and suitable method to generate appropriate trajectories for bipedal walking in real time despite the use of modest computing resources such as commercial embedded microcontrollers. Considering that PTA requires only (6) to (12) to be fully functional, it only demands modest computing resources easing its application to other humanoid robotic platforms.

Acknowledgement

The work presented in this paper is sponsored by CONACYT under the project “Ciencia Básica 82877.”

References


[33] S. Bououden, F. Abdessamed, & B. Abderraouf, Control of a bipedal walking robot using a fuzzy precompensator, in A. Häkansson et al. (Eds.), Agent and multi-agent systems: Technologies and applications (Berlin: Springer-Verlag, 2009), 1344–1370.


Biographies

Erik Cuevas received his B.S. degree with distinction in Electronics and Communications Engineering from the University of Guadalajara, Mexico, in 1995, the M.Sc. degree in Industrial Electronics from ITESO, Mexico, in 2000, and the Ph.D. degree from Freie Universität Berlin, Germany, in 2005. Since 2001, he was awarded a scholarship from the German Service for Academic Interchange (DAAD) as full-time researcher. From 2006 he has been with University of Guadalajara, where he is currently a full-time professor in the Department of Computer Science. From 2008 he is also a member of the Mexican National Research System (SNI). His research interest includes computer vision and artificial intelligence applications in medical image.

Daniel Zaldivar received his B.S. degree with distinction in Electronics and Communications Engineering from the University of Guadalajara, Mexico, in 1995, the M.Sc. degree in Industrial Electronics from ITESO, Mexico, in 2000, and the Ph.D. degree from Freie Universität Berlin, Germany, in 2005. From 2001 he was awarded a scholarship from the German Service for Academic Interchange (DAAD) as full-time researcher. From 2006 he has been with University of Guadalajara, where he is currently a professor in the Department of Computer Science. From 2008 he is also a member of the Mexican National Research System (SNI). His current research interest includes biped robots design, humanoid walking control, and artificial vision.

Marco Pérez-Cisneros received his B.S. degree with distinction in Electronics and Communications Engineering from the University of Guadalajara, Mexico, in 1995, the M.Sc. degree in Industrial Electronics from ITESO University, Mexico, in 2000, and the Ph.D. degree from the Control Systems Centre, UMIST, Manchester, UK, in 2004. From 2005 he has been with University of Guadalajara,
where he is currently a professor and head of Department of Computer Science. From 2007 he has spent yearly spells at the University of Manchester as an invited honorary professor. He is a member of the Mexican National Research System (SNI) since 2007. His current research interest includes robotics and computer vision in particular visual servoing applications on humanoid walking control.

*Marte Ramírez-Ortegón* is a Ph.D. student at the Department of Mathematics and Computer Science of the Freie Universität Berlin. In 2002, he received a B.S. degree (Computer Science) from the University of Guanajuato (UG)/the Centre for Mathematical Research (CIMAT), Guanajuato, Mexico (1998–2002). His current research fields include pattern recognition, edge detection, binarization, multithresholding, and text recognition.