Nonlinear Model Predictive Running Control of Kangaroo Robot: a One-Leg Planar Underactuated Hopping Robot

Nicolas Carlési, Ahmed Chemori

▶ To cite this version:

HAL Id: lirmm-00544726

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00544726v1

Submitted on 8 Dec 2010

Nonlinear Model Predictive Running Control of Kangaroo Robot: a One-Leg Planar Underactuated Hopping Robot

Nicolas Carlési and Ahmed Chemori

Abstract— The control of dynamically stable hopping robots has made great progress in the last decades. This paper deals with modeling and control of *Kangaroo* hopping robot. It is a one-leg planar hopping robot which imitates the mode of displacement of kangaroos. Lagrangian dynamic model of the hopping robot is computed on the different phases of the jumping cycle. A new control scheme is proposed to control the leg thrust during stance phase for planar hopping. A nonlinear model predictive control has been combined with Raibert's approach which has improved significantly its performances. A simulator has been developed to simulate the behavior of the controlled robot. The proposed control approach is validated in simulation and is compared with Raibert's approach.

I. RELATED WORKS

Hopping robots have been widely studied during last decades. Usually made of a body and a compliant leg, these systems are capable of moving in one, two or three dimensions [1] alternating phases of ground contact and flight. Consequently, they can be regarded as *variable structure systems* [2], so that mechanical constraints can vary as the contact condition changes.

Some studies were conducted on passive movement of planar hopping robots. These are robots that have a compliance at the leg and hip. The choice of adequate initial conditions and mechanical parameters enable these passive systems to move on a slope under the effect of gravity [3]. However, their mobility is reduced because of friction that provoke energy loss in the system [4]. By adding actuators, some controls were specifically designed to compensate the effects of friction [5],[6].

One of the pioneers of hopping robots control is M. Raibert. He was at the origin of most current controls in the field of the hopping robots. In [1], the control of a planar one-legged hopping robot is treated as three separate control problems: the thrust for jumping, the forward speed and the body attitude.

Indeed, during the last two decades, different control approaches have been proposed to deal with these three issues. The control of the thrust of the leg [7] has been the subject of study of vertical hopping robots. At first, open-loop control have been developed to achieve a constant hopping height [8]. Then, research has focused on closed loop control. In [2], authors show that the height of jump converges to a stable limit cycle by regulating the total energy (Hamiltonian) of the system. In [9] the authors have determined the vertical velocity of the robot body needed at take-off instant to reach a specified apex height. The forward speed control of hopping robot has been particularly studied. To deal with this problem, a SLIP (Spring Loaded Inverted Pendulum) model with a point body mass and a leg is generally used. The proposed approaches consist in computing of the position of the foot at touchdown which enables to accelerate or to slow down the robot during the following stance phase [1]. In [10], an analytical study of the stability of the forward velocity equilibrium behavior has been conducted. It allows to formulate a new law to control the foot position at touchdown. In [11], a polynomial expression of the angular leg trajectory is used as reference for an inverse dynamic controller during flight phase. Inverse dynamic control has also been adopted in [6] for forward speed control as well as for body attitude control during stance phase [6] [11]. A. Sato and his coworkers have proposed a new control approach using a SLIP model composed a single actuator at the hip's level. The proposed control approach allows generation of stable periodic motions in simulation [12] as well as in real-time experiments [13].

The above listed control approaches assume that the ground is continuous and flat. However, it is generally not the case when the robot moves in real environment where the ground can be irregular. Therefore a trajectory planning will be necessary. In [14], a sequence planner is proposed, it aims to find sequences of foot placements on a flat artificial ground with holes. Recently reactive planning techniques have been proposed in [15].

Other recent research works have been particularly focused on the biomimetism. The physiological characteristics of animals have directly been imitated in insect hopping robots [16] or mammals [17].

This paper is organized as follows. Section II describes the mechanical structure and the dynamic model of *Kangaroo* hopping robot during the different phases of the jumping cycle. In section III, a nonlinear model predictive control is combined with Raibert's control approach for one-leg planar hopping robot [1]. Section IV presents the obtained simulation results of the proposed nonlinear model predictive running control approach compared with Raibert's control approach. Finally, conclusion and future work are made in section V.

II. KANGAROO HOPPING ROBOT DESIGN AND MODELING

A. Jump Cycle Description

Hopping robots are variable structure hybrid underactuated systems [18]. Control of jump requires a dynamic modeling

N. Carlési and A. Chemori are with LIRMM, Univ. Montpellier 2 - CNRS, 161 rue Ada, 34392 Montpellier, France `carlesi(chemori)@lirmm.fr`

of the robot for the different phases of the jump cycle (cf. Fig. 1). The number of degrees of freedom of the system changes from one phase to another.
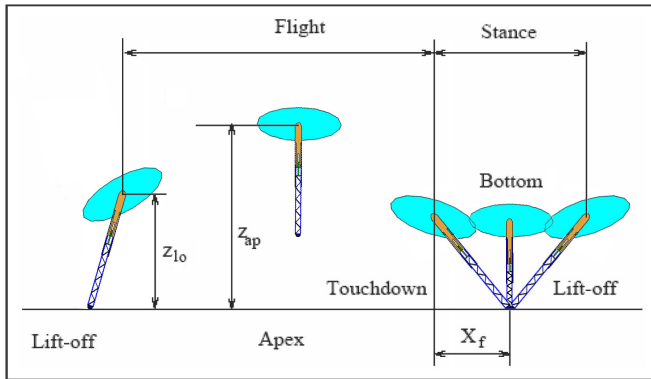

Fig. 1.   Decomposition of a jump cycle

The jump cycle can be decomposed into two main phases and two transitions as illustrated in Fig. 1:

Lift-off transition: it characterizes the switch between the stance phase and the flight phase. At that moment, the impulse of the leg during the contact phase gives enough energy to throw up the robot in flight phase.

Flight phase: a main phase in which the robot is in the air and describes a ballistic movement. It is characterized by the instant when the robot reaches the maximum height which is called Apex.

Touchdown transition: it characterized by the switch between the flight phase and the stance phase.

Stance phase: a main phase in which the robot is in contact with the ground and follows a swing motion. The moment marked by a vertical position of the leg is called Bottom.

B. Kangaroo Hopping Robot Design

Consider *Kangaroo* robot illustrated in Fig. 2. Its a one leg planar hopping robot. It's mechanical structure (cf. Fig. 2) has the following five degrees of freedom:

x, z : two degrees of freedom for the cartesian position of the body in the XOZ sagittal plane

l : one degree of freedom for the length of the leg

ϕ : one degree of freedom for the angle of the body in the sagittal plane

υ : one degree of freedom for the angle of the leg in the sagittal plane

The system includes three bodies, they are considered as three point masses located at their respective center of gravity: m_t for the body of the robot, m_l for the leg and m_u for the foot. The center of gravity of the body m_t is located at a distance d_e from the hip's pivot and forms an angle α with the longitudinal axis of the robot's body (cf. Fig 2). The center of mass of the leg m_l is located at a distance d from the hip of the robot. The first extremity of the leg
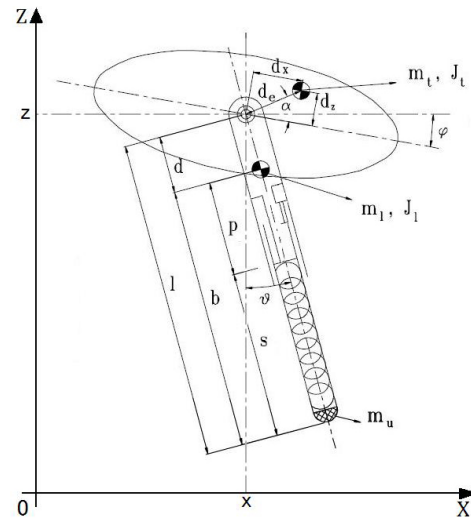

Fig. 2.   *Kangaroo* Robot's mechanical structure [4]

spring is fixed to the leg upper part, at a distance p from the center of gravity of the leg. The other one is fixed at the foot of the robot, the tube containing the spring slides freely in the upper part of the leg. The spring is characterized by a stiffness coefficient kl and a length s (s_0 at rest). The length l represents the total length of the leg (l_0 at rest) and define the distance from the center of gravity of the foot m_u to the hip. The moments of inertia of the body and the leg are denoted respectively by J_l and J_t. The modeled viscous frictions are located respectively at the hip c_h and at the leg c_l. For more details about the dynamic parameters presented above, the reader is referred to [4],[1]. The robot is equipped with two actuators, an angular actuator at the hip which acts on the body's orientation and a linear actuator in the leg acting on the elongation and contraction of the leg.

C. Dynamic Modeling of the Hopping Robot

The *Kangaroo* robot has five degrees of freedom, the vector of generalized coordinates which enables to describe the system is given by $q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 \end{bmatrix}^T = \begin{bmatrix} x & z & l & \phi & \upsilon \end{bmatrix}^T$. In the following, the robot is modeled on the two main phases of the jump cycle.

1) Flight Phase:

During the flight phase, the robot is in a ballistic movement and isn't subject to any constraint. The Lagrangian dynamic model [19] of the robot is written as:

$$M(q)\ddot{q} + N(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) = Su \qquad (1)$$

where M is the inertia matrix, N is the matrix of Coriolis and centrifugal terms, F is the friction vector, G is the gravity vector, S is a matrix of dimensions 5x2 that distributes the control inputs on the generalized coordinates, u is the control input vector.

2) Stance Phase:

The dynamic model during the stance phase is obtained from the one of the flight phase by adding constraints of

contact with the ground. These constraints introduce an additional term representing contact forces with the ground. Assuming that friction between the foot and the ground is sufficient to prevent any sliding. The contact constraints can be expressed as follows:

$$\begin{cases} x_f = q_1 + q_3 \sin(q_5) = constant \\ z_f = q_2 - q_3 \cos(q_5) = 0 \end{cases} \quad (2)$$

where (x_f, z_f) are the Cartesian coordinates of the robot's foot. These constraints are derived twice to be integrated with the dynamics (1). The resulting dynamic model of the stance phase is then given by:

$$\begin{cases} M(q)\ddot{q} + N(q,\dot{q})\dot{q} + F(q) + G(q) = U + J(q)^T\lambda \\ J(q)\ddot{q} + P(q,\dot{q}) = 0 \end{cases} \quad (3)$$

where $J(q)$ is the Jacobian matrix of the contact constraints with the ground, $P(q,\dot{q}) = \dot{J}(q,\dot{q})\dot{q}$ and λ is the vector of Lagrange multipliers related to the contact forces.

3) Transitions between main phases:

During the stance phase, the vertical component of contact force exerted on the ground λ_2 can only be positive. We are in the case of a unilateral constraint, that is to maintain contact with the ground, the hopping robot has to exert only positive force at the contact and the ground don't retain the robot's foot if this last one lift-off. The instant of transition from one phase to another is detected by analyzing the vertical component of the force generated by the constraints. When $\lambda_2 \leq 0$, the robot lift-off and enters in flight phase.

The transition from flying to the contact phase is validated when the robot's foot touches the ground that is $z - l\cos(\nu) \leq 0$. Other conditions on velocity are added to overcome eventual numerical problems (eg. chattering). The robot must be respectively in ascending phase to lift-off and in descendant phase to touchdown. This prevents inadvertent crossing between the main phases.

III. ROBOT RUNNING CONTROL

This section addresses the problem of control for planar moving of *Kangaroo* hopping robot. Firstly, the control approach developed by Raibert for one-leg hopping robot is described. Then, a nonlinear model predictive control approach is proposed to control the leg thrust and combined with Raibert's controller.

A. Raibert's Controller

In [1], Raibert proposed a control system including the three following controllers.

1) Thrust Control:

The basic principle of this controller is to apply, by means of the linear actuator, a constant force F_l on the leg during its extension in stance phase:

$$F_l = constant \quad , \quad \text{if} \quad \dot{l} = \frac{dl}{dt} > 0 \quad (4)$$

This control enables to compensate energy losses due to friction by injecting the energy during the stance phase and to control the height of the jump.

2) Forward Speed Control:

The hopping robot is an underactuated system, therefore it is not possible to control directly the horizontal movement of the robot while controlling the thrust and the body attitude. The position of the foot, when it touches the ground at the end of the flight phase, has a crucial impact on the evolution of the passive part of the robot during the stance phase. Thus, there is a neutral foot position x_{f0} in relation to the hip position at the touchdown which provides unaccelerated travel for the body of the robot during the stance phase. To accelerate/decelerate the machine, the control system has to displace the foot from its neutral position. Acceleration is needed to stabilize the forward speed against errors and external disturbances and to switch from one forward speed to another. The control system uses a linear function of the error in forward speed to find a displacement of the foot from hip at touchdown:

$$x_f = x_{f0} + k_{\dot{x}}(\dot{x} - \dot{x}_d) \quad \text{with} \quad x_{f0} = \frac{\dot{x}T_s}{2} \quad (5)$$

where $k_{\dot{x}}$ is a feedback gain and $\dot{x}$, $\dot{x}_d$ are respectively the forward speed and the desired forward speed. The duration of the stance phase T_s (it depends on the leg stiffness) allows to compute the neutral foot position x_{f0}.

3) Body Attitude Control:

The robot's body can be positioned only during the contact phase. The contact between the ground and the foot of the robot in this phase enables the angular actuator of the hip to control the angle of the body without causing high accelerations on the leg. The control system enables to control the hip's torque Γ thanks to a conventional PD controller and allows the body to follow a desired path:

$$\Gamma = k_p(\phi_d - \phi) - k_v(\dot{\phi}) \quad (6)$$

where ϕ, ϕ_d are the body angle and the desired body angle relative to horizontal axis and k_p, k_v are respectively the proportional and derivative gains of the controller.

Remark 1: For more details about the Raibert's controllers presented above, the reader is referred to [1], Chapter 2.

B. Nonlinear Model Predictive Running Controller

In this section our proposed control scheme is introduced, it consists of an improved Raibert's control. That is, the forward speed and body attitude controllers are those of Raibert, however, a nonlinear model predictive controller [20] is proposed to command the thrust exerted by the leg.

1) Ballistic trajectory:

The ballistic trajectory is determined during lift-off. The force of gravity exerts continuously an action on the vertical acceleration of the robot. Therefore, the horizontal component of the velocity of the robot's body $\dot{x}$ does not vary

during the flight phase, and should correspond to the desired speed of movement of the robot at lift-off $\dot{x}_{lo} = \dot{x}_d$. In [9], the vertical velocity of the robot's body needed at the lift-off $\dot{z}_{lo}$ to jump to a desired height z_{apd} has been computed:

$$\dot{z}_{lo} = \sqrt{2g(z_{apd} - z_{lo})} \qquad (7)$$

where z_{lo} is the body height at lift-off and g is the acceleration of gravity. To follow the vertical and horizontal components of velocity, the corresponding angle and speed of the leg has to be identified. By deriving the coordinates of the center of mass of the robot's body at lift-off, the following set of nonlinear equations is obtained and used to identify the desired angle υ_d and the leg's speed $l\dot{\upsilon}$.

$$\begin{cases} \dot{x}_{lo} = -\dot{l}_d \sin(\upsilon_d) - l\dot{\upsilon}\cos(\upsilon_d) \\ \dot{z}_{lo} = \dot{l}_d \cos(\upsilon_d) - l\dot{\upsilon}\sin(\upsilon_d) \end{cases} \qquad (8)$$

2) Proposed control approach:

When the robot is in contact with the ground and the leg begins relating, the nonlinear model predictive control calculates a control sequence to minimize a performance criterion J subject to constraints:

$$\min_{u} \quad J = \sum_{k=1}^{Np} Q_1(l_0(k) - l(k))^2 + Q_2(\dot{x}_{lo}(k) - \dot{x}(k))^2 \\ + Q_3(\dot{z}_{lo}(k) - \dot{z}(k))^2 \qquad (9)$$

$$under \quad |u_i| < u_{max} \quad for \ all \quad i \in \{1, ..., N_c\}$$

where $l(k)$, $\dot{x}(k)$ and $\dot{y}(k)$ are respectively the length of the leg, the horizontal speed and the vertical speed predicted along the prediction horizon Np. This optimization criterion is used to follow horizontal speed $\dot{x}_{lo}$ and vertical speed $\dot{z}_{lo}$ corresponding to the desired ballistic trajectory (cf. section III-B.1). Therefore, the height of the jump as well as the speed of the robot are controlled during flight. In order to limit spending energy, the difference between the length of the leg l at lift-off and this length at rest must be also minimized. The weights Q_1, Q_2 and Q_3 to give more/less importance to the minimization of their corresponding terms. u_i is the control sequence computed along the control horizon Nc and u_{max} is the constraint on the control input.

The optimization problem is nonlinear and constrained, it is impossible to find an analytical solution. Therefore, the computer must solve numerically the optimization problem at each sampling time. It must determine a control sequence of size Nc, which minimizes the criterion J on a prediction horizon Np taking into account the eventual constraints. The control horizon is generally less than or equal to the prediction horizon [20]. At the end of the control horizon, the control input is kept at zero until the end of the prediction horizon.

From the control sequence computed along the prediction horizon, only the first sample is really applied to the system. The current state of the system is measured, the horizon is shifted of one sampling period and the same optimization

problem is repeated considering the new measured state as initial state. This operation (called technique of receding horizon) is repeated at each sampling instant and allows to increase the control robustness towards the modeling errors/uncertainties and the inevitable perturbations that may affect the system.

IV. SIMULATION RESULTS

A. Simulation Environment

A simulator *Kangaroo* robot was developed using the Graphical User Interface of Matlab software. The developed graphical interface of our simulator is shown in Fig. 3.
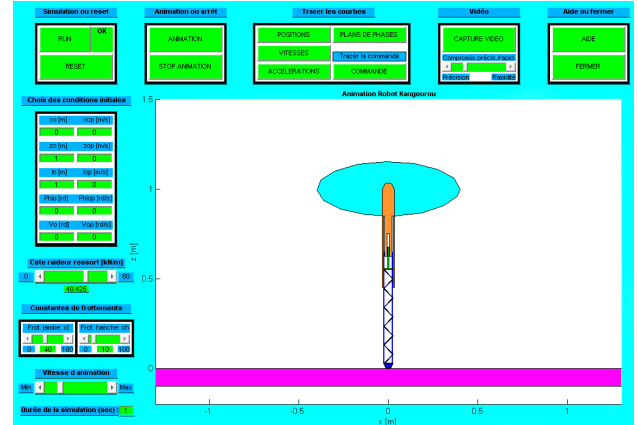

Fig. 3.   Graphical User Interface of *Kangaroo*'s developed simulator

This graphical interface enables to easily manipulate simulations, choose/change initial conditions and parameters of the robot as well as the simulation parameters. It allows also to display the robot movement during simulation.

B. Raibert and Nonlinear Model Predictive Running Control Comparaison

Both Raibert's control and the nonlinear model predictive control are implemented in our simulator. The same modeling parameters are used to compare easily the performance of these two controllers. The different parameters of the Raibert's control (R) and the nonlinear model predictive control (P) are summarized in table I. Simulation results

Parameter	Value	Controller	Parameter	Value	Controller
k_p	1000	R&P	F_l	240 N	R
k_v	50	R&P	Nu	3	P
k_P	200	R&P	Np	6	P
k_V	10	R&P	T_e	0.01 s	P
$k_{\dot{x}}$	0.05	R&P	$Q1$	1	P
$\dot{x}_d$	1 m/s	R&P	$Q2$	1	P
ϕ_d	30 deg	R&P	$Q3$	1	P

TABLE I

PARAMETERS OF RAIBERT AND PREDICTIVE CONTROL

obtained are presented in Fig. 4-7. In Fig. 4 and 7 dash-dots lines correspond to Raibert's control, while solid lines correspond to nonlinear model predictive control.

The *Kangaroo* robot hip is initially positioned at 1.3 *m* in relation to the ground and with a horizontal velocity of 0 *m/s*. The desired forward speed of robot is fixed at 1 *m/s* whereas the desired maximum hopping height of the robot's
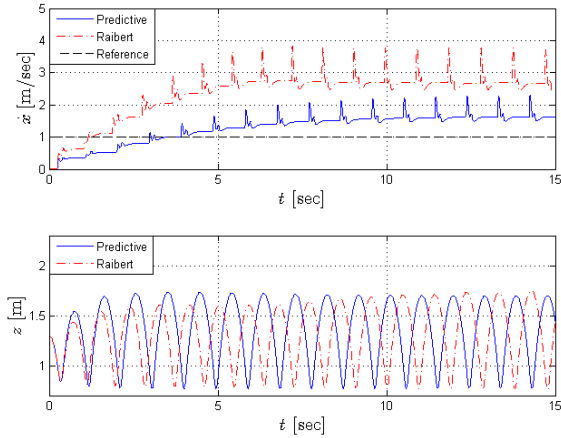
Fig. 4. Evolution of the horizontal speed and the vertical position of the hip for both proposed controllers

Fig. 5. Raibert's control: phase portraits

Fig. 6. Nonlinear model predictive control: phase portraits

hip is 1.8 m. From the evolution of the horizontal velocity (cf. Fig. 4-top) and the vertical displacement (cf. Fig. 4-bottom) it can be noticed that the two schemes enable to control the robot jumps to converge to cyclic trajectories. The desired reference hopping speed is 1 m/s (cf. dashed lines Fig. 4-top). With the assumption that the robot starts movements from rest the initial speed is fixed to zero, then it converges in about 6 s to 2.7 m/s with the Raibert's control and in about 9 s to 1.6 m/s with nonlinear model predictive control. Predictive control converges a little slower than Raibert's control to a steady state, however it gives a more precise tracking. Indeed, the reference on the hopping speed is much better tracked with the proposed predictive control since the error on the speed is reduced to 60 % with respect to 170 % with Raibert's control. For the height of the jump (cf. Fig. 4-bottom), it converges much faster towards the final value with nonlinear model predictive control. The transient regime is about 2 s while it is about 12 s with the Raibert's control. Unlike Raibert's control which uses open loop control to compute the force exerted on the leg, nonlinear model predictive control allows to follow, in closed loop, a reference on the height of the jump. In our simulation, a reference of 1.8 m is fairly well followed, the steady state error is about 6 %.

Phase portraits $(z,\dot{z})$, $(\phi,\dot{\phi})$, $(l,\dot{l})$ and $(v,\dot{v})$ shown in Fig. 5 (for Raibert's control) and Fig. 6 (for nonlinear model predictive control) highlight a faster convergence of the controlled system with nonlinear model predictive control. The generated forces and torques exerted respectively at the leg and the hip are presented in Fig. 7. The open-loop control of Raibert (cf. dash-dots line Fig. 7-top) which controls the thrust of the leg during the stance phase was set to 240 N so that the jumping height converges to 1.8 m. This command does not enable to control the speed of horizontal movement. The thrust of the leg produces a horizontal speed excessive for the flight phase. During this phase, the forward speed controller positions the robot's leg forward to allow the robot
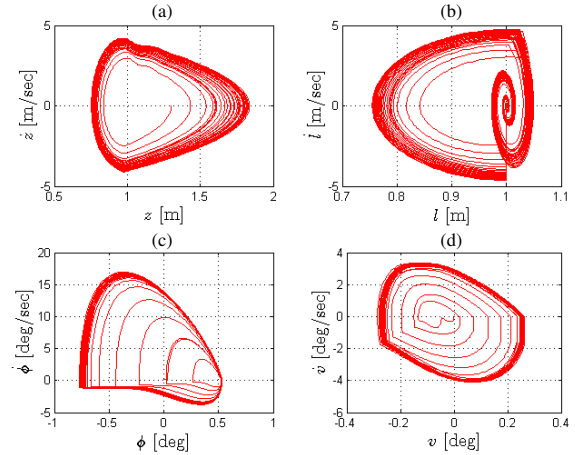
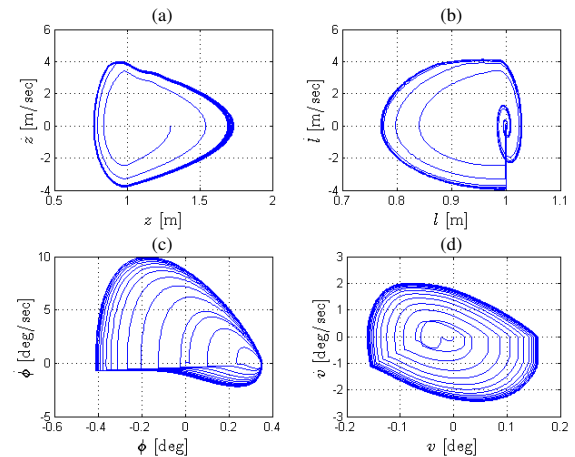to slow down during the following contact phase (cf. section III-A.2). Although the mass and inertia of the body are fairly small compared to those of the leg, the torque exerted at the hip (cf. dash-dots line in Fig. 7-bottom) causes a rotation of the body as wide as the angular clearance of the leg will be important. The speed of the robot is much higher than the reference, the body undergoes an important rotation (cf. Fig. 5.(c)) at touchdown. The control law to reposition the body during the stance phase involves a necessary torque of particular importance at the hip, which can reach 1450 $N.m$.

The nonlinear model predictive control enables to take into account the constraints of the system to control the force exerted on the leg (cf. solid line Fig. 7-top). Here, the control has been deliberately saturated ($U_{max} = 400$ N in (9)). Actually, the system will converge more rapidly towards its steady state as this value is higher. In our case, three or four jumps are necessary to reach the high jump and speed of desired movement. The control of the horizontal speed at lift-off minimizes the angular clearance of the leg and body of the robot during the flight phase (cf. Fig. 6.(c) and 6.(d)). Consequently, the torque exerted to control
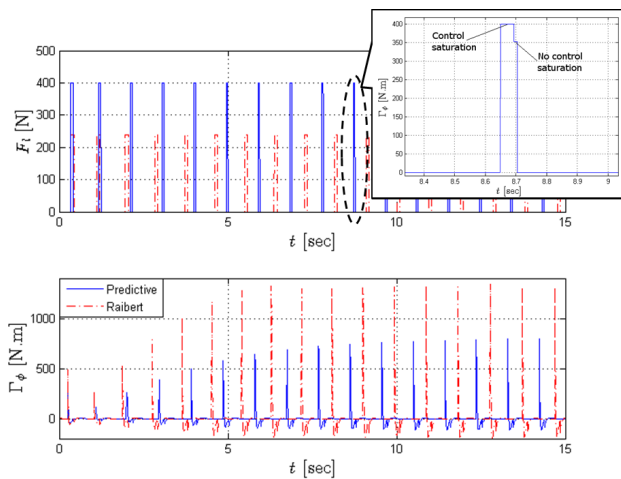
Fig. 7. Evolution of the control inputs versus time

computed and a simulator has been developped. Then a control scheme has been proposed to generate stable periodic motions in the sagittal plane. The proposed control scheme on the whole jumping cycle is derived from Raibert's control approach. To overcome the Raibert's control drawbacks related to the movements speed and jumping height a nonlinear model predictive control has been introduced to improve its performances.

REFERENCES

[1] M. H. Raibert, *Legged robots that balance*. Cambridge, MA, USA: Massachusetts Institute of Technology, 1986.
[2] A. I. M. Ishikawa, M. Neki and J. Hara, "Energy preserving control of a hopping robot based on hybrid port-controlled hamiltonian modeling," in *IEEE Conference on Control Applications*, vol. 2, (Yokohama, Japan), pp. 1136–1141, 2003.
[3] K. Matsuoka, "A mechanical model of repetitive hopping movements," *Biomechanisms*, vol. 5, pp. 251–258, 1979.
[4] H. D. Taghirad, *Analysis, Design, and Control of Hopping Robot*. McGill University, Quebec, Canada: Master report, 1993.
[5] E. P. N. Cherouvim, "Energy saving passive-dynamic gait for a one-legged hopping robot," *Robotica*, vol. 24, no. 4, pp. 491–498, 2006.
[6] M. Ahmadi and M. Buehler, "Stable control of a simulated one-legged running robot with hip and leg compliance," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 1, pp. 96–104, 1997.
[7] M. M. K. G. Naik and J. M. Barden, "Control of a one-legged hopping robot using a hybrid neuro-pd controller," in *IEEE Canadian Conference on Electrical and Computer Engineering*, (Ottawa, Canada), pp. 1530–1533, 2006.
[8] P. G. H. Rad and M.Buehler, "Design, modeling and control of a hopping robot," in *IEEE/RSJ Conference Intelligent Systems and Robots*, vol. 3, (Yokohama, Japan), pp. 1778–1785, 1993.
[9] K. Harbick and G. Sukhatme, "Controlling hopping height of a pneumatic monopod," in *IEEE International Conference on Robotics and Automation*, vol. 4, (Washington, DC , USA), pp. 3998–4003, 2002.
[10] W. Schwind and D. Koditschek, "Control of forward velocity for a simplified planar hopping robot," in *IEEE International Conference on Robotics and Automation*, vol. 1, (Nagoya , Japan), pp. 691–696, 1995.
[11] N. Cherouvim and E. Papadopoulos, "Single actuator control analysis of a planar 3dof hopping robot," in *Robotics: Science and Systems*, (Cambridge, Massachusetts, USA), pp. 145–152, 2005.
[12] A. Sato, "Simulation of one-legged hopping robot with phase plane stability," in *IASTED International Conference: modelling and simulation*, (Montreal, Quebec, Canada), pp. 112–117, 2007.
[13] A. Sato and M. Buehler, "A planar hopping robot with one actuator: design, simulation, and experimental results," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, (Sendai, Japan), pp. 3540–3545, 2004.
[14] G. Zeglin and B. Brown, "Control of a bow leg hopping robot," in *IEEE International Conference on Robotics and Automation*, vol. 1, (Leuven, Belgium), pp. 793–798, 1998.
[15] U. S. Ö. Arslan and O. Morgül, "Reactive footstep planning for a planar spring mass hopper," in *IEEE/RSJ International Conference Intelligent Robots and Systems*, (St. Louis, USA), pp. 160–166, 2009.
[16] A. G. J.-C. Z. M. Kovač, M. Fuchs and D. Floreano, "A miniature 7g jumping robot," in *IEEE International Conference on Robotics and Automation*, (Pasadena, CA, USA), pp. 373–378, 2008.
[17] S. Hyon and T. Mita, "Development of a biologically inspired hopping robot-"kenken"," in *IEEE International Conference on Robotics and Automation*, vol. 4, (Washington, DC, USA), pp. 3984–3991, 2002.
[18] M. Spong, *Underactuated Mechanical Systems*. Springer-Verlag, 1998.
[19] M. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York, USA: John Wiley & Sons, 1989.
[20] E. F. Camacho and C. Bordons, *Model Predictive Control*. London, UK: Springer (2nd Edition), 2007.

the body orientation in the sagittal plane during the stance phase is less important compared to Raibert's controller (cf. Fig. 7-bottom). To show the saturation of the control at U_{max} and the verification of this constraint (imposed in the optimization criterion (9)), a zoom is achieved on the control input computed by the nonlinear model predictive controller (cf. Fig. 7-top).

In order to proceed to a comparison between Raibert's control and nonlinear model predictive control, the mean values of the generated control inputs (force for the linear actuator and torque for the angular actuator) are computed and compared. For Raibert's control scheme, the average of the control force is 24.192 N, and 25.363 N for the nonlinear model predictive control. However, the average values of the torque are respectively 40.069 $N.m$ and 19.954 $N.m$.

It is worth to notice the efficiency of the proposed controller in terms of spending energy w.r.t Raibert's controller. Indeed, it needs a half of energy for the generated torque, with roughly the same energy for the generated force to control the system.

The use of nonlinear model predictive scheme to control the thrust before lift-off has significantly improved the overall behavior of the *Kangaroo* robot. Now it is possible to control the height of jump of the robot, which was impossible with the Raibert's controller. The robot follows more precisely references and converges faster to a stable cyclic movement. Furthermore, this control enables to improve the response of Raibert's forward speed and body attitude control minimizing the torque exerted at the hip. This paper is accompanied with a movie showing the obtained hopping movement of the robot under the proposed control scheme.

V. CONCLUSIONS AND FUTURE WORK

The objective of this work was to design a control approach for the one-leg planar hopping robot *Kangaroo* in order to generate stable periodic motions. To reach this objective, the Lagrangian dynamic model of the hopping robot on the different phases of the jumping cycle has been