Multi-step limit cycle generation for Rabbit’s walking based on nonlinear low dimensional predictive control scheme
Ahmed Chemori, Mazen Alamir

To cite this version:

HAL Id: hal-00083873
https://hal.archives-ouvertes.fr/hal-00083873
Submitted on 27 Nov 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Multi-step limit cycle generation for Rabbit’s walking based on a nonlinear low dimensional predictive control scheme

Ahmed Chemori *, Mazen Alamir

Laboratoire d’Automatique de Grenoble, UMR 5528, BP46, Domaine Universitaire, 38400 Saint Martin d’Hères, France

Received 22 February 2005; accepted 7 December 2005

Abstract

In this paper, a new nonlinear predictive control scheme is proposed for a five-link planar under-actuated biped walking robot. The basic feature in the proposed strategy is to use on-line optimization to update the tracked trajectories in the completely controlled variables (actuated coordinates) in order to enhance the behavior and the stability of the remaining indirectly controlled ones (unactuated coordinates). The stability issue is discussed using the Poincaré’s section tool leading to a computable criterion that enables the stability of the overall scheme to be investigated as well as the computation of a candidate region of attraction. The whole framework is illustrated through simulation case-studies. To attest the efficiency of the proposed scheme, robustness against model uncertainties and ground irregularities are investigated by simulation studies.

Keywords: Biped robots; Dynamic walking; Multi-step limit cycle; Nonlinear optimization; Nonlinear predictive control; Orbital stability

Résumé

Dans cet article une nouvelle approche de commande prédictive non linéaire est proposée pour un robot marcheur bipède à cinq segments sous actionné. La caractéristique principale dans la stratégie proposée est d’utiliser l’optimisation en-ligne pour mettre à jour les trajectoires à poursuivre sur les variables complètement commandables (COORDONNÉES ACTIONNÉES) dans le but d’améliorer le comportement et la stabilité des variables indirectement commandées (COORDONNÉES NON ACTIONNÉES). La stabilité est analysée par un outil graphique basé sur la section de Poincaré. Ceci permet, en plus de l’analyse la stabilité du système en boucle fermée, d’estimer la région d’attraction. L’approche proposée est illustrée à travers différents scénarios de simulations. La robustesse, quant à elle, est analysée par rapport à des incertitudes dans le modèle du robot, et des irrégularités dans le sol.

Motsclés: Robots bipèdes; Marche dynamique; Cycle limite d’ordre multiple; Optimisation non linéaire; Commande prédictive non linéaire; Stabilité orbitale

1. Introduction

In recent years, the robotics community has shown increasing interest in the area of legged walking robots [29,2]. An excellent database of climbing and walking robots built all over the world can be found in [2]. One of the serious reasons for exploring the use of legged robots is the mobility [23], there is a need for vehicles that can travel in difficult terrains, where existing wheeled vehicles cannot go, since wheels excel on prepared surfaces such as rails and roads, but they perform poorly on rough terrains. Moreover, walking robots could co-exist with their
creators without any costly modification to the environment created for humans.

In walking locomotion [23], two gaits could be underlined. Static walking which refers to a system which stays balanced by always keeping the center of mass (c.o.m.) of the system vertically projected over the polygon of support formed by feet. On the contrary, dynamic walking [24,21,13,22] is not constrained in such a manner, therefore the c.o.m. may leave the support polygon for periods of time. Biped robots [23,34,24] have high mobility that allows them to achieve dynamic walking, consequently high speeds could be reached due to the horizontal acceleration.

Currently, many research groups in the world are working on biped robots, either on optimization of leg and foot trajectory, stable walking control, or hardware design. The main thrust of current research on biped control includes many proposed control approaches, such as intuitive control [28], intelligent learning control [19], neural network control [18], passivity based control [33], sliding control [6], impedance control [25], optimal control [14], computed torque control [5], and tracking control [31,30].

From a control viewpoint, the major academic interest of bipeds comes from (1) their hybrid nature resulting from the unavoidable impacts [3] with the ground which can produce discontinuities (jumps) in the generalized velocities. (2) Another interesting point is under-actuation, indeed biped walking robots may be under-actuated (case of Rabbit) that is the robot has fewer number of actuators than the number of degrees of freedom. This way this under-actuation is handled may be used to give a particularly clear insight into the set of solutions proposed so far within the academic control community.

1. One way to overcome the under-actuation-related difficulty is to define virtual controls [8]. Typically, reference trajectories are defined on the whole state including indirectly controlled sub-states. These trajectories are dependent on some parameter vector $p$ that can be either a virtual time, a remaining free polynomial coefficient or both. Generally, the second time-derivative of the parameter $p$ becomes a virtual additional control enabling under-actuation to be conceptually overcome. Clearly, technical details are to be investigated when this second derivative is monitored by the tracking requirements (virtual time needs to be monotonic, coefficient excursions have to be compatible with geometric constraints, etc.) The reference trajectories to be tracked may be computed using classical constrained optimal control tools. Several optimization criterions have been proposed [31,26,30].

2. A second way to handle under-actuation is to use the concept of virtual constraints and the associated zero-dynamics [12,35,36]. Namely, some regulated output is suitably defined that can be exactly tracked using the available control inputs. The constrained dynamics of the remaining sub-state on the zero-output manifold is then called the zero-dynamics [17]. This methodology is therefore based on the analytical study of the resulting zero-dynamics that corresponds to each particular choice of the regulated output. If the latter is taken in a parameterized closed-loop form, off-line optimization can then be used to enhance the asymptotic stability of the zero-dynamics [12,35,36]. A particular feature when dealing with the zero-dynamics associated to bipeds is their hybrid nature [36].

3. A third way to handle under-actuation in nonlinear dynamical systems is predictive control schemes [4,20,11]. Indeed, these schemes ensure stability by controlling the behavior of the whole state at some future time, say $N$-sampling times ahead. This naturally suppresses under-actuation since the number of control d.o.f. is $r \times N$ where $r$ is the number of actuators while the controlled state is still $n$-dimensional. Therefore, under-actuation generically disappears as soon as $Nr \geq n$.

The work proposed in this paper might be viewed as a mixture of the last two categories. Namely a nonlinear predictive control scheme is proposed for the control of a five-link 7 d.o.f. under-actuated biped robot while the stability of the resulting zero-dynamics is explicitly studied. Contrary to the approach adopted in [1], where a somehow black-box formulation is used to define the auxiliary open-loop optimization problem, our approach leads to low dimensional decision variables. This may be crucial in a real-time implementation context. In particular, it is shown that with a scalar such open-loop optimization problem, provably stable and quasi-cyclic motions can be generated.

The basic differences between the approach proposed in this paper and existing provably stable limit-cycle generation can be summarized as follows:

- The limit cycle so-obtained may include several steps. Namely, the robot configuration just after the impact is not necessarily the same as the one just after the preceding impact. This happens especially with very low dimensional (scalar) predictive control. By increasing the d.o.f. of the control parameterization, classical one-step limit cycles may be recovered, but the underlying predictive control may not be real-time implementable.
- The resulting closed-loop trajectories do not necessarily correspond to a periodic motion of the torso. The latter converges to a neighborhood of a stable limit cycle. This is a crucial point since it has been pointed out in [27,9] that such periodic motion is hard to achieve with at least polynomial trajectories of the actuated variables.

Sufficient conditions for the stability of the feedback scheme are derived together with a concrete computation procedure to compute the corresponding region of attraction related to the zero-dynamics of the closed-loop system.
This paper is organized as follows. First, the biped robot prototype is described in Section 2. Then the proposed predictive control approach is presented in a rather general setting (Section 3). Computable sufficient conditions for stability are derived and implementation related topics are discussed in Section 4. Finally simulation results are given in Section 5, illustrating the potentiality of the proposed solution. These simulations include scenarios where a stable walk is obtained from the rest position as well as transitions between different desired mean walking speeds. Robustness against model parameters uncertainties and ground irregularities are also verified. The paper ends by some concluding remarks.

2. The RABBIT prototype description

The academic prototype RABBIT [10] is a biped walking robot with five links and seven d.o.f. (see Fig. 1), which results from the joint effort of several French laboratories (Mechanical engineering, Automatic control, and Robotics) working on a project on control of walking robots. By means of guidance device, RABBIT walks in a circular path (see Fig. 2) while looking like a planar biped. The counter-balance should be used to offset the weight of the lateral stabilization bar in the guidance device. More technical details about the testbed can be found in [10].

2.1. Dynamic model

Using Lagrange formulation [32], the mathematical model describing the biped moving in the sagittal plane is as follows:

\[
M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) = Su + F_{\text{ext}}
\]  

(1)

where \(M(q)\in\mathbb{R}^{7\times7}\) is the inertia matrix, \(N(q, \dot{q})\in\mathbb{R}^{7}\) contains the centrifugal and Coriolis forces terms, \(G(q)\in\mathbb{R}^{7}\) is the vector of gravitational forces, \(u = [u_1 \ u_2 \ u_3 \ u_4]^{T}\in\mathbb{R}^{4}\) is the vector of control inputs, \(S\) is a torque distribution matrix, \(q = [q_{31} \ q_{41} \ q_{32} \ q_{42} \ q_{1} \ x \ y]^{T}\in\mathbb{R}^{9}\) is the vector of generalized coordinates (see Fig. 3). Finally, \(F_{\text{ext}}\) represents the external forces acting on the robot (contact forces with the ground). Following the proposed decomposition of the walking cycle proposed in [12], the walking cycle can be divided into two consecutive phases of motion (see Fig. 4). In the first one, the biped remains in contact with the ground through one foot (single support (SS) phase). The other one is the impact phase [3] that is often considered as instantaneous and characterized by a collision between the swing leg and the ground. Since the assumption that the robot is walking on horizontal surface without obstacles is made, the switching from one walking phase to another is closely related to the vertical position of the robot free leg tip. Let this position be denoted by \(\sigma(q)\), the stance leg is denoted by \((q_{31}, q_{41})\), and the swing leg by \((q_{32}, q_{42})\), therefore

\[\sigma(q) = \frac{q_{2}}{q_{1}}\]

\[
\begin{align*}
\text{SS phase} & : \dot{x} = f_1(x) + g_1(x)u \\
\text{Impact phase} & : x(x^+) = \Delta(x(t)) \\
\sigma(q^+) & > 0 \implies \text{Yes} \\
\sigma(q^+) & \leq 0 \implies \text{No} \\
\text{DS phase} & : \dot{x} = f_2(x) + g_2(x)u
\end{align*}
\]

Fig. 1. RABBIT prototype testbed.

Fig. 2. The guidance device.

Fig. 3. Schematic view of RABBIT’s mechanical structure.

Fig. 4. The walking cycle decomposition.

\[1\] For a detailed information, see http://robot-rabbit.lag.ensieg.inpg.fr/.
On the other hand, the impact model must account for the jumps in velocities and the permutation of coordinates. Our objective is then to derive the post-impact velocities in terms of pre-impact positions and velocities. During the impact we have

\[ M(q) \dot{\bar{q}} + N(q, \dot{q}) \dot{q} + G(q) = \dot{S}u + \delta F_{\text{ext}} \]  

where \( F_{\text{ext}} \) represents the external forces.

Under suitable assumptions (see e.g. \[15\]) on the impact phenomenon, one can deduce the external acting forces by integration of (8) over the impact duration, so one obtains

\[ M(q)(\dot{q}^+ - \dot{q}^-) = F_{\text{ext}} = J_2^T(q) \lambda \]  

where \( \dot{q}^+ \) (respectively, \( \dot{q}^- \)) is the velocity just after (respectively before) impact, and \( F_{\text{ext}} = \int_0^\tau \delta F_{\text{ext}} \). \( J_2(q) \) is the Jacobian matrix of the cartesian coordinates of the swing leg foot, given by

\[
\begin{align*}
J_1(q) &= \frac{y_l + l_1 \cos(q_{31}) + l_4 \cos(q_{32} + q_{42})}{l_3}, \\
J_2(q) &= \frac{x_l - l_1 \sin(q_{31}) - l_4 \sin(q_{32} + q_{42})}{l_3}
\end{align*}
\]

which gives using the expressions of \( x_{p_2}, y_{p_2} \)

\[ J_2(q) \dot{q}^- = 0 \]  

The solution of (9)-(12) leads to

\[ \begin{align*}
q^+ &= \frac{I - M^{-1}J_1^T J_2^T \left( J_2 M^{-1} J_1^T \right)^{-1} J_2}{D(q)} q^- \\
\lambda &= -\left( J_2 M^{-1} J_1^T \right)^{-1} J_2 \dot{\bar{q}}^-
\end{align*} \]

On the other hand, the impact model must account for the re-labelling of the robot coordinates (i.e. the swing leg becomes the new stance leg and vice versa), this can be expressed by

\[ \begin{align*}
\dot{q}^+ &= R(q) \left( \begin{array}{c} q^- \\ \dot{q}^\perp \end{array} \right) \\
\dot{q}^\perp &= \Delta(q) \left( \begin{array}{c} q^- \\ \dot{q}^\perp \end{array} \right)
\end{align*} \]

To summarize, the global impact model that includes both the jumps in velocities and the permutation of coordinates and velocities shortly writes

\[ \begin{align*}
\dot{q}^+ &= R(q) \left( \begin{array}{c} q^- \\ \dot{q}^\perp \end{array} \right) \\
\dot{q}^\perp &= \Delta(q) \left( \begin{array}{c} q^- \\ \dot{q}^\perp \end{array} \right)
\end{align*} \]

2 The system looks like a kinematic chain \[15\].

3 But the generalized positions still unaltered i.e. \( q^+ = q^- = q \).
where

\[
\Delta(q) = \begin{pmatrix} R(q) & 0 \\ 0 & R(q)D(q) \end{pmatrix} R = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\]

3. The key idea: a predictive control scheme

Under single support assumption, the five independent degrees of freedom can be subdivided into two parts

\[ z_1 := q_1 \in \mathbb{R}^1; \quad z_2 := (q_{31}, q_{41}, q_{32}, q_{42})^T \in \mathbb{R}^4 \]

where \( z_2 \) can be assumed to be completely controllable (provided that saturation constraints on actuators and contact conditions are fulfilled). In this section, the sequence of impact instants is denoted by \( (t_k)_{k \in \mathbb{N}} \) with \( t_k = kt_f \) where \( t_f \) is the step duration.

**Remark 1.** Under nominal conditions, the step duration \( t_f \) is fixed and it does not change from one step to another, nevertheless if the biped is required to change the walking speed, among others, a solution could be investigated to change this parameter, but for each speed and configuration corresponds a well specified value of \( t_f \).

Let us choose some target configuration \( z_f^1 \in \mathbb{R}^4 \) (cf. Section 5.1) that is to be reached just before the impact instants \( t_k \) that is \( z_2(t_k) = z_f^1 \). This choice is fixed in all the forthcoming developments, in a way, \( z_f^1 \) has to be considered as a design parameter. The way \( z_f^1 \) may be parameterized is explained in Section 5.1.

Associated to this choice of \( z_f^1 \), the following choice \( z_f^2(z_2) \in \mathbb{R}^4 \) is done for the desired \( z_2(t_k) \), this choice is defined given some desired foot impact velocity \(-v_{p_2} \)

\[
z_f^2(z_2^1, v_{p_2}) := \text{Arg min} \| z_2^1 - z_2 \|^2 \quad \text{under} \quad \frac{\partial y_{p_2}}{\partial z_2} (z_f^2, z_2) = -v_{p_2} \quad \text{and} \quad -\frac{\partial y_{p_2}}{\partial z_2} (z_f^2, z_2) \neq 0
\]

where \( y_{p_2}(z_f^2) \) is the \( y \)-coordinate of the swing foot.

In what follows, the following notations are used

\[
\begin{align*}
\mathcal{Z}_2 & := \left( z_2 \right) \in \mathbb{R}^4; \\
\mathcal{Z}_f^1 & := \left( z_f^1(z_2^1, v_{p_2}) \right) \in \mathbb{R}^4; \\
\mathcal{Z}_1 & := \left( q_1 \right) \in \mathbb{R}^2
\end{align*}
\]

Now, during the step, let us denote by \( \eta > 0 \) the remaining time before impact. One has the following dynamic for \( \eta \)

\[
\dot{\eta} = -1 + \delta(\eta) \cdot t_f
\]

where \( \delta(\cdot) \) is the generalized impulse function. Consider a control sampling period \( t_c > 0 \) such that \( t_f/t_c = N_c \in \mathbb{N} \) (\( N_c \) is also a design parameter).

Basically, a problem of synchronizing the sampling times to the impact times could appear when impact is either detected prematurely (i.e. before the expected instant) or detected with a delay (i.e. after the expected instant). Since RABBIT prototype feet are equipped with switches, the impact instant could easily be detected. This situation is managed as indicated in Section 4.3 concerning implementation issues.

Let us use the following notation to refer to decision instants \( \left[ t_k, t_k + t_f \right] \) on the interval \( \left[ t_k, t_k + t_f \right] \)

\[
t_k = t_k + it_c, \quad i \in \{0, \ldots, N_c - 1\}; \quad k \in \mathbb{N}
\]

During the step, at each decision instant \( t_k^i \), a \( p \)-parameterized reference trajectory

\[
\mathcal{Z}_f^{\text{ref}}(t_k^i, z_2(t_k^i), z_f^1(t_k^i), q_{1(t_k^i)}), p); \quad t_k^i \in \left[ t_k, t_k + t_f \right]; \quad p \in \mathcal{P}
\]

is defined that satisfies for all parameter value \( p \in \mathcal{P} \) the following boundary (initial and final) conditions

\[
\begin{align*}
\mathcal{Z}_f^{\text{ref}}(t_k^i, z_2(t_k^i), z_f^1(t_k^i), q_{1(t_k^i)}), p) &= \mathcal{Z}_1(t_k^i) \\
\mathcal{Z}_f^{\text{ref}}(t_k+1, z_2(t_k+1), z_f^1(t_k+1), q_{1(t_k+1)}), p) &= \mathcal{Z}_2(t_k^i)
\end{align*}
\]

namely, the reference trajectory \( \mathcal{Z}_f^{\text{ref}}(\cdot, z_2(t_k^i), z_f^1(t_k^i), q_{1(t_k^i)}), p) \) is updated at each decision instant \( t_k^i \) to start at the present value \( \mathcal{Z}_2(t_k^i) \), and to join the desired final value \( \mathcal{Z}_2(t_k+1) \) just before next impact.

It is worth noting that \( p \in \mathcal{P} \) is the remaining free parameter, once the constraints (19) and (20) have been structurally imposed, on some initial parameterization. This is typically easy to realize with polynomial parameterization [9] of trajectories since (19) and (20) are linear constraints in the polynomial coefficients.

A relevant question is: how to choose \( p \in \mathcal{P} \)?

The role of \( p \) is clearly to optimize the behavior of the indirectly controlled sub-state \( \mathcal{Z}_1 \). Indeed, imagine that a perfect tracking of the reference trajectory \( \mathcal{Z}_f^{\text{ref}}(\cdot, z_2(t_k^i), z_f^1(t_k^i), q_{1(t_k^i)}), p) \) is performed over \( [t_k^i, t_k+1] \). What are the consequences of such tracking on the value of both \( \mathcal{Z}_1 \) and \( \mathcal{Z}_2 \) just before the \( (k + 1) \) step?

- For \( \mathcal{Z}_2 \), one would clearly have, because of the perfect tracking [see (20)]

\[
\mathcal{Z}_2(t_{k+1}) = \mathcal{Z}_2(t_k^i)
\]

- For the \( \mathcal{Z}_1 \) dynamic, let us consider the torso equation extracted from the dynamic model (1), and given by

\[\text{In [16] for instance such trajectories are generated using Van der Pol oscillators.}\]
\[ \left( \frac{1}{4} m_1 l_1^2 + l_2 \right) \ddot{q}_1 = \frac{1}{2} m_1 l_1 \cos(q_1) \ddot{x} - \frac{1}{2} m_1 l_1 \sin(q_1) (\ddot{y} + g) - u_1 - u_2 \]  

where \( m_1 \) is the mass of the torso, \( l_1 \) its length, \( u_1 \) and \( u_2 \) are the torques of the femurs.

Using Eqs. (4) and (5) and notations (16), this dynamic should be written

\[ \dot{X}_1 = f(X_1, X_2, u) \]  

The closed-loop system is obtained by state feedback,

\[ v = K(X_1, X_2) \]  

where \( K \) is defined on the completely controlled variables (actuated joints) and tracked using a nonlinear time varying feedback during the time interval \([t_k^i, t_k^{i+1}]\). At the next decision instant \( t_k^{i+1} \) a new reference trajectory

\[ \dot{X}_2^{ref}(\tau') = (X_2(t_k^i), X_2(\tau'), \eta(t_k^i), \dot{\theta}(t_k^i)) ; \tau' \in [t_k^i, t_k^{i+1}] \]

is defined, based on the new measurements and is tracked during the time interval \([t_k^{i+1}, t_k^{i+2}]\) and the scheme is repeated until the impact instant. This defines a predictive control scheme in which the open-loop auxiliary optimization problem is given by (28). The solution of such optimization problem is performed using the DBCPOL function from the IMSL math library of Digital Fortran 5.0.

### 4. Stability and implementation issues

The stability can be investigated using the Poincaré's section [17] just before the impact, namely at instants \( t_k^i \). Indeed, if this discrete-time map converges, then a cyclic trajectory results (see Figs. 5 and 6). To study the stability of the Poincaré's map, note that, by definition of the predictive control strategy depicted in the previous section, one clearly has

\[ X_2(t_k^i) = X_2 \]  

where \( X_2 \) is the desired final "just before impact" configuration defined by (15) and (16) and depending only on the desired final position \( z_f \). Consequently, the overall stability depends on the stability of the sequence

\[ (X_2(t_k^i))_{k \in \mathbb{N}} \]

under the constraint (30).

#### 4.1. Stability definition

As it can be easily understood from Figs. 5 and 6, an asymptotically stable \( k_0 \)-cyclic trajectory on the whole state results whenever the following property holds for the closed-loop system's behavior:

\[ \|X_2(t_{k+1}) - X_2 \| \leq \mu \|X_2(t_{k}) - X_2 \|, \quad \mu < 1 \]  

for some small \( \epsilon > 0 \).

To summarize,\(^{\text{a}}\) during the step, at each decision instant \( t_k^i \) with \( i < N_c - 1 \) the reference trajectory

\[ X_2^{ref}(\tau', X_2(t_k^i), \eta(t_k^i), \dot{\theta}(t_k^i)) ; \quad \tau' \in [t_k^i, t_k^{i+1}] \]

is defined on the completely controlled variables (actuated joints) and tracked using a nonlinear time varying feedback during the time interval \([t_k^i, t_k^{i+1}]\). At the next decision instant \( t_k^{i+1} \) a new reference trajectory

\[ X_2^{ref}(\tau', X_2(t_k^{i+1}), X_2(\tau'), \eta(t_k^{i+1}), \dot{\theta}(t_k^{i+1})) ; \quad \tau' \in [t_k^i, t_k^{i+1}] \]

is defined, based on the new measurements and is tracked during the time interval \([t_k^{i+1}, t_k^{i+2}]\) and the scheme is repeated until the impact instant. This defines a predictive control scheme in which the open-loop auxiliary optimization problem is given by (28). The solution of such optimization problem is performed using the DBCPOL function from the IMSL math library of Digital Fortran 5.0.

---

\(^{\text{a}}\) A chart flow better illustrating the principle of the approach is given in Section 4.3.
Proposition 1  

1. If for some \((k_0, N_c) \in \mathbb{N} \times \mathbb{N}\), there is \(\varrho > 0\) such that

\[
\sup_{\|\mathcal{X}_1 - \mathcal{X}_f\|_0^2 < \varepsilon} \left\| \Gamma^{k_0}(\mathcal{X}_1, \mathcal{X}_f, N_c) - \mathcal{X}_f \right\|_0^2 \leq \varrho \leq \rho \tag{35}
\]

then the predictive control closed-loop leads to a stable walk for all initial conditions belonging to the set

\[
\mathcal{E}_0 := \left\{ \mathcal{X} = \left( \mathcal{X}_1 \mathcal{X}_2 \right) \text{ s.t. } \mathcal{X}_1 \in \mathcal{M}_c \right\} \tag{36}
\]

where for all \(\varrho > 0\), \(\mathcal{M}_c := \left\{ \mathcal{X}_1 : \|\mathcal{X}_1 - \mathcal{X}_f\|_0^2 \leq \varrho \right\} \).

2. If in addition, the following condition holds for some \(\mu \in [0, 1]\)

\[
\text{For all } 0 < r < q \quad \psi(r, N_c, k_0) := \sup_{\|\mathcal{X}_1 - \mathcal{X}_f\|_0^2 < \varepsilon} \|\Gamma^{k_0}(\mathcal{X}_1, \mathcal{X}_f, N_c) - \mathcal{X}_f \|_0^2 \leq \mu \cdot r \tag{37}
\]

then the closed-loop trajectories asymptotically converges to a stable limit cycle of length \(k_0\) defined by the pair \((\mathcal{X}_1, \mathcal{X}_f)\) for all initial conditions in \(\mathcal{E}_0\).

3. If (35) holds, and if (37) holds for all \(r \in [\varepsilon, q]\) and

\[
\text{for all initial conditions in } \mathcal{E}_1 := \left\{ \mathcal{X} = \left( \mathcal{X}_1 \mathcal{X}_2 \right) \text{ s.t. } \mathcal{X}_1 \in \mathcal{M}_c \right\} \tag{39}
\]

(see Fig. 7 for a typical situation) then the set

\[
\mathcal{E}_1 := \left\{ \mathcal{X} = \left( \mathcal{X}_1 \mathcal{X}_2 \right) \text{ s.t. } \mathcal{X}_1 \in \mathcal{M}_c \right\}
\]

is invariant and attractive for all initial conditions in \(\mathcal{E}_0\) (an \(\varepsilon\)-neighborhood of the limit cycle is reached).

Proof

1. Straightforward since condition (35) implies that the set \(\mathcal{M}_c\) is invariant under the composed map \(\Gamma^{k_0}(\cdot, \mathcal{X}_f, N_c)\), more precisely

\[
\left\{ \mathcal{X}_1 \in \mathcal{M}_c \right\} \Rightarrow \left\{ \Gamma^{k_0}(\mathcal{X}_1, \mathcal{X}_f, N_c) \in \mathcal{M}_c \right\} \tag{40}
\]

Therefore, starting from some initial value \(\mathcal{X}_1^0 \in \mathcal{M}_c\), the sequence

\[
\psi(r, N_c, k_0)
\]

Fig. 7. Typical situation where (35), (37) and (38) hold (point 3 of Proposition 1).
belong to the compact set $M_e$.

2. Condition (37) implies that for all $j \in \mathbb{N}$, one has
\[
\|z_1(t_{j+1}) - z_1(t_j)\|_Q \leq \mu\|z_1(t_j) - z_1(t_{j-1})\|_Q
\]
(41)

where $\mu < 1$). Accordingly, by recurrence, one obtains
\[
\|z_1(t_j) - z_1(t_{j-1})\|_Q \leq \mu^n\|z_1(t_{j-n}) - z_1(t_{j-n-1})\|_Q
\]
(42)

which implies that
\[
\lim_{j \to \infty} z_1(t_j) = z_2(t)
\]

This shows that the closed-loop trajectories tends to a

limit cycle of length $k_0$ defined by the pair of desired values

$\{z_1^*, z_2^*\}$. The obtained trajectory, that satisfy these constraints,

may be illustrated in Fig. 8.

The used subroutines provide the cubic spline interpola-

tion, which should be used to evaluate the trajectory and

its derivatives, at each instant $t'$. The switching to a new

step is closely related to the impact occurrence. The imple-

mented simulator (using visual Fortran 5.0 and Mat-

lab 6.5 softwares) switches to a new step once it detects

an impact, therefore three possible cases could be under-

lined.

Note again that the investigation of (35)-(38) may be

done off-line simultaneously and in a deterministic way

by solving the following two-dimensional optimization

problem
\[
\psi(r, N_c, k_0) := \sup_{|z_1 - z_1^*|_Q > \varepsilon} \|z_1^*(z_1, z_2, N_c) - z_1\|_Q
\]

for increasing values of $\varepsilon$ and check wether the curve so ob-
tained [see Fig. 7] satisfies (35)-(38) for some $\varepsilon > 0$ and

$\epsilon > 0$. The whole procedure may be repeated for different

values of $k_0$. Concrete examples of such plots are given in

Section 5 for a specific choice of the design parameters

$z_1^*, z_2^*$ and $N_c$. It is then shown that the condition of point

3. of Proposition 1 are satisfied for $k_0 = 3$ (see Fig. 11)

while it is not satisfied for $k_0 = 1.2$. This shows the need

for non trivial multi-step map (34) in establishing the sta-

bility of the underlying closed-loop behaviour.

4.3. Implementation issues

The reference trajectories (18) are implemented using

Matlab cubic spline interpolation functions with various

end-conditions. They are parameterized with a free param-

eter $p$ which should be computed by solution of the optimi-

zation problem (28). The use of the cubic spline functions

requires the definition of the end-conditions, in our case

they are given by

- initial-time conditions, given by Eq. (19),
- intermediate-time conditions, given by the parameter $p$
  to be computed,
- final-time conditions, given by Eq. (20).

The obtained trajectory, that satisfy these constraints,

may be illustrated in Fig. 8.

The used subroutines provide the cubic spline interpola-

tion, which should be used to evaluate the trajectory and

its derivatives, at each instant $t'$. The switching to a new

step is closely related to the impact occurrence. The imple-

mented simulator (using visual Fortran 5.0 and Mat-

lab 6.5 softwares) switches to a new step once it detects

an impact, therefore three possible cases could be under-

lined.

1. The biped walks without external disturbances, the
dynamic model is perfect, as well as the tracking of

the optimal reference trajectories.

2. During walking, because of external disturbances,

model imperfections, or obstacles, the impact is detected

prematurely.

3. The biped, during walking is subject to external distur-

bances, model imperfections, or environment changes,
as a consequence the impact is not detected at the

expected time (instant).

How the control system would react?

In the first case there is no problem, the whole closed-

loop system behavior looks like the predicted one. In the

second case, when the impact is detected, the reached con-

figuration after the impact is considered as an initial

configuration, the final desired configuration is then com-

puted, and a new step starts (illustrated in simulation 4). While

in the third case, the impact is not detected at the

expected instant, so the control system proceed to

extrapolation of the computed trajectories (using the

Matlab PPVAL function) until the occurrence of the forth-

coming impact. The whole control approach is summarized

in the diagram depicted in Fig. 9 that illustrates how it

works.

5. Illustrative simulations

Consider the biped robot model (7) and (14) with the

parameters summarized in Table 1. The control parameter

$N_c = 1$ is used in the following simulations, enabling a large

admissible on-line computation time. Indeed, with this

choice, $\tau_c = t_f$ and the trajectories being tracked during

the step are updated just after each impact. The following
choice of the parameter $p$ is used in the definition of the predictive control law (see Section 3):

$p(t_k) := \pi - \arctan \left( \frac{d}{y} \right) \frac{p}{\rho} - \varphi_{31}$
$q_{32} = \pi + \arctan \left( \frac{1 - \rho d}{y} \right) - \varphi_{32}$
$q_{41} = \pi - \varphi_{41} = \arccos \left( - \frac{l_3^2 + l_4^2 - (1 - \rho)^2 d^2 - y^2}{2l_3l_4} \right)$
$q_{42} = \pi - \varphi_{42} = \arccos \left( - \frac{l_3^2 + l_4^2 - \rho^2 d^2 - y^2}{2l_3l_4} \right)$

where

- The second one illustrates the transition between several walks with different mean walking velocities.

For robustness evaluation of the proposed controller, two scenarios are investigated, namely

- Robustness against uncertainties in the robot model parameters.
- Robustness against ground irregularities.

Let us first illustrate how $z_f^2$ is chosen by means of a reduced dimensional parameterization. The way such choice of $z_f^2$ may be made optimal in some sense is beyond the scope of the present paper and will be investigated in later works.

5.1. Reduced dimensional parameterization of the position vector $z_2$

Consider the instantaneous double support configuration. The position vector $z_2 := (q_{31}, q_{41}, q_{32}, q_{42})^T$ is defined by three simple parameters, namely $y, d$ and $\rho$ that are illustrated in Fig. 10.

Table 1
The model parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mass (kg)</th>
<th>Length (m)</th>
<th>Inertia (kg m$^2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Torso</td>
<td>20</td>
<td>0.625</td>
<td>2.22</td>
</tr>
<tr>
<td>Femur</td>
<td>6.8</td>
<td>0.4</td>
<td>1.08</td>
</tr>
<tr>
<td>Tibia</td>
<td>3.2</td>
<td>0.4</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Two simulation scenarios are proposed:

- The first one shows how the biped reaches a stable walk with constant mean velocity starting from rest.
This enables a simple choice of the desired final configuration just before the impact \( z_f \) using parameters that are directly linked to the mean velocity and the geometric configuration \( \mathcal{C}_0 \).

5.2. Simulation 1: cyclic forward walking starting from rest (standing position)

The aim of this simulation is to take the robot from a rest position to a constant speed periodic walking. The configuration \( z_f \) and the other control design parameters are summarized in Table 2.

5.2.1. Stability analysis according to Proposition 1

In this section, it is shown that under the feedback defined above, the sufficient conditions invoked in point 3. of Proposition 1 are satisfied. This can be verified on Fig. 11, that shows the multi-step map

\[
\psi(r, N_c, k_0) = \sup_{||\mathcal{F}_1 - \mathcal{F}_f||^2_Q < r} ||\mathcal{F}_1||^2_Q
\]

invoked in Proposition 1, for the two cases corresponding to \( k_0 = 1, 2 \). Note that:

- For \( k_0 = 1 \) the conditions of Proposition 1 are not satisfied. Higher values of \( k_0 \) must be investigated in order to prove stability of the closed-loop system. Recall that \( k_0 \) is only an analysis tool and not a design tool.

- For \( k_0 = 2 \), the conditions of point 3 of Proposition 1 are satisfied with \( q \approx 0.56 \) and \( \varepsilon \approx 0.08 \), therefore for all initial conditions in the set \( \mathcal{C}_0 \) given by (36) with

\[
\begin{align*}
\varphi_{31} &= \arccos \left( \frac{l_3^2 - l_2^2 + \rho^2 d^2 + y^2}{2l_3 \sqrt{\rho^2 d^2 + y^2}} \right) \\
\varphi_{32} &= \arccos \left( \frac{l_3^2 - l_4^2 + (1 - \rho)^2 d^2 + y^2}{2l_3 \sqrt{(1 - \rho)^2 d^2 + y^2}} \right) \\
\varphi_{41} &= \arccos \left( \frac{l_2^2 + l_4^2 - \rho^2 d^2 - y^2}{2l_2 l_4} \right) \\
\varphi_{42} &= \arccos \left( \frac{l_2^2 + l_4^2 - (1 - \rho)^2 d^2 - y^2}{2l_2 l_4} \right)
\end{align*}
\]

This example shows clearly the need to the multi-step stability analysis tool developed in Proposition 1, since for \( k_0 < 2 \), stability cannot be claimed.

Remark 3. To give some concrete idea about how large is the region of attraction, note that the set of initial conditions \( \mathcal{M}_e \) leading to convergence to the neighborhood

\[
\mathcal{M}_e := \left\{ \mathcal{F}_1 \ | \ ||\mathcal{F}_1 - \mathcal{F}_f||^2_Q \leq 0.08 \right\}
\]

which is a neighborhood of the desired limit cycle of length 2 defined by \( \mathcal{F}_f = 0 \).

\[
\mathcal{M}_s := \left\{ \mathcal{F}_1 \ | \ ||\mathcal{F}_1 - \mathcal{F}_f||^2_Q \leq 0.56 \right\}
\]

the closed-loop trajectories impacts on the Poincaré section converge to the invariant and attractive set \( \mathcal{C}_1 \) given by (39) with

\[
\mathcal{M}_s := \left\{ \mathcal{F}_1 \ | \ ||\mathcal{F}_1 - \mathcal{F}_f||^2_Q \leq 0.08 \right\}
\]

This example shows clearly the need to the multi-step stability analysis tool developed in Proposition 1, since for \( k_0 < 2 \), stability cannot be claimed.

![Fig. 11. (Sim 1) A stability analysis tool: the curve \( \psi(r, N_c, k_0) \) for different values of \( k_0 \).](image)

![Fig. 12. (Sim 1) Evaluation of computation time.](image)

**Table 2**

<table>
<thead>
<tr>
<th>Significance</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_r )</td>
<td>0.75 s</td>
</tr>
<tr>
<td>( y )</td>
<td>0.775</td>
</tr>
<tr>
<td>( d )</td>
<td>0.3 m</td>
</tr>
<tr>
<td>( \rho )</td>
<td>0.5</td>
</tr>
<tr>
<td>( v_p )</td>
<td>-0.25 m/s</td>
</tr>
<tr>
<td>( x_10 )</td>
<td>(0,0)</td>
</tr>
<tr>
<td>( Q )</td>
<td>\begin{pmatrix} 1 &amp; 0 \ 0 &amp; 0 \end{pmatrix}</td>
</tr>
<tr>
<td>( v_{\text{mean}} )</td>
<td>-0.4 m/s</td>
</tr>
</tbody>
</table>
of the limit cycle corresponds, among others to the following two initial conditions:
\[
(q_1, \dot{q}_1) = (\pm 42.87\degree, 0\degree/s)
\]

5.2.2. More simulation results

The behavior of the closed-loop system is to be illustrated through the following simulation results. In Fig. 13 the phase portrait \((q_1 - \dot{q}_1)\) [16] of the unactuated coordinate (torso) is displayed, where we note the convergence to a neighborhood of a limit cycle of length 2, which confirm the stability result discussed above. The mean walking velocity evolution is shown in Fig. 14 where the transition from a rest to the desired mean velocity stable walk can be observed. Note that the mean velocity is computed as the ratio between \(\Delta x(k)\) and \(t_f\) where \(\Delta x(k) = x(t_k) - x(t_{k-1})\).

The position and velocity of the torso coordinate is shown in Fig. 15 where we note through its trajectory that it remains close to the vertical. The cartesian coordinates (and their corresponding velocities) of the hips are depicted in Fig. 17 (for the \(x\) coordinate) and in Fig. 18 (for the \(y\) coordinate), furthermore the resulting trajectory of the hips in the plane \(x - y\) is illustrated in Fig. 16.

The system control inputs (i.e. joint torques) to be applied to the actuated joints are depicted in Fig. 19 for both femurs, and in Fig. 20 for both tibias. We note that RABBIT is equipped with dc motors of a maximum torque of 150 N m, therefore according to the figures of the generated torques we conclude that this bound is largely satisfied, but it should also be checked that the power requirement remain within the admissible limit.

To check the admissibility of the actuators required power, the idea is to plot the angular velocity of the actuators versus their absolute torques, and check if the obtained curves remain within the admissible region given by the manufacturer of the actuators (DC motors). If it is
the case it could be concluded that required actuators power is admissible. To compute the velocities of the motors and their absolute torques, based on articular velocities and the motors gear ratio which is of 50, the following formulas are used:

\[
\begin{align*}
    v_{\text{mot}} \text{ [rpm]} &= \frac{v_{\text{art}} \times 50 \times 60}{2 \times \pi} \\
    \tau_{\text{mot}} \text{ [N m]} &= \frac{\tau_{\text{art}}}{50}
\end{align*}
\]  

(45)

where \(v_{\text{mot}}\) [rpm] is the velocity of the motor shaft, \(v_{\text{art}}\) [rad/s] is the relative velocity between the two adjacent links of the concerned articulation, \(\tau_{\text{mot}}\) [N m] is the motor torque, and \(\tau_{\text{art}}\) [N m] is the torque applied on the links.

The application of this verification technique is illustrated on Fig. 21, which depicts the shaft speed versus torque.

Fig. 17. (Sim 1) The x position and velocity versus time.

Fig. 18. (Sim 1) The y position and velocity versus time.

Fig. 19. (Sim 1) The torques of the femurs versus time.

Fig. 20. (Sim 1) The torques of the tibias versus time.

Fig. 21. (Sim 1) Absolute value of actuator angular velocities (revolutions per minute) versus absolute value of actuator torques [N m], and admissible region.
que, for the four robot actuators, where it could be clearly seen that the actuators power requirement is admissible. The contact foot interaction forces with ground are plotted versus time in Fig. 22, where we note that the condition of the friction Coulomb’s law is largely satisfied (this is clearly seen through the amount of the ratio $\frac{\lambda_t}{\lambda_n}$ with respect to the friction coefficient which is $\mu_0 = 0.7$). Fig. 23 illustrates the movement of the robot by means of a set of walking stick figures (for the three first steps).

### 5.2.3. Computation time evaluation

In order to evaluate the computation time of the proposed control scheme, let us consider biped walking for 50 steps with a constant speed. The evaluation of the computing time is displayed in Fig. 12, which represents the evolution of the computing time versus cycles (steps). The maximum value is given by $t_{\text{max}} = 0.97$ s. For real time implementation the on-line optimization is replaced by an interpolation procedure. The basic idea is to define a grid on the space $(q_1, q_1)$, and for all the points the optimization problem is resolved off-line to define the corresponding optimization parameter $p$, so that at the end of the procedure a look-up table is obtained. In the experiments this look-up table is used to find, for the chosen configuration, the optimization parameter at each sample time knowing the initial condition (position and velocity) on the unactuated coordinate.

### 5.3. Simulation 2: transition between different mean walking velocities

In this simulation, it is shown that the proposed feedback enables transitions between different mean walking velocities to be easily obtained. To show this, 46 walking cycles have been produced during which different desired velocities of 0.24 m/s, 0.3 m/s and 0.40 m/s are successively applied during 12, 14 and 20 cycles respectively. Because of the proportional dependency between the duration of the cycle $t_f$ and the mean walking speed we have chosen to change $t_f$ under constant $d = 0.3$ m in order to increase (go faster), or to decrease (go slower) the walking speed. Since the step length is 0.3 m, the choice of the cycle end-time corresponding to the yet mentioned speeds (0.24, 0.3, and 0.40 m/s respectively) is (1.25, 1, and 0.75 s respectively).

Fig. 24 shows the phase portrait of the unactuated coordinate (torso), where it is well shown the transition between the different stable limit cycles (each limit cycle is relative to a walking speed). In Fig. 25 the mean walking speed is plotted, showing thus the switching between the different proposed walking speeds. The behavior of the unactuated coordinate (torso) is illustrated in Fig. 26 giving its position (horizontal, respectively vertical) of the hips are depicted.
respectively in Figs. 28 and 29, and a more illustrative plot of the hip’s trajectory in the sagittal plane is given in Fig. 27.

The generated torques are plotted in Figs. 32 and 33 for the femurs and tibias respectively, while in Fig. 30 it is checked that power requirement remain within the permitted limit. The contact forces with ground of the stance leg foot are depicted in Fig. 31, where we see clearly that the robot keeps contact with ground during walking.

5.4. Simulation 3: robustness against parameters uncertainty

In order to investigate the robustness of the proposed controller, let us introduce parameter uncertainties. The inertias of the robot links, namely $I_1$ (the torso), $I_3$ (the femur) and $I_4$ (the tibia) cf. Table 1, are considered with an uncertainty of 10% of their nominal values, that is...
I_iu = I_i + \Delta I_i; \quad I_{iu} = I_3 + \Delta I_3; \quad I_{iu} = I_4 + \Delta I_4

where the uncertainties \( \Delta I_i = 0.1I_i \) for \( i \in \{1,3,4\} \). Figs. 34–39 present the corresponding simulation results over 12 walking steps.

In Fig. 34 the positions and velocities of the first leg femur are plotted for the nominal system (solid line), as well as for the uncertain system (dashed line). Whereas in Fig. 35, the positions and velocities of the tibia are plotted. It can be seen clearly that the introduced uncertainty affects more the femur coordinates.

The behavior of the unactuated coordinate (torso) is represented in Fig. 36 which plots the evolution of its position and velocity versus time. A convergence to a new stable cyclic trajectory is observed. This fact can be seen also on the phase portrait of Fig. 37, where a convergence to a neighborhood of a new limit cycle (different from that of the nominal system) of length 2 is observed for the uncertain system.

In Figs. 38 and 39 the control inputs of the robot are plotted, they represent the torques generated by the proposed controller for the femurs articulations (Fig. 38) and for the tibias articulations (Fig. 39). For both figures the uncertain system torques are slightly different from those of the nominal system.

Let us now consider another test of the robustness of the proposed control approach. This time consider an uncertainty of 15% on the mass of the unactuated coordinate (torso), that is

\[ m_{iu} = m_1 + \Delta m_1; \quad \Delta m_1 = 15\% \]

To see the effect of the introduced uncertainty on the closed-loop system two figures are given. On Fig. 40 the
Fig. 34. (Sim 3) Position and velocity of the femur of the first leg.

Fig. 35. (Sim 3) Position and velocity of the tibia of the first leg.

Fig. 36. (Sim 3) Position and velocity of the torso.

Fig. 37. (Sim 3) Phase portrait of the torso.

Fig. 38. (Sim 3) Torques of the femurs.

Fig. 39. (Sim 3) Torques of the tibias.
According to the robot configuration, namely the step length \( d = 0.3 \) m, the robot hits the stair during the fifth walking step.

The approach parameters are the same as previous simulations except the weighting matrix in the optimization criterion which is chosen \( Q = \text{Diag}(1,0.1) \), and the step duration which is of \( t_f = 1 \) s now. This change in these two parameters have been adopted because it gives a better results, namely a configuration with these parameters choice is more robust than the configuration with the old values.

The corresponding simulation results are depicted in Figs. 42–47. It is worth to note that the impact is detected at the instants \( t = 1 \) s for all steps, except for the step during which it hits the stair (fifth step), where the impact instant corresponding to this last one is \( t = 0.935 \) s, which is before the expected time \( t = 1 \) s.

5.5. Simulation 4: robustness against ground irregularities

The aim is to investigate the robustness of the proposed controller, against ground irregularities. Let us make the robot walking on a horizontal surface with a stair at a distance of 1.4 m from the robot, the stair is 1 cm height.

Fig. 40. (Sim 3) Position and velocity of the torso.

Fig. 41. (Sim 3) Phase portrait of the torso.

Fig. 42. (Sim 4) Position and velocity of the robot femurs.

Fig. 43. (Sim 4) Position and velocity of the robot tibias.
Figs. 42 and 43 display the positions and velocities of the actuated robot’s limbs versus time. The effect of the unexpected impact could be seen on Fig. 42 as a removal from the cyclic trajectory followed by a convergence to the same trajectory. This effect can also be observed on the behavior of the unactuated coordinate (torso). Especially Fig. 44 which shows the evolution of the angular position and velocity of the torso, and Fig. 45 which displays its phase portrait. On both figures a removal from the cyclic trajectory is observed, followed by a convergence to the same trajectory. It is worth to note that the limit cycle is of length $l$ for this simulation. Fig. 46 depicts the hips trajectory in the sagittal plane, whereas Fig. 47 illustrates through a stick figures of the robot postures the climbing of the stair.

The limit of stair height beyond it the robot falls is closely related to the chosen configuration. For the actual configuration the limit is of 5 cm. Nevertheless this limit could be increased by changing the configuration, or the approach parameters.

6. Conclusion and future work

In this paper, a nonlinear low dimensional predictive control approach is proposed for the control of RABBIT, a walking five-link, seven d.o.f. under-actuated biped robot. The basic idea of the approach is to split up the vector of coordinates into actuated and unactuated variables. Then on-line optimization is used to update reference trajectories on the actuated coordinates, to aim to enhance the behavior as well as the stability of the unactuated variables.

The stability analysis of the resulting closed-loop system is carried out using a graphical tool based on the Poincaré section. Sufficient conditions for the stability of the motion are proposed and a concrete computation procedure is given to estimate the corresponding region of attraction related to the zero-dynamics of the closed-loop system. The particular case of scalar predictive control is success-
fully investigated by simulation and a reasonable regions of attraction are obtained.

The resulting feedback seems to be real-time implementable thanks to the low dimension of the optimization problem.

The whole framework is illustrated through simulation case studies. Indeed four simulations are proposed. In the first one walking at constant average speed starting from rest is investigated, while the second scenario concerns switching between different walking speeds. Robustness of the proposed nonlinear predictive based-upon controller is verified through the two last applications. In the first one a robot model including parameters uncertainties (namely uncertainties on inertias and masses) is considered, while in the second, ground irregularities are considered. In spite of these both significant disturbances the controller is able to guide the robot suitably for walking while keeping it stable (no slipping, no falling).

In simulations, after each impact the reference trajectories on actuated coordinates are computed based on a predefined step frequency. These trajectories are tracked while checking at each decision instant if there is impact (RABBIT prototype is equipped with switches at feet used to detect impacts). If an impact is detected, the two legs are relabelled, the configuration of the robot is measured and a new step starts up.

Future works may include other features, that should be deeply investigated. In particular, one may be able to choose the design parameters $t_0$, $z_1$ and $Q$ in order to optimize some desired feature (mean energy, torque, robustness). However, the key future work is naturally the experimentations. This is currently in progress.

References

[6] Chang T, Hurmuzlu Y. Sliding control without reaching phase and its checking at each decision instant if there is impact (RABBIT prototype is equipped with switches at feet used to detect impacts). If an impact is detected, the two legs are relabelled, the configuration of the robot is measured and a new step starts up.