A CLOSED-LOOP OPTIMAL CONTROL APPROACH FOR ONLINE CONTROL OF A PLANAR MONOPED HOPPER

SOHA POUYA, MOSTAFA AJALLOOEIAN, AUKE JAN IJSPEERT

Biorobotics Laboratory, Institute of Bioengineering, Ecole Polytechnique Federale de Lausanne (EPFL), CH 1015, Lausanne, Switzerland.

In this paper we present a closed-loop optimal control approach for the online control of a legged robot locomotion, particularly the hopping of a simulated monoped robot. Modeling is done based on the spring loaded inverted pendulum (SLIP) model suggested as the animal and human running gait template. The key idea is to efficiently inject energy to the system so that the monoped can track the desired apex height and forward velocity. The state of the system is observed in the Poincaré section at the apex point and the corresponding discrete dynamics is formulated by using available analytical solutions. The goal is then to synthesize an optimal control law which can bring the apex state at any step to the desired state at the next step. We show the controller performance in providing fast and accurate response in the presence of noise and through different scenarios while minimizing the control effort.

1. Introduction

Modeling and control of legged locomotion has been the field of interest and research from both biology and robotics viewpoints to understand the underlying locomotor mechanisms, as well as take inspiration to build efficient and powerful robotics platforms. Studies have been done at different levels of abstraction from very simplified models to sophisticated neuromechanical models (Ref. 1). For animal and human running gaits, the Spring-Loaded Inverted Pendulum (SLIP) has been proposed as an archetypal model, predicting correctly ground reaction forces and center of mass trajectories (Ref. 2). The SLIP model is a two-state hybrid dynamical system consisting of a point mass attached to a massless springy leg. Control laws can be added to this model to set the leg orientation at touchdown and can extensively improve the running stability (Ref. 3). In comparison to static robotic locomotion, this model presents dynamic and more efficient legged locomotion by exploiting the leg natural dynamics. The role of exploiting compliance in the legged locomotion has long been pointed out (Ref. 4). Raibert and his colleagues have successfully used this idea by using SLIP-based simple control rules for the control of their hopping and running robots (Ref. 5). Saranli et. al. (Ref. 6) proposed a deadbeat control approach using the inverse return map of SLIP model (Ref. 7) and compared the controller performance with the modified Raibert decoupled control laws. Their coupled control approach shows better performance in tracking the desired jumping height and forward velocity while not needing tedious tuning of controller parameters. In this paper, we show how one can use closed-loop optimal control methods for this tracking problem while minimizing the control effort (energy consumption). Open-loop optimal control methods have been used to produce stable and efficient periodic gaits (Refs. 8 and 9). Tedrake (Ref. 10) has suggested an optimal control approach to optimize the region of initial conditions from which the robot can recover. Our focus however is to enable the system to track the input commands in an efficient way. We use the same analytical return map as (Ref. 7) for the later comparison. The controller tracking performance is investigated in the presence of noise and through different scenarios including step, sinusoidal, complex and random commands. We first briefly explain the methods used for return-map modeling and optimal control formulation and finally present the results obtained for the different scenarios.

2. Mathematical Modeling of System Dynamics

The modeling of the leg dynamics is performed in the discrete form by degrading the continuous dynamics to one less dimension of Poincare map. Control of the system state variables is also performed in the discrete fashion. The key idea is to exploit the natural dynamics of the system and only manipulate the behavior by injecting energy at specific moments of the natural motion. This approach stands between the passive dynamic walkers where no actuation is involved and the static walkers such as Asimo with continuous and high accuracy control. Accordingly we base our modeling tools on the Poincare map also called *return map* formulation. Mathematically speaking we collapse the dimensions of the system by transecting the system at one characteristic point resulting a discrete map called Poincare map. The general form of such a system can be stated as follows:

$$x[k+1] = f(x[k], u[k]), \quad y[k] = h(x[k], u[k]) \tag{1}$$

where $x[k] \in \mathbb{R}^n$ is the state of the system at time k (an integer), $u[k] \in \mathbb{R}^p$ is the control input and $y[k] \in \mathbb{R}^q$ is the system output. Functions f and h are the mapping functions which should be extracted for the system. For our system, the characteristic point selected is the *apex* point where the

Table 1: physical properties and the initial values.

Physics		Initial States	
Hip Mass (m)	50.48 kg	$\theta_{lo}[0]$	25°
Gravity (g)	9.81 m/s^2	$E_{lo}[0]$	500 J
Rest Leg Length (l_0)	1 m	$\psi_{lo}[0]$	1
Control Bounds ($[u_{min}, u_{max}]$)		Initial Controls Guess	
$\theta_{td}[i]$	$[\pi/4,\ \pi/2]\ rad$	$\theta_{td}[0]$	20°
$k_c[i]$	$[100, 500]\ N.m^3$	$k_c[0]$	250 $N.m^3$
$k_d[i]$	$[100, 500]\ N.m^3$	$k_d[0]$	250 $N.m^3$

leg reaches to the highest point in its trajectory and its vertical velocity is zero. The state of the system consists of the position and the forward velocity at the apex point. Closed form solutions for this system is not trivial in contrast to its simple dynamical model. We use the analytical solutions presented in (Ref. 7) together with a definition of energy consumption measurement to perform the task. We present the overall formulation and more detailed equations can be found in Ref. 6.

The state of the SLIP model is selected as $\mathcal{X}_l = \{X_l | X_l = [\bar{x},\ \bar{y},\ \bar{v_x}]\}$ with $\bar{x}$ and $\bar{y}$ being the positions and $\bar{v_x}$ the forward velocity of the apex point. State is selected such that it can give an intuitive description of the system behavior including the apex *jumping height* and *forward velocity.* The control variables are selected similar to the Raibert controllers as $\bar{\mathcal{U}} = \{\bar{u} | \bar{u} = [\theta_{td},\ k_c,\ k_d]^T\}$ with θ_{td} being the angle of attack at touchdown and k_c, k_d respectively the compression and decompression spring constants in the stance phase. The spring used for this formulation is an *air spring* with the potential energy formulated as $U(x) = \frac{K}{2}(\frac{1}{x^2} - \frac{1}{x_0^2})$. Closed-form formulation can be extracted in the liftoff coordinate system by transferring the state to the new set of state variables $\mathcal{Z}_{lo} = \{Z_{lo} | Z = [\theta, E, \psi]^T_{lo}\}$ respectively, the angle of attack, energy and ratio of forward to vertical velocity at liftoff. The apex states can then be calculated from the liftoff state variables as:

$$v_{x_{ap}} = \text{sign}(\psi_{lo})\sqrt{\frac{2E_{lo}\psi_{lo}^2}{m(1+\psi_{lo}^2)}}; \quad y_{ap} = \frac{E_{lo}}{mg(1+\psi_{lo}^2)} + l_0 \cos\theta_{lo}$$

We use these equations to describe the system dynamics together with the control laws detailed in the next section to perform the desired tasks.

3. Closed-Loop Optimal Control

The role of the control algorithm is to find the suitable control input $u^* = [\theta_{td}^*,\ k_c^*,\ k_d^*]^T$ which can take the current apex state X_{ap} to the desired state X^*_{ap} while satisfying the problem constraints. Previous works have suggested simple decoupled controllers (Ref. 5) or using the approximated inverse map as the deadbeat control law (Ref. 7). We propose an optimal control approach which provides a coupled control of the state and in addition minimizes the required control effort. Moreover this approach can replace the tedious procedure of computing the inverse return map and particularly makes it more suitable for higher dimensional systems. We show the effect of taking the coupled dynamics of the system as a whole throughout the experiments in the section 4.

The optimal control problem is formulated as presented in the table 2. The goal is to minimize the control effort while satisfying the state and control constraints. This introduces a constrained optimization problem which can be solved by using the nonlinear programming (NLP).We formulate the *control effort* as the quadratic cost of change of the spring stiffness. In other words, we assume there is an operating point for the spring stiffness, k_0, and any change form that would need consuming energy. The cost function therefore is formulated as $J = (k_c[i+1] - k_0)^2 + (k_d[i+1] - k_0)^2$. *Constraints* of the problem are formulated regarding the physical constraints on the state variables (figure 1) and the control constraints due to the actuators saturation. The boundary for the angle of attack is $[45,\ 90]$ and for the range of the air spring stiffness values we roughly rescale the spring potential energy from Ref. 7 resulting in the boundary of $[100,\ 500] N.m^3$. We use the quadratic errors of tracking the desired state trajectories as *nonlinear equality constraints*. The sequential quadratic programming (SQP) solver provided with MATLAB *fmincon* function is used to solve the problem. To be able to embed this optimization problem in the control loop, we use only one step horizon for measuring the cost function. While it works for this simple problem, for more complex problems using a bigger horizon (such as model predictive control methods) is suggested to improve the stability of the solutions. Finally this proposed algorithm uses the measured values of the state variables at each step to find the required control variables for the next step. In other words the control is performed in closed-loop helping the system to overcome the deviations in the presence of noise. We show the performance of the controller with a 10% noise on the sensor data.

4. Results and Discussions

The performance of the proposed control approach is evaluated through different scenarios assuming 10% noise on liftoff angle measurement. Figure 1 shows the results for selected scenarios each row showing one scenario:

Table 2: Pseudo-code for the optimal control formulation

do for each step i
$\quad u[i]^* = \underset{u}{\mathrm{argmin}}\left((k_c[i+1]-k_0)^2 + (k_d[i+1]-k_0)^2\right)$
$\quad\quad u_{min} < u < u_{max}$
$\quad\quad y_{ap}[i+1] - y_{ap}^{desired}[i+1] = 0$
$\quad\quad v_{x_{ap}}[i+1] - v_{x_{ap}}^{desired}[i+1] = 0$
$\quad Z_l[i+1] = f(Z_l[i], \bar{N}.u[i]^*)$
$\quad$ set $u_0[i+1] = u[i]^* + \epsilon$
$\quad$ control the system with $u[i]^*$ and measure state $\mathcal{Z}_{lo}$
until failure or maximum successful steps

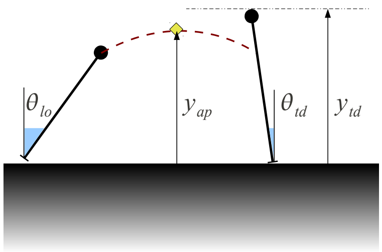

Fig. 1: Constraints imposed on the system state variables including feasibility of the apex height and liftoff angle (physiological constraints) and the system energy.

a-step, b-sine c-complex and d-random command inputs. The desired and controlled state is shown in the left and the control variables in the right graphs. It should be noted that the horizontal axis shows the steps (due to the discrete form of the system). Results show that for the first three cases the monoped is able to track the command accurately and after only a few first steps. The computation time for each step is about 50-100 milliseconds (using MATLAB on an Intel(R) quad-core PC) which is less than the flight duration and enables the online use of this algorithm.

(i) Response to step: As the first experiment we use the simplest scenario by imposing a step command for the forward velocity of the hopper while keeping the altitude fixed. This scenario also can be used for the qualitative comparison of our controller performance with the one from Ref. 7. Accordingly we use the same values including fixed altitude of $1.2m$ and step increase on forward velocity from 0.5 to 2.5 m/s. Figure 2-a shows the resulting behavior for following these commands. Similar to Ref. 7 we use the first 10 steps to bring the system to steady-state forward velocity of 0.5 m/s. At this stage the control problem is how to use the available actu-

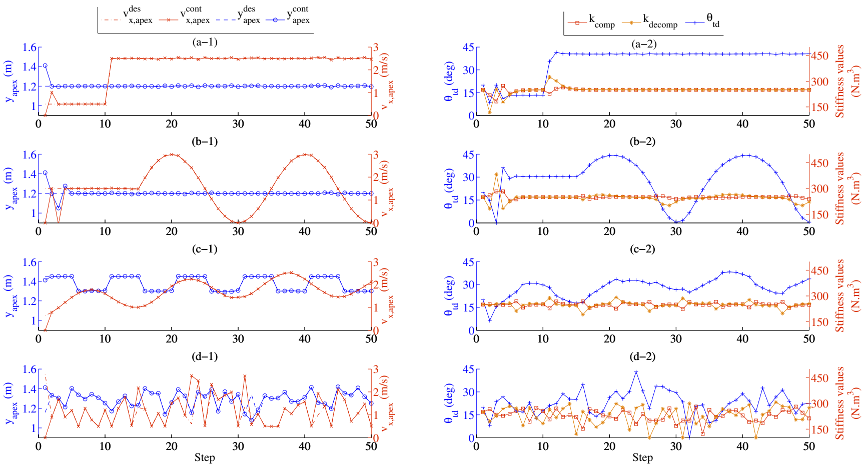

Fig. 2: Results of the designed scenarios (from top to down): *a*-step in forward velocity while fixing apex height, *b*-sinusoidal command for forward velocity while fixing apex height, *c*-complex commands for both and *d*-random commands for both. The desired and controlled state is shown in the left and the control variables in the right graphs.

ators to increase the velocity with 2 m/s in an efficient and responsive way. The nonlinear programming methods embedded in the control loop solve this control problem as an optimization problem and provide the control variables which optimize the control effort at the same time taking both the state and control constraints into account. The resulting control variables are represented in the graph (a-2). As expected the angle of attack plays the main role to increase the forward velocity and the stiffness values tend to keep the minimum-energy values for this gait. However they have an important role to stabilize the gait when the angle of attack is in the transient phase. This coupled dynamics which is neglected in the simplified decoupled control schemes can be exploited to improve the controllability of the hopper. When the angle of attack reaches to the steady state, the stiffness values converges to their rest value to minimize the control effort required for this task. It is worth-noticing that the controller is using only one step horizon to minimize the control effort. Increasing the horizon can result in more optimized control effort by taking to account the consequences of the current control on the N next step.

(ii) Response to sinusoidal trajectory: In the second experiment we use a more difficult scenario by imposing a sinusoidal desired trajectory for the forward velocity and keeping the altitude fixed. Using the same values as Ref. 7, we set the apex height fixed to 1.2 m and change the velocity

in a sinusoidal pattern with amplitude and offset both of 1.5 m/s. This scenario introduces a difficult task for the hopper since it should be able decelerate to zero speed and accelerate to $3m/s$ while still keeping stable hopping to a fixed height. Again the optimal control formulation is used to provide the right feasible control variables at each step. The results for the state and control variables are represented in figure 2, graphs (b-1) and (b-2) respectively. They clearly show the main role of the angle of attack in the speed control and at the same time the complementary role of the air springs in supplying the rest of the required energy to fulfill the task. Angle of attack follows a similar pattern to forward velocity. A closer look at the k_c and k_d patterns shows that while the hopper is speeding up, the stiffness has slightly higher values in the decompression than the compression phase which injects energy to the system ($\Delta E_U = U_{k_d}(r_0) - U_{k_c}(r_0) > 0$), and the opposite occurs while decelerating in order to remove the energy from the system. It should be noted that this task can not be performed successfully by using only the angle of attack taking its constraint into account.

(ii) Response to a complex pattern: In contrast to the two previous experiments, this experiment is designed to not only excite one of the state variables of the system but also to impose a coupled and complex pattern to both velocity and height of apex. As depicted in the figure 2 graph (c-1), forward velocity increases from zero to 2 m/s in a periodic fashion while the apex height value switches every five steps (between 1.3 and 1.45 m/s). This scenario is a good example of the situations where a more sophisticated control law is required to be able to take the coupled dynamics into account. As represented in the graph (c-2) of the figure 2 the angle of attack is not selected in accordance to the forward velocity pattern anymore since its value can derange the desired apex height. However this is still visible that the angle of attack is mostly contributing to forward velocity control and the change of spring stiffness is being used to inject or remove energy respectively in each step increase or decrease in apex height. While this has been already known and used in Raibert like control rules, we propose here a method to take their combinational effect into account.

(iv) Response to a random task: This scenario presents an ultimate situation where both the desired apex height and forward velocity are changing in random fashion. One can imagine it as hopping in an unstructured environment where the hopper should be quick and responsive to adjust its speed and jumping height. Graph (d-1) of figure 2 shows a sample of such desired apex height and speed. The results for this example shows that the hopper is more successful to follow the desired apex height than the veloc-

ity. This is due to the actuator saturation that is assumed for this problem. The motor speed to adjust the angle of attack during one step is less than what is required for some of the speed jumps. It should be noted that the simulated values should be adapted for a new real setup and can drastically change the tracking performance.

5. Conclusion and Future Work

We proposed a closed-loop optimal control formulation for the online control of a single leg hopping gait. This work presents the basic approach and idea on a very simplified model. The approach however is scalable and can be used for higher dimensional models with not only the discrete return map but with their continuous dynamics. We used SQP solvers provided by MATLAB for academical applications while there have been great progress in faster and more robust solvers. We are currently extending this work by using the same approach for more realistic SLIP model (including energy dissipation) and more detailed dynamics modeling for 2D monoped and quadruped robots. In the optimal control formulation we use the horizon with only one step. For future works using more steps can improve the robustness of the solutions. We also plan to test the simulated algorithms on our newly designed prismatic leg.

Acknowledgement

This project has received funding from EPFL and from the European Community's Seventh Framework Programme FP7/2007-2013 Future Emerging Technologies, Embodied Intelligence, under the grant agreement No 231688 (Locomorph) and Challenge 2 Cognitive Systems, Interaction, Robotics under grant agreement No 248311 (AMARSi).

References

1. P. Holmes, R.J. Full, D. Koditschek, J. Guckenheimer, *SIAM Rev.* **48**, (2006).
2. R. Blickhan *J. of Biomechanics*, **22**, 1217–1227 (1989)
3. A. Seyfarth, Hartmut Geyer and Hugh Herr, *J. Exp. Biol.*, **206**, (2003).
4. T. A. McMahon *J. Exp. Biol.*, **115**, 263–282 (1985)
5. M. H. Raibert, *J. Exp. Biol.*, **29**, 499–514, (1986).
6. U. Saranli, J. Schwind, D. Koditschek, *ICRA*, (1998).
7. J. Schwind, PhD thesis, U. of Michigan, (1998).
8. K. D. Mombaur, et. al, *Robotics, Automation and Mechatronics*, **2**, 2004.
9. D. Remy, K. Buffinton, and R. Siegwart, *IORS*, (2011).
10. R. Tedrake, PhD thesis, MIT, (2004).