9

Towards Adaptive Control Strategy for Biped Robots

Christophe Sabourin[1], Kurosh Madan[1] and Olivier Bruneau[2]
[1]*Université PARIS-XII, Laboratoire Images, Signaux et Systèmes Intelligents*
[2]*Université Versailles Saint-Quentin-en-Yvelines, Laboratoire d'Ingénierie des Systèmes de Versailles*
France

1. Introduction

The design and the control of humanoid robots are one of the most challenging topics in the field of robotics and were treated by a large number of research works over the past decades (Bekey, 2005) (Vukobratovic, 1990). The potential applications of this field of research are essential in the middle and long term. First, it can lead to a better understanding of the human locomotion mechanisms. Second, humanoid robots are intended to replace humans to work in hostile environments or to help them in their daily tasks. Today, several prototypes, among which the most remarkable are undoubtedly the robots Asimo (Sakagami , 2002) and HRP-2 (Kaneko, 2004), have proved the feasibility of humanoid robots. But, despite efforts of a lot of researchers around the world, the control of the humanoid robots stays a big challenge. Of course, these biped robots are able to walk but their basic locomotion tasks are still far from equalizing the human's dynamic locomotion process. This is due to the fact that the control of biped robot is very hard because of the five following points:

- Biped robots are high-dimensional non-linear systems,
- Contacts between feet and ground are unilateral,
- During walking, biped robots are not statically stable,
- Efficient biped locomotion processes require optimisation and/or learning phases,
- Autonomous robots need to take into account of exteroceptive information.

Because of the difficulty to control the locomotion process, the potential applications of these robots stay still limited. Consequently, it is essential to develop more autonomous biped robots with robust control strategies in order to allow them, on the one hand to adapt their gait to the real environment and, on the other hand, to counteract external perturbations.

In the autonomous biped robots' control framework, our aim is to develop an intelligent control strategy for the under-actuated biped robot RABBIT (figure 1) (RABBIT-web) (Chevallereau, 2003). This robot constitutes the central point of a project, within the framework of CNRS ROBEA program (Robea-web), concerning the control of walking and running biped robots. The robot RABBIT is composed of two legs and a trunk and has no foot. Although the mechanical design of RABBIT is uncomplicated compared to other biped

Source: Humanoid Robots: Human-like Machines, Book edited by: Matthias Hackel
ISBN 978-3-902613-07-3, pp. 642, Itech, Vienna, Austria, June 2007

robots, its control is a more challenging task, particularly because, in phase of single support, this robot is under-actuated. In fact, this kind of robots allows studying real dynamical walking leading to the design of new control laws in order to improve biped robots' current performances.

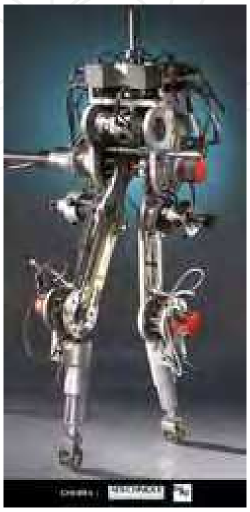


Figure 1. RABBIT prototype

In addition to the problems related to control the locomotion process (leg motions, stability), it is important to take into account both proprioceptive and exteroceptive information in order to increase the autonomy of this biped robot. The proprioceptive perception is the ability to feel the position or movements of parts of the body and the exteroceptive perception concerns the capability to feel stimuli from outside of the body. But the both proprioceptive and exteroceptive information are not treated in the same manner. The proprioceptive information, which are for example the relative angles between two limbs and the angular velocity, allow to control the motion of the limbs during one step. The exteroceptive perception must allow to obtain information about the environment around the biped robot. These exteroceptive information allow using predictive strategies in order to adapt the walking gait regarding the environment.

In fact, although the abilities of RABBIT robot are limited in comparison to other humanoid robots, our goal in middle term, is to design a control strategy for all biped robots.

In our previous works, we used CMAC (Cerebellar Model Articulation Controller) neural networks to generate the joint trajectories of the swing leg but, for example, the length of the step could not be changed during the walking (Sabourin, 2005) (Sabourin, 2006). However, one important point in the field of biped locomotion is to develop a control strategy able to modulate the step length at each step. In this manner, in addition to modulate the step length according to the average velocity, like human being, the biped robot can choice at each step the landing point of the swing leg in order to avoid obstacle. But in general, as in the case of human being, the exteroceptive information allowing to give information about obstacles in the near environment of the robot are not precise measures. Consequently, we prefer to use fuzzy information. However this implies to deal with heterogeneous data, which is not a trivial problem. One possible approach consists to use soft-computing techniques and/or pragmatic rules resulting from the expertise of the walking human.

Moreover, this category of techniques takes advantage from learning (off-line and/or on-line learning) capabilities. This last point is very important because generally the learning ability allows increasing the autonomy of the biped robot.

Our control strategy uses a gait pattern based on Fuzzy CMAC neural networks. Inputs of this gait pattern are based on both proprioceptive and exteroceptive information. The Fuzzy CMAC approach requires two stages:

- First, the training of each CMAC neural networks is carried out. During this learning phase, the virtual biped robot is controlled by a set of pragmatic rules (Sabourin, 2005) (Sabourin, 2004). As a result, a stable reference dynamic walking is obtained. The data learnt by CMACs are only the trajectories of the swing leg.
- After this learning phase, we use a merger of the CMAC trajectories in order to generate new gaits.

In addition, a high level control allows us to modify the average velocity of the biped robot. The principle of the control of the average velocity is based on the modification, at each step, of the pitch angle.

The first investigations, only realized in simulation, are very promising and proved that this approach is a good way to improve the control strategy of a biped robot. First, we show that, with only five reference gaits, it is possible to adjust the step of the length as a function of the average velocity. In addition, with a fuzzy evaluation of the distance between feet and an obstacle, our control strategy allows to the biped robot to avoid obstacle using step over strategy.

This paper is organized as follows. After a short description of the real robot RABBIT, section 2 gives the main characteristics of the virtual under-actuated robot used in our simulations. In Section 3, firstly you remind the principles of CMAC neural networks and the Takagi-Sugeno fuzzy inference system, secondly Fuzzy CMAC neural networks are presented. Section 4 describes the control strategy with a gait pattern based on the Fuzzy CMAC structure. The learning phase of each CMAC neural network is presented in section 5. In section 6, we give the main results obtained in simulation. Conclusions and further developments are finally set out.

2. Virtual modelling of the biped robot RABBIT

RABBIT robot has only four joints: one for each knee, one for each hip. Motions are included in the sagittal plane using a radial bar link fixed on a central column that allows to guide robot's advance around a circle. Each joint is actuated by a servo-motor RS420J. Four encoders make it possible to measure the relative angles between the trunk and the thigh for the hip, and between the thigh and the shin for the knee. Another encoder, installed on the bar link, gives the pitch angle of the trunk. Two binary contact sensors detect whether or not the leg is in contact with the ground. Based on the information given by the encoders, it is possible to calculate the length of the step L_{step} when the two legs are in contact with the ground. The duration of the step t_{step} is computed using the contact sensor information (the duration from take-off to landing of the same leg). Furthermore, it is possible to estimate the average velocity V_M using (1).

$$V_M = \frac{L_{step}}{t_{step}} \qquad (1)$$

The characteristics (masses and lengths of the limbs) are summarized in table 1.

Limb	Weight(Kg)	Length(m)
Trunk	12	0.20
Thigh	6.8	0.40
Shin	3.2	0.47

Table 1. Robot's limb masses and lengths

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 stance leg) to control three parameters (vertical and horizontal position of the platform and pitch angle). The numerical model of the robot previously described was designed with the software ADAMS[1] (figure 2)

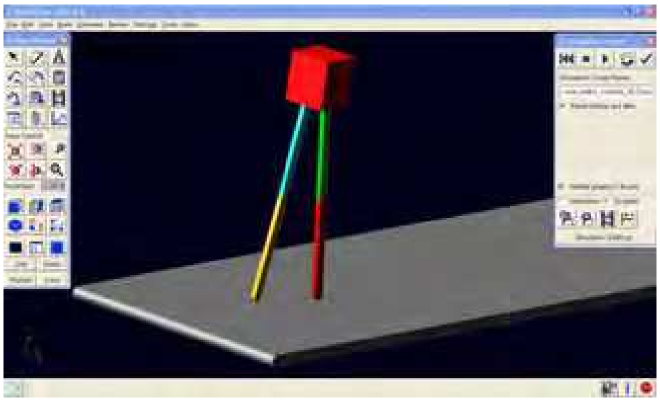


Figure 2. Modelling of the biped robot with ADAMS

This software, from the mechanical system's modelling point of view (masses and geometry of the segments) is able to simulate the dynamic behaviour of such a system and namely to calculate the absolute motions of the platform as well as the limb relative motions when torques are applied on the joints by virtual actuators. Figure 3 shows references for the angles and the torques required for the development of our control strategy.

q_{i1} and q_{i2} are respectively the measured angles at the hip and the knee of the leg i. q_0 corresponds to the pitch angle. T_{knee}^{sw} and T_{hip}^{sw} are the torques applied respectively to the knee and the hip during the swing phase, T_{knee}^{st} and T_{hip}^{st} are the torques applied during the stance phase.

The interaction between feet and ground is based on a spring-damper modelling. This approach allows to simulate more realistic feet-ground interaction namely because the contact between the feet and the ground is compliant. However, in order to take into account the possible phases of sliding, we use a dynamic friction modelling when the tangential contact forces is located outside the cone of friction. The normal contact force F_n is given by equation (2):

[1] ADAMS is a product of MSC software.

$$F_n = \begin{cases} 0 & if \quad y > 0 \\ -\lambda_n |y| \dot{y} + k_n |y| & if \quad y \leq 0 \end{cases} \tag{2}$$

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 and λ_n are respectively the generalized stiffness and damping of the normal forces. They are chosen to avoid the bouncing and limit the foot penetration in the ground. Tangential contact force F_t is computed by using equation 3 with F_{t1} and F_{t2} which are respectively the tangential contact force without and with sliding.

$$F_t = \begin{cases} F_{t1} & if \quad \|F_{t1}\| < \mu_s \|F_n\| \\ F_{t2} & if \quad \|F_{t1}\| \geq \mu_s \|F_n\| \end{cases} \tag{3}$$

With:

$$F_{t1} = \begin{cases} 0 & if \quad y > 0 \\ -\lambda_t \dot{x} + k_t (x - x_c) & if \quad y \leq 0 \end{cases} \tag{4}$$

$$F_{t2} = \begin{cases} 0 & if \quad y > 0 \\ -(\text{sgn}(\dot{x}))\lambda_g F_n - \mu_g \dot{x} & if \quad y \leq 0 \end{cases} \tag{5}$$

x and $\dot{x}$ are respectively the foot position and the velocity with regard to the position of the contact point x_c at the instant of impact with the ground. k_t and λ_t are respectively the generalized stiffness and damping of the tangential forces. λ_g is the coefficient of dynamic friction depending on the nature of surfaces coming into contact, μ_g a viscous damping coefficient during sliding, and μ_s is the static friction coefficient.

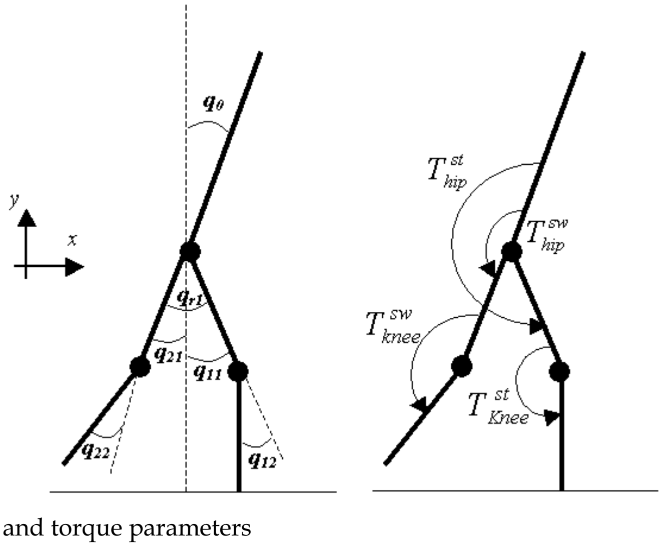


Figure 3. Angle and torque parameters

In the case of the control of a real robot, its morphological description 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. From the characteristics of servo-motor RS420J used for RABBIT, we thus choose to apply the following limitations:

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

3. Fuzzy-CMAC neural network

The CMAC is a neural network imagined by Albus from the studies on the human cerebellum (Albus, 1975a), (Albus, 1975b). CMAC is a neural network with local generalization abilities. This means that only a small number of weights are necessary to compute the output of this neural network. Consequently, the main interest is the reduction of training and computing times compared with other neural networks (Miller, 1990). This is of course a considerable advantage for real time control. Numerous researchers have investigated CMAC and have applied this approach to the field of control namely for biped robots' control and related applications (kun, 2000), (Brenbrahim, 1997). However, it is pertinent to remind that the memory used by CMAC (e.g. the needed memory size) depends firstly on the input signal quantification step and secondly of the input space size (dimension). For real CMAC based control applications, the CMAC memory size becomes quickly very big. In fact, on the one hand, in order to increase the accuracy of the control the chosen quantification step must be as small as possible; on the other hand, generally in real world applications the input space dimension is greater than two. In order to overcome the problem relating to the size of the memory, a hashing function is used. But in this case, because the size of the memory allowing to store the weights of the neural network is smaller than the size of the virtual addressing memory, some collisions can occur. Another problem occurring in the case of multi-input CMAC is the necessity to set out a learning database covering the whole input space. This is due to the CMAC local generalization abilities and results in yielding enough data (either by performing a large number of simulations available from a significant experimental setup) to wrap all possible states.

We propose a new approach making it possible to take advantage of both local and global generalization capacities with the Fuzzy CMAC neural networks. Our Fuzzy CMAC approach is based on a merger of all the outputs of several Single Input/Single Output (SISO) CMAC neural networks. This merger is carried out using Takagi-Sugeno Fuzzy Inference System. This allows both to decrease the size of the memory and to increase the generalization abilities compared with a multi-input CMAC. In this section, as a first step, we present a short description of SISO CMAC neural network. Sub-section 3.2 describes the Takagi-Sugeno Fuzzy Inference System. Finally, in sub-section 3.3 the proposed Fuzzy-CMAC approach is presented.

3.1 SISO CMAC neural networks

CMAC is an associative memory type neural network. Its structure includes a set of N_d detectors regularly distributed on several N_l layers. The receptive fields of these detectors cover the totality of the input signal but each field corresponds to a limited range of inputs. On each layer, the receptive fields are shifted to a quantification step Δ_q. When the input signal is included in the receptive field of a detector, it is activated. For each value of the input signal, the number of activated detectors is equal to the number of layers N_l (a parameter of generalization). Figure 4 shows a simplified organization of the receptive fields having *14* detectors $(N_d = 14)$ distributed on *3* layers $(N_l = 3)$. Taking into account the receptive fields overlapping, neighbouring 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 (local generalization). The output O of the CMAC is computed using two mappings. The first mapping projects an input space point e into a binary associative vector $D = [d_1,...,d_{Nd}]$. Each element of D is associated with one detector. When one detector is activated, the corresponding element in D of this detector is 1 otherwise it is equal to 0.

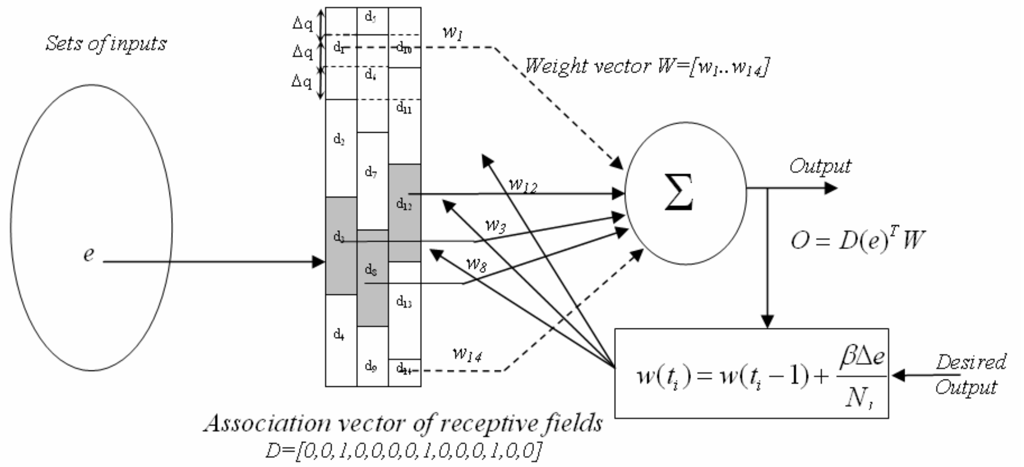

Sets of inputs
e
Δq
Δq
Δq
d5
d1
d10
d6
d11
d2
d7
d12
d3
d8
d13
d4
d9
d14
w1
Weight vector W=[w1..w14]
w12
w3
w8
w14
Σ
Output
O = D(e)^T W
w(t_i) = w(t_i − 1) + βΔe / N_l
Desired Output
Association vector of receptive fields
D=[0,0,1,0,0,0,0,1,0,0,0,1,0,0]

Figure 4. Description of the simplified CMAC with 14 detectors distributed on 3 layers

The second mapping computes the output O of the network as a scalar product of the association vector D and the weight vector $W = [w_1,...,w_{Nd}]$ according to the relation 6, where $(e)^T$ represents the transpose of the input vector.

$$O = D(e)^T W \tag{6}$$

The weights of CMAC are updated by using equation 7:

$$w(t_i) = w(t_{i-1}) + \frac{\beta\, \Delta e}{N_l} \tag{7}$$

$w(t_i)$ and $w(t_{i-1})$ are, respectively, the weights before and after training at each sample time t_i (discrete time). N_l is the generalization number of each CMAC and β is a parameter included in $[0,1]$. Δe is the error between the desired output O^d of the CMAC and the computed output O of the corresponding CMAC.

3.2 Takagi-Sugeno fuzzy inference system

Generally, the Takagi-Sugeno Fuzzy Inference System (TS-FIS) is described by a set of $R_k \ (k = 1..N_k)$ fuzzy rules such as equation 8:

$$if \ x_1 \ is \ A_1^j ...and... x_i \ is \ A_i^j \ then \ y_k = f_k(x_1,...x_{Ni}) \tag{8}$$

$x_i \ (i = 1..N_i)$ are the inputs of the FIS with N_i the dimension of the input space. $A_i^j \ (j = 1..N_j)$ are linguistic terms, representative of fuzzy sets, numerically defined by membership functions distributed in the universe of discourse for each input x_i. Each output rule y_k is a linear combination of input variables $y_k = f_k(x_1,...,x_{Ni})$ (f_k is a linear function of x_i). Figure 5 shows the structure of TS-FIS. It should be noted that TS-FIS with Gaussian membership functions is similar to the Radial Basis Function Neural Networks.

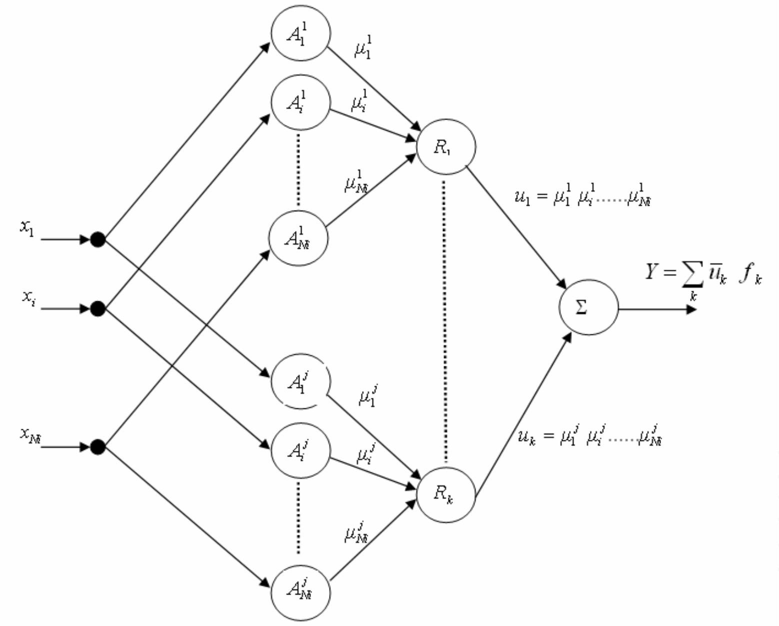


Figure 5. Description of the Takagi-Sugeno Fuzzy Inference System

The calculation of one output of TS-FIS is decomposed into three stages:

- The first stage corresponds to fuzzification. For each condition "x_i *is* A_i^j"; it is necessary to compute μ_i^j which is the numerical value of x_i input signal in the fuzzy set A_i^j.

- In the second stage, the rule base is applied in order to determine each u_k $(k = 1..N_k)$. u_k is computed using equation 9:

$$u_k = \mu_1^j \; \mu_2^j ....\mu_{Ni}^j \tag{9}$$

- The third stage corresponds to the defuzzification phase. But for TS-FIS, the output numerical value Y is carried out using the weighted average of each rule output y_k (equation 10) .

$$Y = \sum_k \overline{u_k} \; y_k \tag{10}$$

With $\overline{u_k}$ is given by equation 11:

$$\overline{u_k} = u_k \, / \sum_{k=1}^{N_r} u_k \tag{11}$$

Furthermore, in the case of the zero order Takagi-Sugeno, the rule outputs are a singleton. Consequently, for each k rule, $y_k = f_k(x_1,.....,x_n) = C_k$ where C_k is a constant value independent of the x_i input.

3.3 Fuzzy CMAC

Our Fuzzy CMAC architecture uses a combination of a set of several Single Input/Single Output CMAC neural networks and Takagi-Sugeno Fuzzy Inference System. Figure 6 describes the Fuzzy-CMAC structure with two input signals: e and X. e is the input signal which is applied at all the $CMAC_k$. $X = [x_1,...,x_{Ni}]$ corresponds to the input vector of FIS. Consequently, the output of the Fuzzy CMAC depends on the one hand on TS-FIS and on the other hand on the outputs of a set of SISO CMAC.

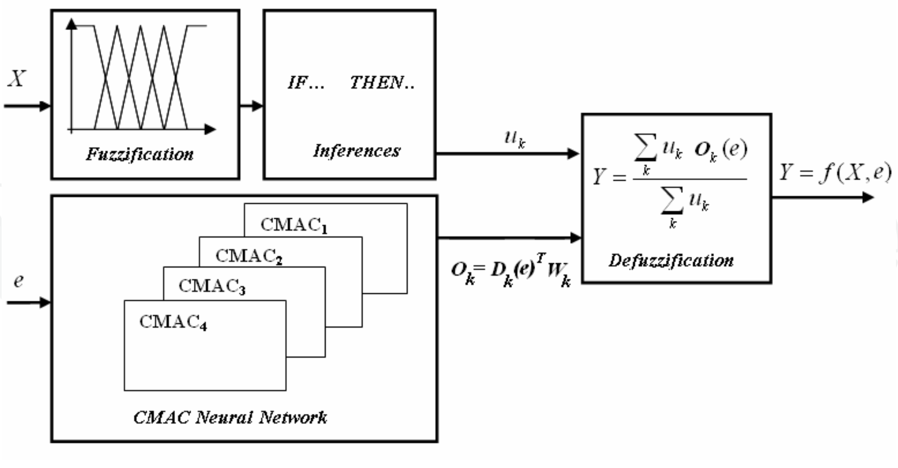


Figure 6. Bloc-diagram of the proposed Fuzzy CMAC structure

The calculation of Y is carried out in two stages:

- First, the output of each $CMAC_k$ is given by equation (12). D_k and W_k are respectively the binary associate vector and the weight vector of each $CMAC_k$ (see section 3.1).

$$O_k(e) = D_k(e)^T W_k \tag{12}$$

- Second, the output Y is carried out using equation (13). In fact, Y is computed using the weighted average of all CMAC outputs.

$$Y = \sum_k \overline{u_k}\, O_k(e) \tag{13}$$

This approach is an alternative solution of the Multi Input/Multi Output CMAC neural networks. The main advantages of the Fuzzy CMAC structure compared to MIMO CMAC are:

- First, the reduction of the size memory because the Fuzzy CMAC uses a small set of SISO CMAC,
- The global generalization capabilities because the Fuzzy CMAC uses a merger of all outputs of CMACs.

In our control strategy, we use Fuzzy CMAC to design a gait pattern for the biped robot. After a training phase of each CMAC, the Fuzzy CMAC allows us to generate the motion of the swing leg. In the next section, we present the principle used to train each CMAC neural network.

4. Training of the CMAC neural networks

During the learning phase, we use an intuitive control, based on five pragmatic rules, allowing us to perform a dynamic walking of our virtual under-actuated robot without reference trajectories. It must be pointed out that during this first stage, we both consider that the robot moves in an ideal environment (without any disturbance) and the frictions are negligible. As frictions are negligible, these fives rules allow us to generate the motions of the legs using a succession of passive and active phases. This intuitive control strategy, directly inspired from human locomotion, allows us to perform a stable dynamic walking using the intrinsic dynamic of the biped robot. It is thus possible to modify the length of the step and the average velocity by an adjustment of several parameters (Sabourin-2004). Consequently, this approach allows us to generate several reference gaits which are learnt by a set of CMAC neural networks.

In the next sub-section, a short description of the pragmatic rules to control the biped robot during the training of the CMAC neural network is presented. In sub-section 4.2, we show how the CMAC neural networks are trained. Finally, we give the main parameters for five walking used during the learning phase (Sub-section 4.3).

4.1. Pragmatic rules

The intuitive control strategy is based on the following five intuitive rules:

- During the swing phase, the torque applied to the hip given by equation (14) is just an impulse with a varying amplitude and a fixed duration equal to $(t_2 - t_1)$.

$$T_1 = \begin{cases} K_{hip}^{pulse} & if\ t_1 < t < t_2 \\ 0 & otherwise \end{cases} \tag{14}$$

Where K_{hip}^{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_{hip}^{pulse}.

- After this impulse, the hip joint is passive until the swing leg is blocked in a desired position using a PD control given by equation (15), which makes it possible to ensure a regular step length.

$$T_2 = K_{hip}^{p}(q_{r1}^{dsw} - q_{r1}) - K_{hip}^{v}\dot{q}_{r1} \quad if\ q_{r1} > q_{r1}^{dsw} \tag{15}$$

q_{r1} and q_{r1}^{dsw} are respectively the measured and desired relative angles between the two thighs, and $\dot{q}_{r1}$ is the relative angular velocity between the two thighs.

- During the stance phase, the torque applied to the hip, given by the equation (16), is used to ensure the stability of the trunk.

$$T_3 = K_{trunk}^{p}(q_0^{d} - q_0) - K_{trunk}^{v}\dot{q}_0 \tag{16}$$

Where q_0 and $\dot{q}_0$ are respectively the angle and the angular velocity of the trunk and q_0^d corresponds to the desired pitch angle of the trunk.

- 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 (17) is applied to lock this joint in a desired position q_{i2}^{dsw}.

$$T_4 = K_{knee}^{p}(q_{i2}^{dsw} - q_{i2}) - K_{knee}^{v}\dot{q}_{i2} \tag{17}$$

q_{i2} and $\dot{q}_{i2}$ are respectively the measured angular position and angular velocity of the knee joint of the leg i.

- During the stance phase, the torque is computed by using equation (18).

$$T_5 = K_{knee}^{p}(q_{i2}^{dst} - q_{i2}) - K_{knee}^{v}\dot{q}_{i2} \tag{18}$$

We choose $q_{i2}^{dst} = 0$ at the impact with the ground in equation (18) which contributes to propel the robot if $q_{i2}^{dst} > q_{i2}^{dsw}$. During the continuation of the stance phase, the same control law is used to lock the knee in the position $q_{i2}^{dst} = 0$.

4.2. Training CMACs

Figure 7 shows the method used to train CMACs neural networks. For each reference gait, four SISO $CMAC_l$ $(l = 1,..,4)$ neural networks learnt the trajectories of the swing leg (in terms of joint positions and velocities). Furthermore, we have considered that the trajectories of each leg in swing phase are identical. This allows to divide by two the number of CMAC and to reduce the training time. Consequently, two SISO CMACs are necessary to memorize the joint angles q_{i1} and q_{i2} and two other SISO CMACs for angular velocities $\dot{q}_{i1}$ and $\dot{q}_{i2}$. q_{i1} and q_{i2} are respectively the measured angles at the hip and the knee of the leg i; $\dot{q}_{i1}$ and $\dot{q}_{i2}$ are respectively the measured angular velocities at the hip and the knee of the leg i (see figure 3).

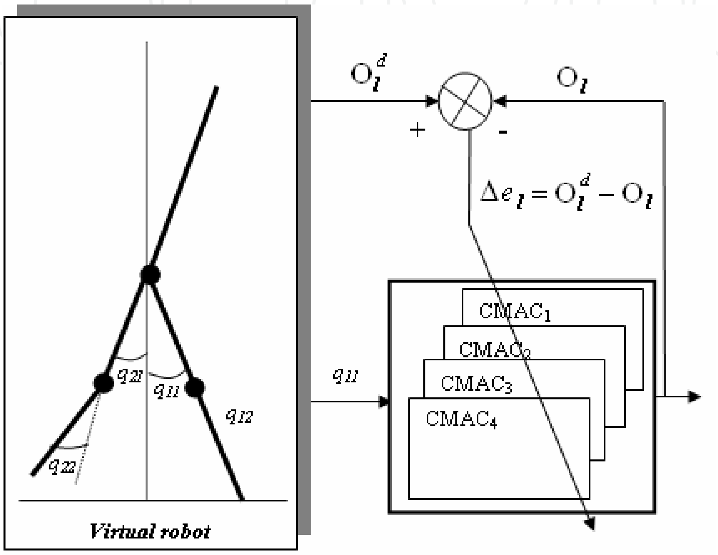
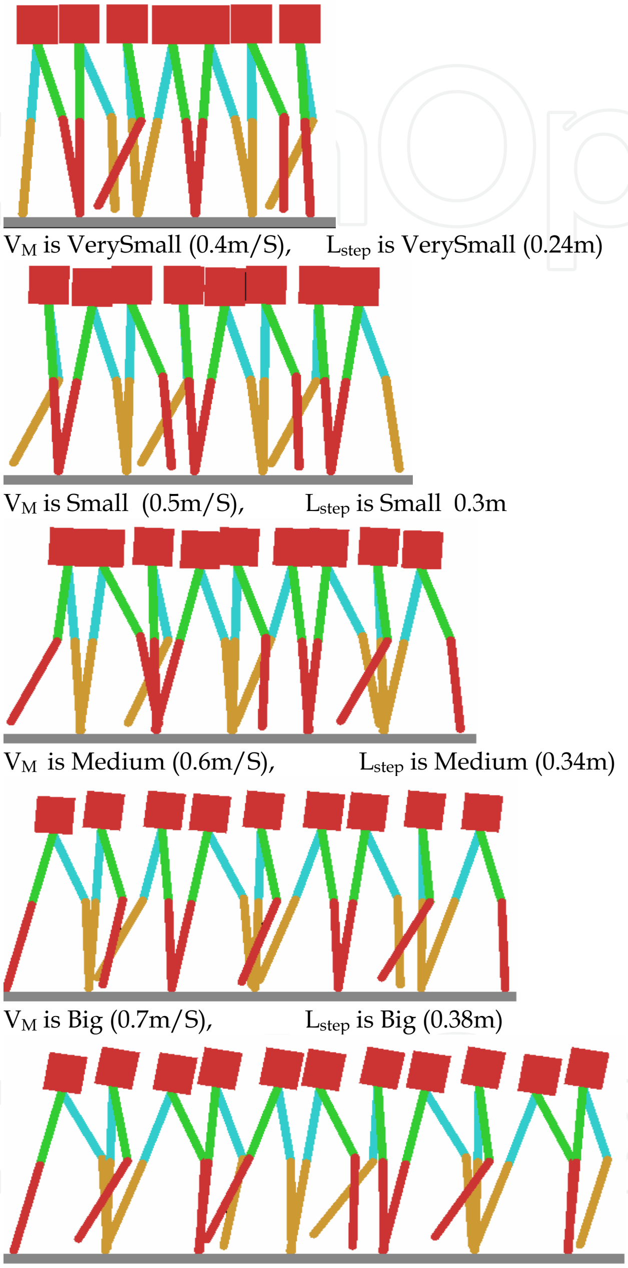


Figure 7. Principle of the learning phase of CMAC neural networks ($e = q_{11}$)

When leg 1 is in support, the angle q_{11} is applied to the input of each $CMAC_l$ ($e = q_{11}$) and when leg 2 is in support, this is the angle q_{21} which is applied to the input of each $CMAC_l$ ($e = q_{21}$). Consequently, the trajectories learnt by the neural networks are a function of the geometrical pattern of the robot. The weights of each $CMAC_l$ are updated by using the error between the desired output O_l^d ($O_1^d = q_{i1}^d, O_2^d = q_{i2}^d, O_3^d = \dot{q}_{i1}^d, O_4^d = \dot{q}_{i2}^d$) of each $CMAC_l$ and the computed output O_l of the corresponding $CMAC_l$. Based on the previous consideration, it is possible to learn N_r different reference walking using $N_r \times 4$ CMACs.
In the case of the simulations presented in this section, each CMAC has 6 layers ($N_l = 6$). The width of the receptive fields is equal to $1.5°$ and the quantification step Δ_q is equal to $0.25°$.

4.3. Reference gaits

During the training stage, five reference gaits with an average velocity V_M included in $[0.4..0.8]$ have been learnt by 5×4 single input/single output $CMAC^r$ ($N_r = 5$ and 4 CMACs for one reference walking). Table 2 gives the main parameters which are used during the learning phase according to the average velocity V_M.



V_M is VerySmall (0.4m/S), L_{step} is VerySmall (0.24m)

V_M is Small (0.5m/S), L_{step} is Small 0.3m

V_M is Medium (0.6m/S), L_{step} is Medium (0.34m)

V_M is Big (0.7m/S), L_{step} is Big (0.38m)

V_M is VeryBig (0.8m/S), L_{step} is VeryBig (0.43m)

Figure 8. Stick-diagram of the walking robot for five different velocities. V_M and L_{step} are respectively, from the top to the bottom, *VerySmall, Small, Medium, Big and VeryBig*.

	$V_M\,(m/s)$	$L_{step}\,(m)$	$q_{r1}^{dsw}\,(°)$	$q_{i2}^{dsw}\,(°)$	$q_0^d\,(°)$
$CMAC^1$	0.4	0.24	20	-7	0
$CMAC^2$	0.5	0.3	25	-10	1.5
$CMAC^3$	0.6	0.34	30	-14	4
$CMAC^4$	0.7	0.38	35	-20	6.5
$CMAC^5$	0.8	0.43	40	-25	10.5

Table 2. Parameters used during the learning stage for five different average velocities

q_{r1}^{dsw} is the desired relative angle between the two thighs (see equation 15), and q_0^d the desired pitch angle of the trunk (see equation 16). q_{i2}^{dsw} corresponds to the desired angle of the knee at the end of the knee extension of the swing leg just before the double contact phase (see equation 17). Each reference walking is characterized by a set of parameters $(q_{r1}^{dsw}, q_{i2}^{dsw}, q_0^d)$ allowing to generate different walking gaits (V_M, L_{step}). Figure 8 shows stick-diagrams representing, for five average velocities V_M and the corresponding step of the length L_{step}, the walking of the biped robot during 2.8s (approximately 6 steps). V_M and L_{step} are respectively, from the top to the bottom, *VerySmall, Small, Medium, Big, VeryBig*. It must be pointed out that each reference gait are really different and the step length L_{step} increases when V_M increases.

Based on the five reference gaits, the goal of our approach is to generate new gaits using a merger of these five learnt gaits. Consequently, after this training phase, we use a mixture between Fuzzy-Logic and the outputs of the CMACs neural network in order to generate the trajectories of the swing leg and consequently to modulate the length of the step.

In the next section, we present the control strategy based on the Fuzzy CMAC neural networks. In addition, we present the control which is used to regulate the average velocity.

5. Control strategy based on both proprioceptive and exteroceptive information

Figure 9 shows the control strategy which is used to control the walking robot. It should be noted that the architecture of this control can be decomposed into three parts:

- The first is used to compute the trajectories of the swing leg from several outputs of the $CMAC_l$ neural networks and a Fuzzy Inference System (Gait pattern). The goal of this part is, on the one hand, to adjust the step length as function of the average velocity, and on the other hand, to adapt step length in order to the robot step over obstacle.
- The second one allows the regulation of the average velocity V_M from a modification of the pitch angle q_0. When the pitch angle increases, the average velocity increases and when the pitch angle decreases, the average velocity decreases. It's in fact a good and easy way to control the average velocity of the biped robot because V_M is function of q_0.
- The third is composed by four PD control in order to ensure the tracking of the reference trajectories at the level of each joint.

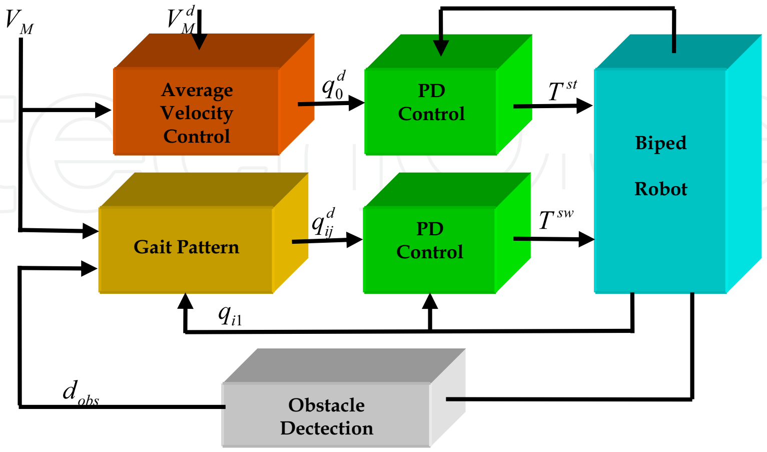


Figure 9. Structure of the control strategy for biped robot

In this section, sub-section 5.1 describes the gait pattern based on the Fuzzy CMAC approach. In sub-section 5.2, the principle of the control of the average velocity is presented. And finally, we give the control laws making it possible the track of the desired trajectories.

5.1 Gait pattern

Our gait pattern is specially designed to adjust the length of the step during walking taking into account of both proprioceptive and exteroceptive information. The inputs of the gait pattern are $e = q_{i1}$ and $X = [V_M, d_{obs}]$ where d_{obs} and V_M represent respectively, the distance between the foot, and the obstacle and the measured average velocity. During the walking, the input e is directly applied at each input of each $CMAC_k^l$. $e = q_{11}$ if leg 1 is in support, and $e = q_{21}$ if leg 2 is in support. But, the measures V_M and d_{obs} are represented using fuzzy sets. Figures 10 and 11 show the membership functions used respectively for V_M and d_{obs}. V_M and d_{obs} are modelled by five fuzzy sets (VerySmall, Small, Medium, Big, VeryBig).

Consequently, the desired angles q_{i1}^d and q_{i2}^d, and the desired angular velocities $\dot{q}_{i1}^d$ and $\dot{q}_{i2}^d$ are carried out by using a merger of the five learnt trajectories. This merger is realized by using TS-FIS. The choice of the fuzzy rules is carried out using pragmatic rules.

Without obstacle ($d_{obs} > 0.5m$), the length of the step is only a function of the average velocity. As human being, more V_M increases and more L_{step} increases. The five following rules allow us to adjust the step of the length as a function of the measured average velocity:

- If d_{obs} *is VeryBig* and V_M *is VerySmall* then L_{step} *is VerySmall*
- If d_{obs} *is VeryBig* and V_M *is Small* then L_{step} *is Small*
- If d_{obs} *is VeryBig* and V_M *is Medium* then L_{step} *is Medium*
- If d_{obs} *is VeryBig* and V_M *is Big* then L_{step} *is Big*
- If d_{obs} *is VeryBig* and V_M *is VeryBig* then L_{step} *is VeryBig*

This implies that when L_{step} *is VerySmall*, L_{step} *is Small*, L_{step} *is Medium*, L_{step} *is Big*, L_{step} *is VeryBig*, the trajectories of the swing leg $[q_{i1}^d, q_{i2}^d, \dot{q}_{i1}^d, \dot{q}_{i2}^d]$ are computed using respectively data held into $CMAC^1$, $CMAC^2$, $CMAC^3$, $CMAC^4$, $CMAC^5$.

When an obstacle is near of the robot ($d_{obs} < 0.5m$), the length of the step depends of the distance between the foot of the robot and this obstacle. Consequently, if d_{obs} *is Big* or d_{obs} *is Medium*, we choice to decrease the length of the step. And, if d_{obs} *is Small* or d_{obs} *is VerySmall*, we prefer to increase step length in order to the robot directly step over the obstacle. Table 3 shows all rules used by the Fuzzy CMAC in the case of the presented gait pattern.

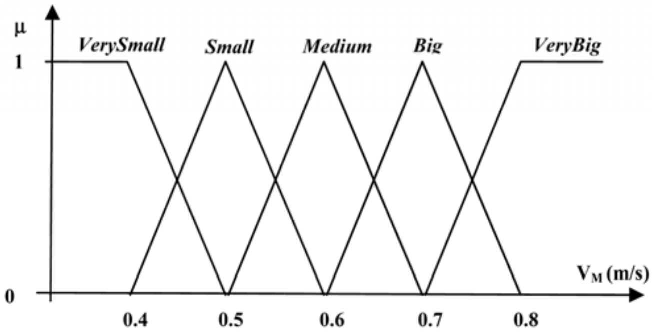


Figure 10. Membership functions used to compute V_M

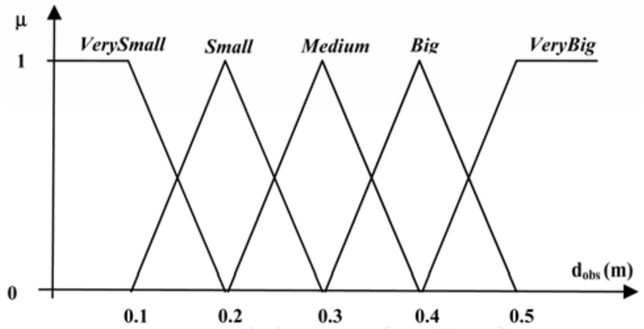


Figure 11. Membership functions used to compute d_{obs}

V_M / d_{obs}	*VerySmall*	*Small*	*Medium*	*Big*	*VeryBig*
VerySmall	O^4	O^4	O^4	O^4	O^5
Small	O^5	O^5	O^5	O^5	O^5
Medium	O^1	O^1	O^1	O^1	O^1
Big	O^1	O^2	O^3	O^4	O^4
VeryBig	O^1	O^2	O^3	O^4	O^5

Table 3. Fuzzy rules (O^k correspond to the output of the $CMAC^k$)

5.2 Average velocity control

This high level control allows us to regulate the average velocity by adjusting the pitch angle of the trunk at each step using the error between the average velocity V_M and the desired average velocity V_M^d and its derivative. V_M is calculated using equation 1. At each step, Δq_0^d, which is computed using the error between V_M and V_M^d and its derivative (equation 19), is then added to the pitch angle of the previous step $q_0^d(n)$ in order to carry out the new desired pitch angle of the following step $q_0^d(n+1)$ as shown in equation 20.

$$\Delta q_0^d = K^P(V_M^d - V_M) + K^V \frac{d}{dt}(V_M^d - V_M) \tag{19}$$

$$q_0^d(n+1) = q_0^d(n) + \Delta q_0^d \tag{20}$$

5.3 PD control

The third one is composed by four PD control in order to be sure of tracking the reference trajectories on each joint. The torques T_{knee} and T_{hip} applied respectively to the knee and to the hip are computed using the PD control. During the swing stage, the torques are carried out by using equations 21 and 22. q_{ij}^d and $\dot{q}_{ij}^d$ are respectively the reference trajectories (position and velocity) of the swing leg from the output of the Fuzzy-CMAC (j=1 for the hip, j=2 for the knee).

$$T_{hip}^{sw} = K_{hip}^{p}(q_{i1}^d - q_{i1}) + K_{hip}^{v}(\dot{q}_{i1}^d - \dot{q}_{i1}) \tag{21}$$

$$T_{knee}^{sw} = K_{knee}^{p}(q_{i2}^d - q_{i2}) + K_{knee}^{v}(\dot{q}_{i2}^d - \dot{q}_{i2}) \tag{22}$$

Secondly, the knee of the stance leg is locked, with $q_{i2}^d = 0$ and $\dot{q}_{i2}^d = 0$ (equation 23), and the torque applied to the hip allows to control the pitch angle of the trunk (equation 24). q_0 and $\dot{q}_0$ are respectively the measured absolute angle and angular velocity of the trunk. q_0^d is the desired pitch angle.

$$T_{knee}^{st} = -K_{knee}^{p}\ q_{i2} - K_{hnee}^{v}\ \dot{q}_{i2} \tag{23}$$

$$T_{hip}^{st} = K_{trunk}^{p}(q_0^d - q_0) - K_{trunk}^{v}\ \dot{q}_0 \tag{24}$$

6. Results

The goal of the two main results presented in this section is to show the interest of the proposed approach. First, we present results about the walking of the biped robot when the average velocity increases. Second, we show that the robot can step over a static obstacle.

6.1. Step length function of average velocity

Figure 12 shows the stick-diagram of the biped robot walking sequence when the desired average velocity increases. It must be noticed that the control strategy, based on the five reference gaits learnt during the training phase of CMAC neural networks (see section 4.3), allows adapting progressively the length of the step as a function of the average velocity.

Figure 13 shows the desired average velocity V_M^d, measured velocity V_M and step length L_{step}. When V_M^d increases from $0.4\,m/s$ to $1\,m/s$, V_M increases gradually and converges towards the new value of V_M^d. L_{step} increases automatically from $0.25\,m$ to $0.43\,m$ from the measured average velocity at each step. The regulation of the average velocity at each step is obtained thanks to an adequate adjustment of the pitch angle (see section 5.2). But, given that the swing leg trajectory depends on the average velocity, the length of the step is automatically adjusted as a function of V_M thanks to the Fuzzy CMAC. It must be pointed out that the average velocity is bigger than $0.8\,m/s$, the length of the step stay constant ($L_{step} = 0.43\,m$).

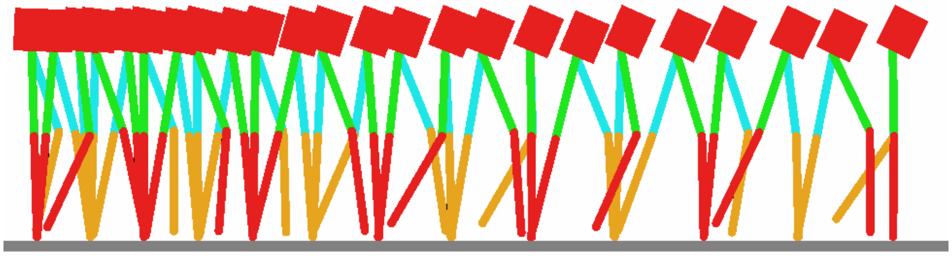


Figure 12. Stick-diagram of the walking robot when the average velocity increases

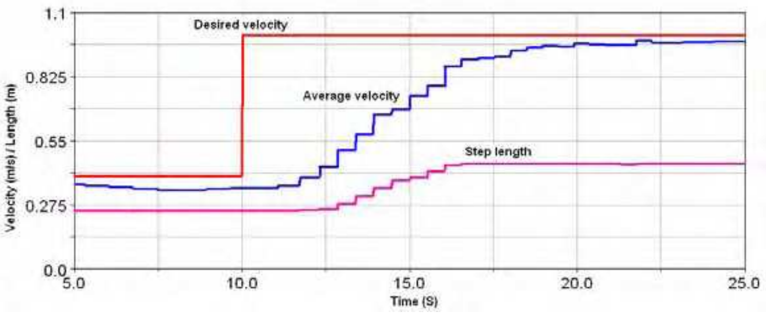


Figure 13. Average velocity and step length when the desired average velocity increases from 0.4m/s to 1m/s

6.2. Avoidance obstacle using step over strategy

The goal of this simulation is to show how the robot can step over an obstacle. In this example, the length and the height of the obstacle are respectively $0.2m$ and $0.05m$. Figures 16 and 17 show respectively stick-diagrams when the biped robot is walking on the floor whit and without obstacle. Without obstacle, the length of the step depends only of the average velocity. Consequently, L_{step} is quasi-constant during the walking. But if an obstacle occurs, our control strategy allows adjusting the step of the length in order to the robot steps over this obstacle. Figure 16 shows the length of the step when the robot is walking on the floor without and with obstacle. In the case of the presented example, the step length is adjusted in order to the landing point of the swing leg is located just before the obstacle. The next step, the step length increases allowing to the robot to step over the obstacle.

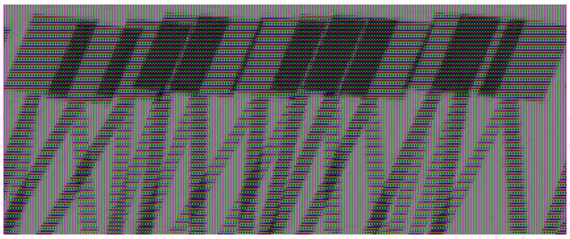


Figure 14. Walking of the biped robot without obstacle on the floor

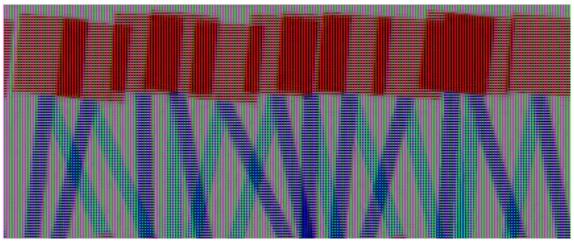


Figure 15. Walking of the robot when it steps over an obstacle

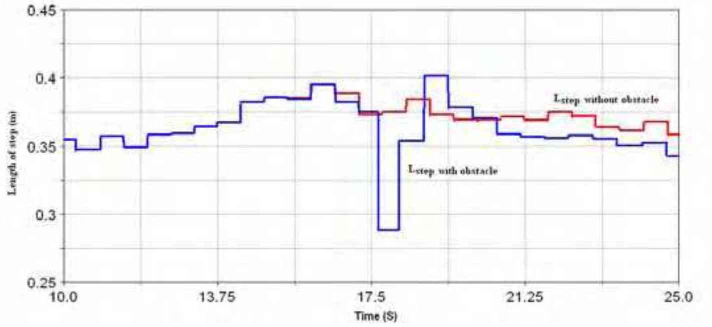


Figure 16. Length of the step when the robot is walking on the floor without and with obstacle

7. Conclusion and further works

In this chapter, we have described a control strategy based on both proprioceptive and exteroceptive information for autonomous biped robots. The first presented results, carried out on the basis of computer based simulation techniques, are very promising and prove that the proposed approach is a good way to improve the control strategy of a biped robot. First, we show that, with only five reference gaits, it is possible to generate other gaits. The adjustment of the step length as a function of the average velocity is due to the gait pattern based on the Fuzzy CMAC structure. Moreover, with a fuzzy evaluation of the distance between the robots' feet and an obstacle, our control strategy allows to the biped robot to avoid an obstacle using step over strategy.

However, it is important to remind that fuzzy rules are based on pragmatic approach and are constructed on the basis of some pre-defined membership functions shapes. For this reason, the presented control strategy may reach some limitation when biped robot comes across more complex obstacles. Furthermore, in the real word, exteroceptive perception needs to use sensors as camera. Consequently, our further works will focus on two complementary directions: the first one will concern the study of the reinforcement learning strategy in order to increase the abilities of obstacles avoidance; the other one will investigate potentials of the exteroceptive information using vision. Based on these futures works, it will be possible to carry out experimental validations on the real robot RABBIT.

8. References

Albus, J. S. (1975). A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems*, Measurement and Control,pp. 220--227.

Albus, J. S. (1975). Data storage in the cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems, Measurement and Control*, pp. 228--233.

Bekey, G. A. (2005).Autonomous Robots, from *Biological Inspiration to Implementation and Control*. The MIT Press.

Brenbrahim, A.; Franklin, J. (1997). Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, Vol.22 , pp. 283--302.

Chevallereau, C.; Abba, G., Aoustin, Y.; Plestan, F.; Westervelt, E.R.; Canudas-de-Wit, C.; Grizzle, J.W. (2003). RABBIT: A testbed for advanced control theory. *IEEE Control Systems Magazine*, Vol.23, N°5, pp. 57--79.

Kaneko, K.; Kanehiro, F.; Kajita, S.; Hirukawa, H.; Kawasaki, T.; Hirata, M.; Akachi, K.; Isozumi, T. (2004). Humanoid robot HRP-2. *Proc. IEEE Conf. on Robotics and Automation*, pp. 1083--1090.

Kun, A. L.; Miller, T. (2000). The design process of the unified walking controller for the UNH biped. *Proc. IEEE Conf. on Humanoid Robots*.

Miller, W. T.; Glanz, F. H.; Kraft, L. G. (1990). CMAC: An associative neural network alternative to backpropagation. *Proceedings of the IEEE, Special Issue on Neural Networks*, vol.78, N°10, pp. 1561-1567.

RABBIT-web: http://robot-rabbit.lag.ensieg.inpg.fr/

Robea-web: http://www.laas.fr/robea/

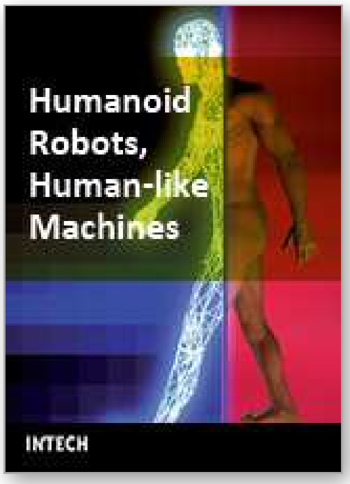
Sabourin, C.; Bruneau, O.; Fontaine, J-G. (2004). Start, stop and transition of velocities on an underactuated bipedal robot without reference trajectories. *Internationnal Journal of Humanoid Robotics*, Vol.1, N°2, pp. 349--374.

Sabourin, C.; Bruneau, O. (2005). Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks. *Robotics and Autonomous Systems*, Vol.23, pp. 81--99.

Sabourin, C.; Bruneau, O.; Buche, G. (2006). Control strategy for the robust dynamic walk of a biped robot. *The International Journal of Robotics Research* (IJRR), Vol.25, N°9, pp. 843--860.

Sakagami, Y.; Watanabe, R.; Aoyama, C.; Matsunaga, S.; Higaki, N.; Fujimura, K. (2002). The intelligent ASIMO: system overview and integration. *Proc. IEEE Conf. on Intelligent Robots and Systems*, pp. 2478--2483.

Vukobratovic, M.; Bocovac, B.; surla, D.; Stokic., D. (1990). Biped locomotion, *Scientific fundamentals of robotics* (vol 7) - Spinger-Verlag.



Humanoid Robots, Human-like Machines
Edited by Matthias Hackel

ISBN 978-3-902613-07-3
Hard cover, 642 pages
Publisher I-Tech Education and Publishing
Published online 01, June, 2007
Published in print edition June, 2007

In this book the variety of humanoid robotic research can be obtained. This book is divided in four parts: Hardware Development: Components and Systems, Biped Motion: Walking, Running and Self-orientation, Sensing the Environment: Acquisition, Data Processing and Control and Mind Organisation: Learning and Interaction. The first part of the book deals with remarkable hardware developments, whereby complete humanoid robotic systems are as well described as partial solutions. In the second part diverse results around the biped motion of humanoid robots are presented. The autonomous, efficient and adaptive two-legged walking is one of the main challenge in humanoid robotics. The two-legged walking will enable humanoid robots to enter our environment without rearrangement. Developments in the field of visual sensors, data acquisition, processing and control are to be observed in third part of the book. In the fourth part some "mind building" and communication technologies are presented.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Christophe Sabourin, Kurosh Madan and Olivier Bruneau (2007). Towards Adaptive Control Strategy for Biped Robots, Humanoid Robots, Human-like Machines, Matthias Hackel (Ed.), ISBN: 978-3-902613-07-3, InTech, Available from: http://www.intechopen.com/books/humanoid_robots_human_like_machines/towards_adaptive_control_strategy_for_biped_robots

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821