Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks

Christophe Sabourin, Olivier Bruneau

Vision and Robotics Laboratory, High National School of engineering of Bourges, 10 Boulevard Lahitolle 18020 Bourges Cedex France

Abstract

In this paper, we propose a control strategy allowing us to perform the dynamic walking gait of a virtual under-actuated robot even subjected to destabilizing external disturbances. This control strategy is based on two stages. The first one consists in using a set of pragmatic rules in order to generate a succession of passive and active phases allowing us to perform the dynamic walking of the robot. This first stage is used as a reference control to learn, by using neural networks, a set of articular trajectories. In the second stage, we use these neural networks to generate the learned trajectories during the first stage. The goal of the use of these neural networks is to increase the robustness of the control of the dynamic walking gait of this robot in the case of external disturbances.

Key words: Under-actuated biped robot; Dynamic walking gait; CMAC neural networks; Robustness.

1 Introduction

The design and the control of bipedal robots are one of the more challenging topics in the field of robotics and were treated by a great number of researchers and engineers since many years. The potential applications of this research are very important in the medium and long term. Indeed this can lead initially to a better comprehension of the human locomotion, what can be very helpful for the design of more efficient orthosis. Secondly, the bipedal humanoid robots are intended to replace the human for interventions in hostile environments or to
help him in the daily tasks. In addition to the problems related to autonomy and decision of such humanoid robots, the essential locomotion task is still today a big challenge. If it is true that some prototypes were constructed and prove the feasibility of such robots, of which most remarkable is undoubtedly today the robots Asimo [1] and HRP-2P [2], the performances of these are still far from equalizing the human from the point of view of the dynamic locomotion process. The design of new control laws making it possible to ensure real time control for real dynamic walking in unknown environments is thus today fundamental. Most of the time, the control of bipedal walking robots is carried out with methods based on tracking of temporal reference trajectories [3] [4] [5] [6] [7] [8] generally associated with a control of the Zero Moment Point (ZMP) [9]. The torque applied at each joint is thus computed by using the difference between the desired and real trajectories.

However, this kind of approaches involves some difficulties with regard to the autonomy of the robot:

• Firstly, it is necessary to be able to modify on-line, for instance, the desired average velocity. If the trajectories are computed on-line, many calculations are necessary and require powerful data-processing tools in order to reduce the computing time. And if the trajectories are calculated off-line, it is necessary to memorize a set of joint trajectories for each velocity of each gait and for each transition between two different tasks.
• Secondly, the tracking of reference trajectories do not ensure the global stability of the robot, especially when the ground is not regular or when a big perturbation force is applied to the robot. Indeed, in the first case the calculation of a trajectory requires to predict the position of the impact of the foot with the ground. From this constraint, which can not easily be carried out on an irregular ground, it results that a difference between the desired and real impact point can occur. The actuator torques, generally computed by using a PD control, can then become very big and involve an instability, even the fall of the robot. In the second case, an external perturbation force can destabilize the robot because the movements are not well coordinated to react in the right way to maintain the dynamic stability of the robot. Indeed, the foot-ground contact being unilateral, it can occur one uncontrolled kinetic momentum of the system involving a rotation of the foot with regard to the ground surface. In this case, the tracking of temporal reference trajectories does not allow any more synchronization of the leg movements.
• The criterion of stability ZMP, considered as dynamic criterion of stability is restrictive with regard to the irregularity of the ground and to the speed of the movements. Moreover, the use of the ZMP implies that the robot has feet. It is due mainly to the fact that the dynamic stability from the point of view of the ZMP is related to the width of the support base. For instance, if the support base is lower than the foot surface (crossing of an
obstacle), this one is not valid. Moreover, this criterion is also inoperative in the phases of flight during the run.

Globally speaking, the control techniques using the tracking of temporal reference trajectories associated with an on-line control of ZMP are today limited on the one hand with regard to the complexity of calculations required to generate full dynamical motions and on the other hand with regard to the structure of robots (unusable for robots without feet). In fact, in order to get an efficient walking gait for biped robots, we think that it is necessary to solve four main problems:

- First of all, we have to maintain the dynamic stability of the robot. This can be performed by using an instantaneous criterion, the Foot Rotation Indicator (FRI) [10], the Virtual Generalized Stabilizer (VGS) [11] or by considering the stability in the sense of the limit cycles [12] [13].
- Secondly, in order to improve the autonomy of a robot, we have to minimize the energy cost. In order to carry out this, we can use the intrinsic dynamics of the robot and minimize the energy cost of the actuators [14] [15].
- Thirdly, we have to decrease the computing time by using simple and discrete control strategies [16] [17].
- Finally, it is essential to be able to adapt the control strategies and to modify the walking gait with regard to the environment (characteristics and geometry of the ground and of obstacles, external forces) and the state of the robot (carried load, level of energy). In this case, one solution is to use learning methods [18] [19] [20].

Commonly, these four goals are separately treated. Our aim is to propose a method in which these four points are taken into account at the same time.

In this article, we present a new control strategy within the framework of the control of an under-actuated robot: RABBIT [21] [22]. The main characteristic of this robot is that it does not have a foot and consequently that its walking gait is completely dynamic. In this case, the stability criterion ZMP cannot be used.

Our strategy of control is based on two stages:

- The first one uses a set of pragmatic rules making it possible to stabilize the pitch angle of the trunk and to generate the leg movements [23]. This simple control strategy, in the absence of disturbances, allows to generate on the one hand a stable dynamic walking and on the other hand velocity transitions and transitions of initiation and stop of walking gait. But the use of passive phases implies that this control strategy is very sensitive to the external disturbances that could occur during the progression of the robot. Moreover, frictions applied on each joint can be important and considerably limit the dynamics of the passive phases. Consequently, we
propose to use a neural network allowing on the one hand to increase the robustness of our control strategy and on the other hand to compensate articular frictions on a real robot. In fact, in the first stage, the pragmatic rules are used as a reference control to learn, by using Cerebellar Model Controller Articulation (CMAC) [24] [25], a set of articular trajectories. During this training phase, we consider that the robot moves in an ideal environment (without disturbance) and that frictions are negligible.

• In the second stage, we use these neural networks to generate the trajectories learned during the first stage. A PD control allows us to ensure the tracking of this reference trajectories. But in our case, the trajectories learned by the neural networks are not a function of time but a function of the geometrical pattern of the robot. This implies that the movements of the legs are always coordinated and this, even in the presence of disturbances (external force, sliding, irregular ground). The robustness of the control of the walking robot is considerably improved what reduces the possibilities of robot fall.

This paper is organized as follows. Section 2 presents the characteristics of the real and virtual under-actuated robot. In Section 3, the control strategies using the pragmatic rules to produce a stable walking gait and taking into account the servo-motors limitations are described. Section 4 presents the method using the neural networks. In section 5, in order to evaluate the robustness of this control strategy, we present results when the progression of the robot is disturbed. Conclusions and further developments are finally given.

2 Model of the bipedal walking robot

A lot of advanced humanoid bipedal robots such as Asimo robot and HRP2 robot, are not able to produce walking gaits with high velocity or running gaits with high dynamics. In addition to the problem of actuators, this is namely due to the fact that the stability of walking is ensured by keeping the ZMP strictly inside the basis of support. If it is true that, in its design, our robot RABBIT, shown by the figure 1, is simpler compared to a robot with feet, from another point of view, it is more difficult to control, particularly because, in phase of single support, the robot is under-actuated. This implies to control one passive degree of freedom indirectly with the actuated degree of freedom. In this case, the analysis and the control of the intrinsic dynamics of walking (gravity, inertial effects) are of primary importance. For instance, the control of the pitch angle of the trunk permits to replace the moment generally applied to the ankle in the case of robot with actuated ankles. Consequently, the study of robot without feet can lead to an improvement of the current performances of the biped robots by the design of new control laws.
2.1 Prototype RABBIT

The robot RABBIT is an experimentation of seven french laboratories (IR-CCYN Nantes, LAG Grenoble, LMS Poitiers, LVR Bourges, LGIPM Metz, LRV Versailles, LIRMM Montpellier) concerning the control of a biped robot for walking and running within the framework of a CNRS project ROBEA [26]. This robot is composed of two legs and a trunk and has no foot as shown on figure 1.

The joints are located at the hip and at the knee. This robot has four actuators: one for each knee, one for each hip. The characteristics (masses and lengths of the limbs) of RABBIT are summarized in table 1. Motions are included in the sagittal plane by using a radial bar link fixed at a central column that allows to guide the direction of progression of the robot around a circle. This robot represents the minimal system able to generate walking and running gaits. Since the contact between the robot and the ground is just one point (passive dof), the robot is under-actuated during the single support phase: there are only two actuators (at the knee and at the hip of the contacting leg) to control three parameters (vertical and horizontal position of the platform and pitch angle).

<table>
<thead>
<tr>
<th>Limb</th>
<th>Weight (Kg)</th>
<th>Length (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trunk</td>
<td>20</td>
<td>0.625</td>
</tr>
<tr>
<td>Thigh</td>
<td>6.8</td>
<td>0.4</td>
</tr>
<tr>
<td>Shin</td>
<td>3.2</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Table 1
MASSES AND LENGTHS OF THE LIMBS OF THE ROBOT
Fig. 2. Modeling of the robot with Adams.

2.2 Virtual Prototype

The numerical model of the robot previously described was designed with the software ADAMS\(^1\). This software, from the modeling of the mechanical system (masses and geometry of the segments) is able to simulate the dynamic behavior of this system and namely to calculate the absolute motions of the platform and the relative motions of the limbs when torques are applied on the joints by the virtual actuators (Fig. 2).

The model used to simulate the interaction between feet and ground is exposed in [27]. The normal contact force is given by equation (1):

\[
F^n_c = \begin{cases} 
0 & \text{if } y > 0 \\
-\lambda^n c |y| \dot{y} + k^n c |y| & \text{if } y \leq 0
\end{cases}
\]  

(1)

Where \(y\) and \(\dot{y}\) are respectively the position and the velocity of the foot (limited to a point) with regard to the ground. \(k^n c\) and \(\lambda^n c\) are respectively the generalized stiffness and damping of the normal forces. They are chosen in order to avoid the rebound and to limit the penetration of the foot in the ground. The tangential contact forces are computed with the equation (2) in the case of a contact without sliding or with the equation (3) if sliding occurs.

\[
F^t_c = \begin{cases} 
0 & \text{if } y > 0 \\
-\lambda^t c \dot{x} + k^t c (x - x_c) & \text{if } y \leq 0
\end{cases}
\]  

(2)

\[
F^t_c = \begin{cases} 
0 & \text{if } y > 0 \\
-(\text{sgn}(\dot{x})) \lambda_g F^n_c - \mu_g \dot{x} & \text{if } y \leq 0
\end{cases}
\]  

(3)

\(^1\) ADAMS is a product of MSC software
Where $x$ and $\dot{x}$ are respectively the position and the velocity of the foot with regard to the position of the contact point $x_c$ at the instant of impact with ground. $k_t^c$ and $\lambda_t^c$ are respectively the generalized stiffness and damping of the tangential forces. $\lambda_g$ is the coefficient of dynamic friction depending on the nature of surfaces in contact and $\mu_g$ a viscous damping coefficient during sliding. After each iteration, the normal and tangential forces are computed from the equations (1) and (2). But, if $F_t^c$ is located outside the cone of friction ($\|F_t^c\| > \mu_s \|F_n^c\|$ with $\mu_s$ the static friction coefficient), then the tangential force of contact is computed with equation (3). The interest of this model is that it is possible to simulate walking with or without phases of sliding allowing us to evaluate the robustness of the control.

2.3 Actuator limits

Within the framework of the control of a real robot, the morphological description of this one is insufficient. It is thus necessary to take into account the technological limits of the actuators in order to implement the control laws used in simulation on the experimental prototype. Figure 3 gives the fields of application of servo-motor RS420J used for RABBIT. This one shows that it is necessary to limit the torque according to velocity. From these characteristics we thus choose to apply the following limitations:

- when velocity is included in $[0; 2000] \text{rpm}$, the torque applied to each actuator is limited to $1.5 Nm$ what corresponds to a torque of $75 Nm$ at the output of the reducer (ration gear equal to 50),
- when velocity is included in $[2000; 4000] \text{rpm}$ the power of each actuator is limited to $315 W$,
- when the velocity is bigger than $4000 \text{rpm}$, the torque is imposed to be equal to zero.

![Fig. 3. Field of application of the RS420J (continuous line) and chosen limits for the applied control torques (dotted line).](image)
The control strategy presented in this section is based on three points:

- the observation of the relations between joint movements and the evolution of the parameters describing the motions of the robot platform,
- an interpretation of the muscular behavior,
- the analysis of the intrinsic dynamics of a biped.

Based on these considerations, it is possible to determine a set of pragmatic rules. The objective of this strategy is to generate the movements of the legs by using a succession of passive and active phases. Figure 4 shows the reference angles required for the development of our control and the subsections 3.1 and 3.2 explain in details the laws used for the calculation of the actuator torques. Subsection 3.3 presents a high level control which allows to control the average velocity by adjusting the pitch angle of the trunk.

### 3.1 Control at the hip

During the swing phase, the torque applied to the hip given by equation (4) is just an impulse with a varying amplitude and a fixed duration equal to $(t_2 - t_1)$.

\[
T_{\text{sw}}^{\text{hip}} = \left\{ \begin{array}{ll}
K_{\text{hip}}^{\text{pulse}} & \text{if } t_1 < t < t_2 \\
0 & \text{otherwise}
\end{array} \right.
\]  

(4)
Where $K_{\text{hip}}^{\text{pulse}}$ is the amplitude of the torque applied to the hip at the beginning of the swing phase, and $t_1$ and $t_2$ are respectively the beginning and the end of actuation $K_{\text{hip}}^{\text{pulse}}$. It should be noted that this impulse is carried out by using a polynomial interpolation of the third order during a very short time $(t_2 - t_1 = 50\,\text{ms})$ and that the amplitude of $K_{\text{hip}}^{\text{pulse}}$ is included in $[0\,\text{Nm}, 75\,\text{Nm}]$. The duration of this impulse for all the simulations presented in this paper will be always equal to $50\,\text{ms}$. However, this duration can be modified if needed.

After this impulse, the hip joint is passive until the blocking of the swing leg in a desired position by using a PD control given by the equation (5), what allows to ensure a regular step length.

$$T_{\text{hip}}^{\text{sw}2} = K_{\text{hip}}^{p}(q_{\text{r}1}^{\text{dsw}} - q_{\text{r}1}) - K_{\text{hip}}^{w}q_{\text{r}1} \quad \text{if} \quad q_{\text{r}1} > q_{\text{r}1}^{\text{dsw}}$$

$q_{\text{r}1}$ and $q_{\text{r}1}^{\text{dsw}}$ are respectively the measured and desired relative angles between the two thighs (figure 4), $q_{\text{r}1}$ is the absolute angular velocity of the hip of the leg $i$. During the stance phase, the torque applied to the hip, given by the equation (6), is used to ensure the stability of the trunk.

$$T_{\text{hip}}^{\text{st}} = K_{\text{trunk}}^{p}(q_{\text{0}0}^{d} - q_{\text{0}0}) - K_{\text{trunk}}^{w}q_{\text{0}0}$$

where $q_{\text{0}0}$ and $q_{\text{0}0}^{d}$ are respectively the angle and the angular velocity of the trunk and $q_{\text{0}0}^{d}$ corresponds to the desired pitch angle of the trunk (figure 4).

### 3.2 Control at the knee

During the swing phase, the knee joint is free and the torque is equal to zero. At the end of the knee extension, a control torque, given by the equation (7) is applied to lock this joint in a desired position $q_{\text{i}2}^{\text{dsw}}$. This avoids on one hand an inverse knee flexion and on the other hand the rebound after a possible knee extension which is not the case by using a simple block stop.

$$T_{\text{knee}}^{\text{sw}} = K_{\text{knee}}^{p}(q_{\text{i}2}^{\text{dsw}} - q_{\text{i}2}) - K_{\text{knee}}^{w}q_{\text{i}2}$$

$q_{\text{i}2}$ and $q_{\text{i}2}^{d}$ are respectively the measured angular position and angular velocity of the knee joint of the leg $i$. $q_{\text{i}2}^{\text{dsw}}$ is a constant value included in $[-20^\circ; 0^\circ]$ which depends on the desired propulsion during the next stance phase of this leg. During the stance phase, the torque is computed by using equation (8).

$$T_{\text{knee}}^{\text{st}} = K_{\text{knee}}^{p}(q_{\text{i}2}^{\text{dest}} - q_{\text{i}2}) - K_{\text{knee}}^{w}q_{\text{i}2}$$

We choose $q_{\text{i}2}^{\text{dest}} = 0$ at the impact with the ground in equation (8) what contributes to propel the robot if $q_{\text{i}2}^{\text{dest}} > q_{\text{i}2}^{\text{dsw}}$. Indeed, the change of the desired
position of the knee joint implies that the control law generates an impulse torque which physically acts like a spring-damper and which namely depends on \((q_{d_{2}}^{\text{dst}} - q_{2}^{\text{dsw}})\). Indeed, just before the impact, \(q_{2}^{d}\) is equal to the theoretical value \(q_{2}^{\text{dsw}}\). During the continuation of the stance phase, the same control law is used to lock the knee at the position \(q_{2}^{\text{dst}} = 0\).

### 3.3 High level velocity control

The strategy previously presented allows us to produce the dynamic walking gait of the bipedal robot without reference trajectories and with an average velocity included in \([0 \text{m/s}; 1 \text{m/s}]\). The figure 5 shows, for a set of fixed parameters \((q_{0}^{d} = 20^\circ, q_{2}^{\text{dsw}} = -10^\circ, q_{r_{1}}^{\text{dsw}} = 42^\circ, K_{\text{hip}}^{\text{pulse}} = 60\text{Nm})\) the stick diagram of the obtained stable dynamic walking gait.

![Fig. 5. Stick diagram of stable dynamic walking gait.](image)

The major problem is that it is necessary to regulate manually, for each desired average velocity, the four mentioned parameters: \(q_{0}^{d}, q_{2}^{\text{dsw}}, q_{r_{1}}^{\text{dsw}}, K_{\text{hip}}^{\text{pulse}}\). In order to solve this problem, we propose to control the average velocity by adjusting the pitch angle of the trunk at each step by using the error between the average velocity \(V_{M}\) and the desired average velocity \(V_{M}^{d}\) and of its derivative as described in figure 6.

![Fig. 6. Synoptic of the high level control for the average velocity.](image)

\(V_{M}\) is simply calculated by using (9) where \(L_{\text{step}}\) is the distance between the two feet at the moment of double impact and \(t_{\text{step}}\) is the duration of the step.
(from takeoff to landing of the same leg).

\[ V_M = \frac{L_{step}}{t_{step}} \]  \hspace{1cm} (9)

At each step, \( \Delta q_{0}^{d} \), which is computed by using the error between \( V_M \) and \( V_M^{d} \) and of its derivative (see equation (10)), is then added to the pitch angle of the previous step \( q_{0}^{d}(n) \) in order to obtain the new desired pitch angle of the following step \( q_{0}^{d}(n + 1) \) as shown in equation (11).

\[ \Delta q_{0}^{d} = K^p (V_M^{d} - V_M) + K^v \frac{d}{dt} (V_M^{d} - V_M) \]  \hspace{1cm} (10)

\[ q_{0}^{d}(n + 1) = q_{0}^{d}(n) + \Delta q_{0}^{d} \]  \hspace{1cm} (11)

Figure 7 shows the evolution of the average velocity when the desired velocity changes. It must be pointed out that the used solution really allows to regulate the average velocity. We can observe that, after each modification of the desired velocity, the real velocity converges towards the desired velocity after around ten steps. The adjustment of the proportional \( K^p \) and derived \( K^v \) parameters are carried out in order to obtain a smooth transition velocity.

Fig. 7. Evolution of the average desired and real velocities when the pitch angle of the trunk is controlled.

Figures 8 shows the torque of the motors at the hip and at the knee during a complete cycle of walking (swing and stance phase). The swing phase is included between the stage one and four, and the stance phase between the stage four and eight. During, the active phases, the limitation of the torque and power are the constraints given in subsection 2.3.

At the stage one, the torque at the hip is just an impulse. Between the stages one and two, the torque at the hip is equal to zero. Between the stages one and three, the torque at the knee is equal to zero too. At the stage two, the hip is locked by using a PD control. And at the stage three, this is the knee which
is locked. It must be pointed that the locking of the knee implies a reaction at the hip. At the stage four, there is impact with the ground. It is the end of the swing phase. After this impact, the trunk is destabilized and the torque at the hip is very large. Moreover, at this instant, an impulse torque is applied at the knee in order to propel the robot forward. During the stance phase, the dynamic coupling implies reaction forces on the stance leg due to motion of the swing leg; stage five, $K_{\text{pulse}}$ of the swing leg; stage six, locking of the hip of the swing leg; stage seven, locking of the knee of the swing leg. At the stage eight, it is the end of the stance phase.

This control strategy allows us to produce the dynamic walking of the bipedal robot without reference trajectories and with an average velocity included in $[0 \text{ m/s}; 1 \text{ m/s}]$. The interest of this method resides on the fact that on the one hand the intrinsic dynamics of the system are exploited by using a succession of active and passive phases and on the other hand that the control strategy is very easy to implement on-line. But the use of passive phases implies that this control strategy is very sensitive to the external disturbances that could occur during the progression of the robot. Moreover, frictions of each articulation can be important on a real robot and consequently limit considerably the dynamics of the passive phases. Consequently, we propose to use a neural network allowing on the one hand to increase the robustness of our control strategy and on the other hand to compensate articular frictions on a real robot. The CMAC, the neural network that we have selected and its implementation are described in the following section.

4 Control Strategy based on CMAC

In this section, in the first step, the principle of the CMAC is presented. Sections 4.2 and 4.3 respectively describe the phase of training and the use of this neural network.
4.1 The CMAC neural network

The CMAC is a neural network imagined by Albus starting from the studies on the human cerebellum. Despite its biological relevance, its main interest is the reduction of the training and computing times in comparison to other neural networks [28]. This is of course a considerable advantage for real time control. Because of these characteristics, the CMAC is thus a neural network relatively well adapted for the control of complex systems with a lot of inputs and outputs and has already been the subject of some researches in the field of the control of biped robots [29] [30].

The CMAC is an associative memory type neural network which is a set of $N$ detectors regularly distributed on several $C$ layers. The receptive fields of these detectors are distributed on the totality of the limited range of the input signal. On each layer, the receptive fields are shifted of a quantification step $q$. Consequently, the width of the receptive field are not always equal. The number of detectors $N$ depends on the one hand of the width of the receptive fields and on the other hand of the quantification step $q$. When the value of the input signal is included in the receptive fields of a detector, this one is activated. For each value of the input signal, the number of activated detector is equal to the number of layers $C$ (parameter of generalization). Figure 9 shows a simplified organization of the receptive fields having 14 detectors distributed on 3 layers. Being given that there is an overlapping of the receptive fields, neighboring inputs will activate common detectors. Consequently, this neural network is able to carry out a generalization of the output calculation for inputs close to those presented during learning.

Fig. 9. Description of the simplified CMAC with 14 detectors distributed on 3 layers.

For each value of the input signal, the number of activated detector is equal to 3. $A = [0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0]$, $Y = A(u)^T W = w_3 + w_8 + w_{12}$

The output $Y$ of the CMAC is computed by using two mappings. The first mapping projects an input space point $u$ into a binary associative vector $A =$
Each element of $A$ is associated with one detector and $N$ is the number of detector. When one detector is activated, the element of $A$ corresponding of this detector is 1 otherwise it is equal to 0. The second mapping computes the output $Y$ of the network as a scalar product of the association vector $A$ and the weight vector $W = [w_1..w_N]$ (equation (12)).

$$Y = A(u)^TW$$  \hspace{1cm} (12)

### 4.2 Learning during a walking gait

During the learning phase, we use the strategy developed in section 3 to control the robot. Furthermore, the trajectories of the swing leg (in terms of joint positions and velocities) are learned with several "single-input/single-output" CMAC neural networks. Indeed, two CMAC are necessary to memorize the joint angles $q_{11}$ and $q_{21}$ and two other CMAC for angular velocities $\dot{q}_{11}$ and $\dot{q}_{21}$. Figure 10 shows the method used during this training phase.

![Fig. 10. Principle of the control strategy used during the training of one CMAC neural network.](image)

When leg 1 is in support ($q_{12} = 0$), the angle $q_{11}$ is applied to the input of each CMAC ($u = q_{11}$) and when leg 2 is in support ($q_{22} = 0$), this is the angle $q_{21}$ which is applied to the input of each CMAC ($u = q_{21}$). Consequently, the trajectories learned by the neural networks are not function of time but function of the geometrical pattern of the robot. The weights $W$ of each CMAC $k$ ($k = 1, .., 4$) are given by equation (13).

$$w_k^i = w_k^{i-1} + \frac{\beta_k e_k}{C_k}$$  \hspace{1cm} (13)

$w_k^{i-1}$ and $w_k^i$ are respectively the weights before and after the training at each sample time $t_i$. $C_k$ is the generalization number of each CMAC and $\beta_k$ is a
parameter included in \([0, 1]\). \(e_k\) is the error between the desired output \(Y_k^d\) of the CMAC \(k\) and the computed output \(Y_k\) of the CMAC. When the stance leg is leg number one, \(Y^d = [Y_1^d, Y_2^d, Y_3^d, Y_4^d] = [q_{11}, q_{12}, q_{21}, q_{22}]\) and when the stance leg is leg number two, \(Y^d = [Y_1^d, Y_2^d, Y_3^d, Y_4^d] = [q_{11}, q_{12}, q_{21}, q_{22}]\). We have to mention that \(Y^d\) are at the same time the output results of the simulation of the robot and the desired output of the CMAC that we try to obtain with the equation \(Y = A(u)^TW\) of the CMAC. Furthermore, we consider that the trajectories of each leg in swing phase are identicals. This makes it possible on the one hand to divide by two the number of CMAC and on the other hand to reduce the training time.

Figure 11 shows the evolution of the output of the CMAC and the desired output \(q_{i1}\) (alternatively \(q_{11}\) and \(q_{21}\)) of a CMAC. One can note that after only some steps of the robot, the CMAC correctly learned the desired trajectory.

![Fig. 11. Evolution of the output of the CMAC during the training, where \(t_j\) (\(j = 1..6\)) are the instants of transition between one leg and the other.](image)

4.3 Walking gait generation with the CMAC

After the training phase, the CMAC networks are used to generate the joint trajectories of the swing leg starting from the geometrical configuration of the stance leg. A PD control makes it possible to ensure the tracking of these trajectories. The control of the stance leg and the high level control, making it possible to control the average velocity of progression of the robot, are identical to that described in section 3. Figure 12 describes the control strategy used after the phase of training.

Figure 13 shows the evolution of the average velocity of the robot during and after the phase of training. At \(t = 7.5s\), the control strategy is modified: the CMAC networks are used to generate the joint trajectories. It should be noted that when the transition of control strategies occurs, the average velocity decreases. But the high level control (see subsection 3.3.) allows to readjust the pitch angle of the trunk and after some steps, the average velocity of the robot tends converges again towards that desired.
Fig. 12. Principle of the control strategy based on the use of the CMAC neural networks, when the leg 1 is in the stance phase.

Fig. 13. Evolution of the average desired and real velocity before and after the learning (phase of training is included in $[0 - 7.5s]$).

After this modification of the control strategy, if the desired average velocity is modified, it can be noticed that the average velocity of the robot converges towards this new desired value. In fact, one of the main advantage of this control strategy resides on the fact that the duration of the swing phase is automatically adapted to the average velocity of the robot.

The figure 14 shows the torque during one step. It must be noted that in this case, there is not really full passive phases but the two control strategies produce similar torques.

In fact, the control strategy presented in this section can be summarized in three points:

- the control of the swing leg is based on trajectories tracking. These trajectories are performed by using CMAC neural networks. The training of the CMAC is carried out during a dynamic walking achieved by a succession of passive and active phases.
- the knee of the stance leg is locked. The control of the hip of the stance leg
makes it possible to determine the pitch angle of the trunk. Consequently, the association of the stance leg and the trunk may be assimilated to an inverse pendulum during the entire phase of support.

- the modification of the pitch angle of the trunk allows us to modify the average velocity of the robot. This one is controlled automatically by exploiting a high level control using the error between the desired and real value.

In order to expose the capacities of this strategy of control, the following section shows the behavior of the robot when it is subjected to external disturbances.

5 Robustness of the robot during a walking gait

If it is true that within the framework of the control strategies such as those presented in this article, the objective is not to prove the stability of the walking of the robot, it is however possible to evaluate the robustness of this control technique when the robot is subjected to environmental disturbances (rough terrain, perturbation forces, sliding).

5.1 Adaptation to irregular grounds

As we specified in the introduction, the fact of using techniques of tracking trajectories is not without posing problems on irregular ground. However, the surface where the humanoid robots move are not always perfect. The control of the walking robot must be sufficiently robust to avoid the fall when the surface of the ground becomes slightly irregular. Thus, in order to evaluate our control strategy compared to these disturbances, we tested the dynamic walking of our virtual robot on an irregular ground. Between $t = 10s$ and $t = 20s$, at each new step, the height of the point of impact is computed in a random
way between 0 and 15mm compared to an absolute reference (Fig.15). The figure 16 shows the average and instantaneous velocities of the robot. When the robot arrives on irregular ground (between $t = 10s$ and $t = 20s$), the average velocity is disturbed. However the robot does not fall and takes again a normal cycle when there is no disturbance.

![Figure 15](image15.png)

**Fig. 15.** Height of the ground in mm when a robot moves on irregular ground (between $t = 10s$ and $t = 20s$).

![Figure 16](image16.png)

**Fig. 16.** Instantaneous horizontal velocity $V_x$ (dotted line) and average velocity $V_M$ (continuous line) when a robot moves on irregular ground (between $t = 10s$ and $t = 20s$).

### 5.2 Avoiding the fall of the robot subjected to an impulsive disturbing force

During walking, a robot moving in an unknown environment can be subjected to external forces involving an imbalance, the most critical stage being the single support phase. And if this force is too big, the robot can then fall. It is thus essential to be able to quickly react in order to find balance. Thus in this simulation, we applied to our robot an external force along horizontal axis which pushes it in rear during its walking. At $t = 12.0s$ we applied an impulse force with an amplitude of 100Nm and a duration of 0.2s. The kinetic energy is insufficient to counter this disturbance, the instantaneous horizontal velocity decreases and becomes negative (Fig. 17). The robot falls then backwards.
The figure 18 shows the successive movements of the robot before and after this disturbance. The motion of the swing leg is then reversed, what makes it possible to stop the robot in a configuration of double support and thus to avoid the fall. This is the result of the trajectories used to generate the movement of the swing leg: they are not functions of time but depend on the geometrical configuration of the stance leg with regard to the ground.

Fig. 17. Instantaneous horizontal velocity (continuous line) and impulsive horizontal perturbation force (dotted line) (equal to \(-100\,N\) when \(t \in [12\,s, 12.2\,s]\)) and 0 elsewhere.

Fig. 18. Successive movements of the robot before and after the perturbation force.

5.3 Avoiding the fall of the robot subjected to sliding

During the walking, the robots, in the same way as the humans beings, can be subjected to losses of adherence following a change of characteristics of the foot ground interaction. It is thus important to be able to evaluate the reliability of the walking of the robot during phases of sliding. The modeling of the interaction between the foot and the ground that we use in our simulations makes it possible to obtain phases of sliding followed by phase of adherence. Indeed, when the tangential force of contact is outside of the cone of friction, the tangential forces of contacts are calculated starting from equation (3). The duration of this phase of slip then depends on the viscous damping coefficient \(\mu_g\).
Fig. 19. Walking of the robot with and without sliding. During the last step of the simulation, the leg number 2 is in the swing phase.

Fig. 20. Angles $q_{21}$ and $q_{22}$ of the robot with and without sliding.

Fig. 21. Angular velocities $\dot{q}_{21}$ and $\dot{q}_{22}$ of the robot with and without sliding.

The figure 19 shows the stick diagram of the walking of the robot with and without phases of sliding. When the sliding occurs, this one is insufficient to destabilize the robot which continues its walking. But at the following step, the slip is too large, the horizontal instantaneous velocity decreases, and the robot does not finish the step but stays stable.

The figures 20 and 21 show respectively the angle and the angular velocity on the leg number two. We can observe that at the last step (stance leg is leg number two) of the robot, the articular movements are reversed. As in the case of an external perturbation force (see subsection 5.2) the robot does not fall down but returns in a stable configuration of double support.
6 Conclusion and further developments

In this paper, we proposed a control strategy allowing us to perform the dynamic walking of a virtual under-actuated robot. This control strategy was based on two stages. The first one consists in using a set of pragmatic rules in order to generate a succession of passive and active phases allowing us to perform the dynamic walking of the robot. This first stage is used as a reference control to learn, by using neural networks, a set of articular trajectories. In the second stage, we use these neural networks to generate the trajectories learned during the first stage. This approach allowed us to walk on irregular grounds and to increase the robustness of the control of the walking of this robot subjected to sliding or disturbing forces. Future work will focus on the implementation of this control strategy using the CMAC on the real robot prototype RABBIT, the training of this neural networks being carried out in simulation.

References


