

Repetitive jumping control for biped robots via force distribution and energy regulation

Gianluca Garofalo and Christian Ott

German Aerospace Center (DLR), Institute of Robotics and Mechatronics,
Münchner Strasse 20, 82234 Wessling, Germany
{gianluca.garofalo, christian.ott}@dlr.de
<http://www.dlr.de/rmc/rm>

Abstract. The paper presents a new control law to initiate and stop a sequence of repetitive jumps in elastically actuated legged robots. The control approach relies on three control strategies, which are effectively combined to realize the task: hierarchical task space decomposition, balancing force redistribution and energy regulation. A toy example motivates the interconnection of the three parts, while an experimental evaluation is used to corroborate the effectiveness of the controller on a complex biped robot.

Keywords: Humanoid robots, task hierarchy control, balancing control, energy control, jumping robots.

1 Introduction

Although jumping robots are present on the robotic scene since a while, during the years this task has been mainly achieved by quadrupedal robots for which the balancing problem is notably simplified. Both bipedal and quadrupedal robots usually use simplified models or offline optimization in order to design force profiles and reference trajectories [14, 16]. Few of them include elastic elements in the legs [9], but it is not clear how the controller is actively using the springs beside the benefit from improved shock absorption at the impacts. This is one of the points on which this paper will focus.

The notion of elastic joint robots has a long history in robotics. Nevertheless, while in the seminal publications by Spong [18] and De Luca [2], the joint elasticity was originally treated as a disturbance of the rigid-body dynamics, more recent drive concepts (like series elastic actuators [15] or variable impedance actuators [19]) deliberately introduce elasticity for implementing torque control, increasing physical robustness, or reaching high output velocities. Therefore, legged robots have been both the application and motivation for the design of such innovative actuation systems. Elastically actuated legged robots are underactuated systems and, therefore, they are challenging to control. The underactuation is a direct consequence of both their floating base nature and the presence of the springs in the joints.

In this paper, the complete underactuated problem is considered. The robot is neither fixed to the floor, nor rigidly actuated. A bipedal robot is used as case study (see Fig. 1). The derivation of the control law is conceptually split in two steps. In the first one, a hierarchical task space decomposition [12] and an energy regulation [6] are used

to generate a first torque reference. In the second step, a balancing force redistribution [10] modifies the torque reference to ensure the feasibility of the controller. The simplifying hypothesis used throughout the paper requires that the balancing torques can be computed assuming to have a rigid joint, i.e. the current spring torques cannot make the robot lose balance, while the flexibility is considered for the rest of the derivation. The assumption is reasonable since the spring torque is part of the state and cannot be instantaneously changed by control.

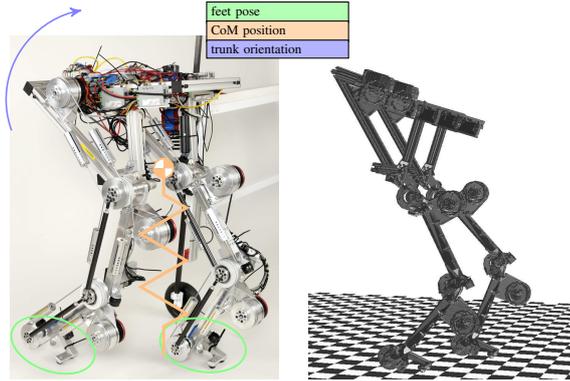


Fig. 1. C-Runner: real test-bed (left) and simulated model in the initial configuration (right) [11].

The main contribution of the paper is a control law that directly uses the elasticity in the joints in order to produce a controlled sequence of jumps, while considering the whole dynamic model of the robot. The resulting jumps are not pre-planned, but a dynamical result of the actions of the springs, whose torques are shaped through the control action, as it is presented in section 6. As the method relies on the interconnection of three control approaches, each one of them is shortly reviewed in section 2 to section 4, while section 5 serves as motivation on how to interconnect these components. The paper is concluded with a final discussion and outline of future work in section 7.

1.1 Notation and model

With a slight abuse of notation, throughout the paper it will be used \mathbf{M} to denote the symmetric and positive definite inertia matrix, \mathbf{C} a Coriolis matrix satisfying $\dot{\mathbf{M}} = \mathbf{C} + \mathbf{C}^T$ and \mathbf{g} the gravity torque vector, with suitable dimensions depending on the type of model considered (i.e. fixed or floating base). When dealing with elastic joints, these matrices have to be understood as link-side quantities, with \mathbf{B} denoting instead the inertia of the motors. The torques $\boldsymbol{\tau}_m$ produced by the motors are an input to the system. These are directly the torques $\boldsymbol{\tau}$ applied to the link for a rigid joint robot (i.e. $\boldsymbol{\tau} = \boldsymbol{\tau}_m$), while for elastic joints the latter are the torques produced by the springs. In addition to the motor torques, the external wrenches stacked in \mathbf{w}_f and mapped through \mathbf{J}_f^T complete the inputs to the system. The most general case considered in this paper is a

floating base robot with linear elastic joints:

$$\mathbf{M}(\mathbf{x})\dot{\mathbf{v}} + \mathbf{C}(\mathbf{x}, \mathbf{v})\mathbf{v} + \mathbf{g}(\mathbf{x}) = \mathbf{Q}^T \boldsymbol{\tau} + \mathbf{J}_f^T(\mathbf{x})\mathbf{w}_f, \quad (1a)$$

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau} = \boldsymbol{\tau}_m, \quad (1b)$$

$$\boldsymbol{\tau} = \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}), \quad (1c)$$

where $\boldsymbol{\theta}, \mathbf{x}, \dot{\boldsymbol{\theta}}, \mathbf{v}$ constitute together the state of the system, being $\boldsymbol{\theta}$ the motor position and \mathbf{x} the link-side configuration, which includes the link position \mathbf{q} and the floating base coordinates. The matrix \mathbf{Q} selects the joint velocities $\dot{\mathbf{q}}$ out of all the velocity coordinates \mathbf{v} , i.e. $\dot{\mathbf{q}} = \mathbf{Q}\mathbf{v}$. Finally, $\mathbf{K} = \text{diag}(K_i)$ is the joint stiffness matrix, with K_i the stiffness constant of the i -th joint. To the assumptions for the derivation of the reduced elastic joint model in [18], the following one is added [6]

Assumption 1

$$\left\| \frac{\partial \mathbf{g}}{\partial \mathbf{q}} \right\| < \min_i K_i \quad \forall \mathbf{x} \in \mathbb{X}$$

where the subset \mathbb{X} of the state space in which all the prismatic joints are kept bounded [1] will always be considered throughout the whole paper.

2 The task hierarchy controller

The task hierarchy controller presented in [12] extends the results in [13] in order to cope with the different priorities that the tasks of a *fixed base manipulator with rigid joints* can have. Given the joint coordinates $\mathbf{q} \in \mathbb{R}^n$, let $\mathbf{y}(\mathbf{q})$ indicate the whole task coordinates, so that $\mathbf{y}(\mathbf{q}) \in \mathbb{R}^n$, and let $\mathbf{y}_i(\mathbf{q})$ be one of the r subtasks. Given the mapping

$$\dot{\mathbf{y}}_i = \mathbf{J}_i(\mathbf{q})\dot{\mathbf{q}} \quad 1 \leq i \leq r, \quad (2)$$

assume additionally that each of the Jacobian matrices $\mathbf{J}_i(\mathbf{q}) \in \mathbb{R}^{m_i \times n}$ is full rank, as well as each Jacobian matrix obtained stacking any $\mathbf{J}_i(\mathbf{q})$. A coordinate transformation can then be considered, which replaces the joint velocities $\dot{\mathbf{q}} \in \mathbb{R}^n$ with $\boldsymbol{\zeta} \in \mathbb{R}^n$ obtained through the extended Jacobian matrix $\bar{\mathbf{J}}_N(\mathbf{q}) \in \mathbb{R}^{n \times n}$, as

$$\boldsymbol{\zeta} = \begin{bmatrix} \zeta_1 \\ \vdots \\ \zeta_r \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{J}}_1(\mathbf{q}) \\ \vdots \\ \bar{\mathbf{J}}_r(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \bar{\mathbf{J}}_N(\mathbf{q})\dot{\mathbf{q}}, \quad (3)$$

where $\bar{\mathbf{J}}_1 = \mathbf{J}_1$ and $\zeta_1 = \dot{\mathbf{y}}_1$, while the others are nullspace velocities $\zeta_i \in \mathbb{R}^{m_i}$. The latter are designed such that they represent the original task as closely as possible, but they are inertially decoupled from the tasks with a higher priority level [12]. The extended Jacobian matrix is invertible, so that

$$\dot{\mathbf{q}} = \bar{\mathbf{J}}_N^{-1}(\mathbf{q})\boldsymbol{\zeta} = \sum_{i=1}^r \mathbf{Z}_i^T(\mathbf{q})\zeta_i, \quad (4)$$

where $\mathbf{Z}_1^T(\mathbf{q}) = \mathbf{J}_1^{+M}(\mathbf{q}) := \mathbf{M}^{-1} \mathbf{J}_1^T (\mathbf{J}_1 \mathbf{M}^{-1} \mathbf{J}_1^T)^{-1}$, while $\mathbf{Z}_i(\mathbf{q}) = \mathbf{J}_i(\mathbf{q}) \mathbf{M}^{-1} \mathbf{Y}_i(\mathbf{q})$ for $2 \leq i \leq r$ and $\mathbf{Y}_i(\mathbf{q})$ the classical nullspace projector [17].

Using this coordinate transformation the system can be written as

$$\Lambda(\mathbf{q})\dot{\zeta} + \Gamma(\mathbf{q}, \dot{\mathbf{q}})\zeta = \bar{\mathbf{J}}_N^{-T}(\mathbf{q})(\boldsymbol{\tau} - \mathbf{g}(\mathbf{q})), \quad (5)$$

where the matrices $\Lambda(\mathbf{q})$ and $\Gamma(\mathbf{q}, \dot{\mathbf{q}})$ are block diagonal and with r blocks (one for each subtask) on the main diagonal. The same stability property as in [13] holds for the closed loop system obtained through the feedback control law in [12], which can easily be rearranged as

$$\boldsymbol{\tau} = \bar{\mathbf{J}}_N^T \left(\bar{\Gamma} \begin{bmatrix} \zeta_1 \\ \vdots \\ \zeta_r \end{bmatrix} + \begin{bmatrix} \mathbf{Z}_1(\mathbf{g} + \mathbf{J}_1^T \boldsymbol{\phi}_1) \\ \vdots \\ \mathbf{Z}_r(\mathbf{g} + \mathbf{J}_r^T \boldsymbol{\phi}_r) \end{bmatrix} \right), \quad (6)$$

where $\bar{\Gamma}$ is obtained from Γ setting the blocks on the main diagonal to zero¹ and $\boldsymbol{\phi}_i(\mathbf{y}_i, \dot{\mathbf{y}}_i)$ is a PD term regulating the task coordinates \mathbf{y}_i to their desired value \mathbf{y}_{d_i} . In addition to [13], here the transient response is guaranteed to have no inertial coupling between the different tasks.

3 The balancing controller

In [10] a balancing controller based on the task hierarchy controller was proposed for a *humanoid robot with rigid joints*. The goal is to have the CoM of the robot and the hip orientation (CoM task) in a given configuration in space. The joint configuration should also be as close as possible to a desired one (posture task). The CoM task together with the posture task completely define the configuration of the floating base robot. Due to the restrictions imposed to the system by the contact state, the posture task might not be fully fulfilled. Assuming to have some of the end effectors of the robot in contact with the environment and some free to move in space, the requirements are to produce balancing wrenches with the first (balancing task) and an impedance behavior with the latter (interaction task). The balancing wrenches have to counteract the effects of all the other task and compensate the gravity, ensuring that the base does not need to be actuated. Indicating by $\boldsymbol{\phi}_i$ the correspondent task force, the tasks are stacked in the task hierarchy as: balancing task ($\boldsymbol{\phi}_1$), interaction task ($\boldsymbol{\phi}_2$), CoM task ($\boldsymbol{\phi}_3$), posture task ($\boldsymbol{\phi}_4$), where all the $\boldsymbol{\phi}_i$ are chosen as PD terms, except for $\boldsymbol{\phi}_1$. The latter is determined through an optimization problem, which guarantees the feasibility of the control law, i.e. it makes sure that each balancing end effector does not lift off, slip or tilt, as well as ensuring that no forces at the floating base are commanded. In fact, the main difference compared to the task hierarchy controller in [12] is that $\mathbf{u} := \mathbf{Q}^T \boldsymbol{\tau}$ cannot be fully chosen, since $\mathbf{u} = [\mathbf{0}^T \boldsymbol{\tau}^T]^T$.

The balancing controller in [10] is designed using a frame attached to the CoM and with orientation given by the hip frame, so that in this coordinates the gravity torque

¹i.e. it is a skew-symmetric matrix.

simplifies to $[m\mathbf{g}_0^T \mathbf{0}^T]^T$, i.e. the weight of the robot with m its total mass. Using a quasi-static argument², the controller (6) becomes

$$\begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} m\mathbf{g}_0 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Xi}_u \\ \boldsymbol{\Xi}_a \end{bmatrix} \boldsymbol{\phi}, \quad (7)$$

where

$$\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{\Xi}_u \\ \boldsymbol{\Xi}_a \end{bmatrix} = [\bar{\mathbf{J}}_{N_1}^T \mathbf{Z}_1 \mathbf{J}_1^T \cdots \bar{\mathbf{J}}_{N_r}^T \mathbf{Z}_r \mathbf{J}_r^T], \quad (8)$$

and all the forces have been stacked in $\boldsymbol{\phi}$. The optimization problem can then be formulated as

$$\min_{\boldsymbol{\phi}_1} (\boldsymbol{\phi}_1 - \boldsymbol{\phi}_{d_1})^T \mathbf{W} (\boldsymbol{\phi}_1 - \boldsymbol{\phi}_{d_1}) \quad (9a)$$

$$\text{s.t. } m\mathbf{g}_0 + \boldsymbol{\Xi}_u \boldsymbol{\phi} = \mathbf{0} \quad (9b)$$

$$f_{min,i} \leq f_{i,\perp} \quad (9c)$$

$$\|\mathbf{f}_{i,\parallel}\| \leq \mu_i f_{i,\perp} \quad (9d)$$

$$\text{CoP}_i(\mathbf{f}_i) \in \mathcal{S}_i \quad (9e)$$

$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\Xi}_a \boldsymbol{\phi} \leq \boldsymbol{\tau}_{max}, \quad (9f)$$

where the cost function minimizes the deviation from a default wrench distribution³ $\boldsymbol{\phi}_{d_1}$ with \mathbf{W} a block-diagonal (one block for each task) positive definite weighting matrix. The equality constraint is derived from the first equation in (7), while from the second it follows

$$\boldsymbol{\tau} = \boldsymbol{\Xi}_a \boldsymbol{\phi}, \quad (10)$$

that is the control law applied to the robot. The last inequality constraint is, therefore, ensuring that the resulting joint torques stay within the limitations of the hardware. Finally, the other constraints take into account the contact model. There, $f_{i,\perp}$ and $\mathbf{f}_{i,\parallel}$ denote the components of the contact force \mathbf{f}_i perpendicular and parallel to the contact surface \mathcal{S}_i . The unilaterality of the contact is taken into account by limiting the normal component to $f_{min,i} \geq 0$. To prevent the end effector from slipping, $\mathbf{f}_{i,\parallel}$ is limited via the friction coefficient μ_i , while tilting is avoided restricting the center of pressure (CoP) of each end effector to \mathcal{S}_i . Each \mathbf{f}_i enters in the dynamic model as part of \mathbf{w}_f .

4 The energy controller

In [6] we introduced a nonlinear dynamic state feedback to initiate and stop a periodic motion for an *elastically actuated manipulator*, therefore extending our previous results

²i.e. neglecting the effects due to the Coriolis matrix.

³Typically, this value is chosen to be an equally distributed gravity compensation between all the end effector in contact with the environment.

for rigid joints [5, 8] and for a single elastic joint [3]. This periodic motion is due to the presence of an asymptotically stable limit cycle generated via a nonlinear feedback of an energy function, while simultaneously forcing the system to evolve on a submanifold of the configuration space. By taking into account the physical total potential energy U (i.e. gravitational plus elastic), the controller in [6] makes use of the energy stored in the springs. In particular, the energy function is given by

$$H(\chi) := \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} + U(\boldsymbol{\eta}, \mathbf{q}) - U(\boldsymbol{\eta}, \bar{\mathbf{q}}_y(\boldsymbol{\eta})), \quad (11)$$

where $\boldsymbol{\eta}, \dot{\boldsymbol{\eta}} \in \mathbb{R}^n$ are the state variables of the dynamic controller. The function $\bar{\mathbf{q}}_y(\boldsymbol{\eta})$ is given by

$$\begin{aligned} \bar{\mathbf{q}}_y(\boldsymbol{\eta}) &:= \arg \min_{\mathbf{q}} U(\boldsymbol{\eta}, \mathbf{q}) \\ \text{s.t. } \mathbf{y}_1(\mathbf{q}) &= \mathbf{0}, \end{aligned} \quad (12)$$

which is well defined thanks to Assumption 1 and satisfies $\dot{U}(\boldsymbol{\eta}, \bar{\mathbf{q}}_y(\boldsymbol{\eta})) = \dot{\boldsymbol{\eta}}^T \mathbf{K}(\boldsymbol{\eta} - \bar{\mathbf{q}}_y(\boldsymbol{\eta}))$, as shown in [6]. The function $\mathbf{y}_1 : \mathbb{R}^n \rightarrow \mathbb{R}^{n-1}$ has full rank Jacobian matrix, so that $\mathbf{y}_1(\mathbf{q}) = \mathbf{0}$ defines a submanifold. Forcing the robot to this submanifold corresponds to reach the configuration in which the robot will perform the periodic oscillation. The dynamic equations of the controller are

$$\mathbf{B} \ddot{\boldsymbol{\eta}} + K_H \tilde{H} \mathbf{K}(\bar{\mathbf{q}}_y(\boldsymbol{\eta}) - \mathbf{q}) + \mathbf{D}_\eta \dot{\boldsymbol{\eta}} + \mathbf{K}_\eta \tilde{\boldsymbol{\eta}} = \mathbf{0} \quad (13a)$$

$$\boldsymbol{\tau}_m = \boldsymbol{\tau}_d + \mathbf{K}(\boldsymbol{\eta} - \mathbf{q}) - K_H \tilde{H} \mathbf{K}(\bar{\mathbf{q}}_y(\boldsymbol{\eta}) - \mathbf{q}) + \mathbf{B} \mathbf{K}^{-1}(\ddot{\boldsymbol{\tau}}_d - \mathbf{D}_\tau \dot{\boldsymbol{\tau}} - \mathbf{K}_\tau \tilde{\boldsymbol{\tau}}) - \mathbf{D}_\eta \dot{\boldsymbol{\eta}} - \mathbf{K}_\eta \tilde{\boldsymbol{\eta}}, \quad (13b)$$

where the positive scalar K_H and the symmetric, positive definite matrices $\mathbf{K}_\tau, \mathbf{D}_\tau, \mathbf{K}_\eta, \mathbf{D}_\eta \in \mathbb{R}^{n \times n}$ are control gains and $\boldsymbol{\tau}_m \in \mathbb{R}^n$ is the output (feedback to the robot as in Fig. 2). Additionally, $\tilde{H} = H - H_d$, $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta} - \boldsymbol{\eta}_d$, with $H_d \in \mathbb{R}$ and $\boldsymbol{\eta}_d \in \mathbb{R}^n$ constant desired values, while $\tilde{\boldsymbol{\tau}} = \mathbf{K}(\boldsymbol{\theta} - \boldsymbol{\eta}) - \boldsymbol{\tau}_d$ being $\boldsymbol{\tau}_d \in \mathbb{R}^n$ the input function. The latter is computed based on the state of the system and of the controller itself [6] and, loosely speaking, it is responsible to enforce the $n - 1$ virtual constraints given by $\mathbf{y}_1(\mathbf{q}) = \mathbf{0}$. Using the notation from section 2, it is straightforward to rewrite $\boldsymbol{\tau}_d$ as

$$\boldsymbol{\tau}_d = \bar{\mathbf{J}}_N^T \left(\bar{\boldsymbol{\Gamma}} \boldsymbol{\zeta} + \begin{bmatrix} \mathbf{Z}_1 (\mathbf{g} - \mathbf{K}(\boldsymbol{\eta} - \mathbf{q}) - \mathbf{J}_1^T (\mathbf{K}_1 \mathbf{y}_1 + \mathbf{D}_1 \dot{\mathbf{y}}_1)) \\ 0 \end{bmatrix} \right), \quad (14)$$

where \mathbf{K}_1 and \mathbf{D}_1 are the coefficients of the proportional and derivative action of the PD term. In [6] it is shown that the resulting closed-loop system has an asymptotically stable solution consisting of an equilibrium point, if $H_d = 0$, or a limit cycle, if $H_d > 0$.

5 A toy model for jumping

Our previous works [3, 6, 8] guarantee the presence of an asymptotically stable limit cycle for the nominal system, i.e. without uncertainties. Although not formally proven

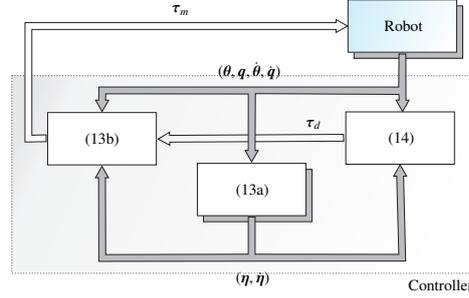


Fig. 2. Closed loop system.

in the stability analysis, the experiments therein show that periodicity is achieved even in case of oscillations around the desired value of the energy, due to unmodeled disturbances. On the other hand, in [7] we showed that a form of energy regulation combined with a control approach based on the Spring Loaded Inverted Pendulum (SLIP), was able to produce a periodic walking pattern for a simulated biped robot with impacts. These works motivate the idea behind the use of energy regulation for a jumping robot. This idea can be exemplified using a 1 - dimensional SLIP model, i.e. the simplest scenario in which elasticity and energy regulation can be combined to obtain a jumping behavior. The system is represented by a single massless spring connected to a mass constrained to move along the vertical direction. The model of the system is:

$$\begin{cases} \ddot{y} + g = -K_h \tilde{H} \dot{y} - k(y - l_0) & y \leq l_0 \\ \ddot{y} + g = 0 & y > l_0 \end{cases}, \quad (15)$$

where y is the position of the mass, H is the total energy of the system, l_0 the rest length of the spring, k the stiffness constant and g the gravitational acceleration. In addition, it is assumed that the velocities before and after the impact are related by a certain coefficient of restitution α , i.e. $\dot{y}^+ = \alpha \dot{y}^-$.

Depending on the values of α and K_h , the different phase portraits are reported in Fig. 3. Firstly, in both cases, the strong asymmetry of the plot is due to the switching of the dynamics. The discontinuity, instead, is due to the impact. Finally, while with a high K_h (left) the system is able to reach the desired value of energy before the next take off, this is not the case with a small value of K_h (right). Nevertheless, a limit cycle is reached for which the energy lost at the impact balances the one injected before the take off. The energy for the two cases is plotted in Fig. 4.

Controlling the energy is therefore not only beneficial to exploit the presence of the elastic joints (as shown in [3, 6]), but it also provides a way to cope with an uncertain impact model. In this simple example, when the energy is quickly regulated to the desired value, the latter is also directly responsible for the jumping height. In Fig. 5, the actual and desired height are plotted, with the latter obtained from the desired energy value. For a given energy loss, the maximum reachable height is limited by the speed at which the energy can be reinjected in the system.

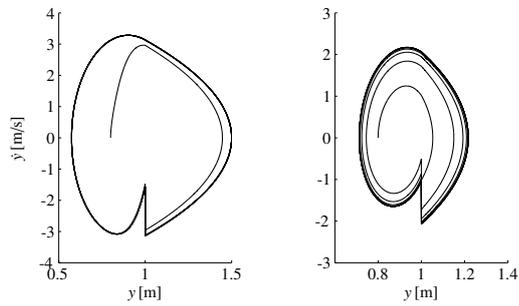


Fig. 3. Phase portrait of the toy model for $K_h = 5 \text{ s/kgm}^2$ (left) and $K_h = 0.5 \text{ s/kgm}^2$ (right).

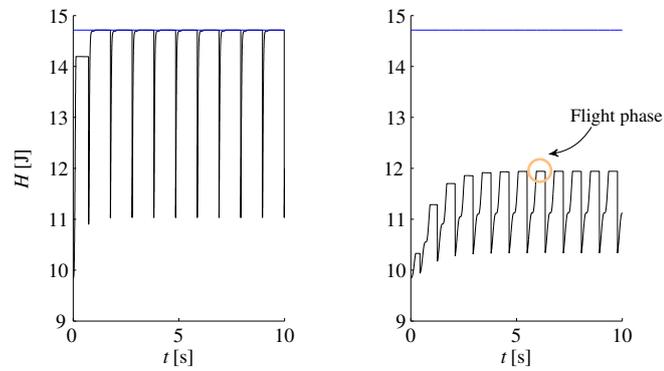


Fig. 4. Total energy and desired value for $K_h = 5 \text{ s/kgm}^2$ (left) and $K_h = 0.5 \text{ s/kgm}^2$ (right). In the right plot, the energy is constant only during the flight phase.

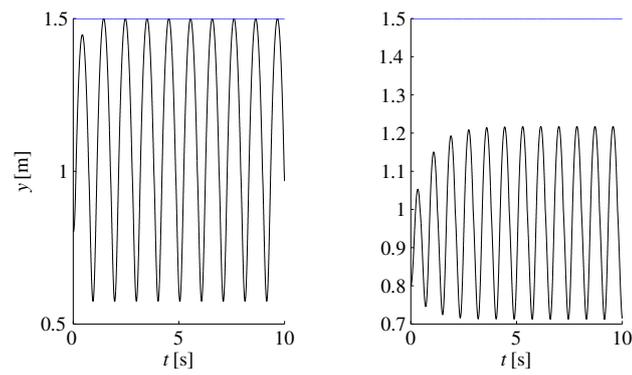


Fig. 5. Jumping height for $K_h = 5 \text{ s/kgm}^2$ (left) and $K_h = 0.5 \text{ s/kgm}^2$ (right).

6 The jumping controller

The main idea behind the jumping controller is to make the vertical motion of the robot's CoM behave similarly to the toy model of the previous section. The robot will be constrained in such a way that a vertical oscillation of the CoM is the only allowed motion. Given these constraints and the presence of the elasticity in the system, the model is conceptually similar to a 1 - dimensional SLIP model, as it is depicted in Fig. 1 for the C-Runner. The regulation of the energy will then produce the desired oscillatory behavior of a jumping motion, as seen in section 5. The input function of the energy controller will be chosen in the same fashion as in (6), in order to enforce the virtual constraints. As in section 3, in the first stage the underactuation and the contact model will not be taken into account. An optimization problem will later on consider these restrictions by redistributing ϕ to generate feasible contact forces.

Starting from the dynamic equations (1) of the model, the dynamic state feedback controller from section 4 is used. This leads to the closed loop system

$$\mathbf{M}(\mathbf{x})\dot{\mathbf{v}} + \mathbf{C}(\mathbf{x}, \mathbf{v})\mathbf{v} + \mathbf{g}(\mathbf{x}) = \mathbf{Q}^T \boldsymbol{\tau} + \mathbf{J}_f^T(\mathbf{x})\mathbf{w}_f \quad (16a)$$

$$\ddot{\boldsymbol{\tau}} + \mathbf{D}_\tau \dot{\boldsymbol{\tau}} + (\mathbf{K}_\tau + \mathbf{K}\mathbf{B}^{-1})\tilde{\boldsymbol{\tau}} = \mathbf{0} \quad (16b)$$

$$\mathbf{B}\ddot{\boldsymbol{\eta}} + K_H \tilde{\mathbf{H}}\mathbf{K}(\bar{\mathbf{q}}_y(\boldsymbol{\eta}) - \mathbf{q}) + \mathbf{D}_\eta \dot{\boldsymbol{\eta}} + \mathbf{K}_\eta \tilde{\boldsymbol{\eta}} = \mathbf{0}, \quad (16c)$$

where, the input function $\boldsymbol{\tau}_d$ will be designed through an optimization based approach similar to the one used in section 3. Notice also that

$$\mathbf{Q}^T \boldsymbol{\tau} = \mathbf{Q}^T \mathbf{K}(\boldsymbol{\eta} - \mathbf{q}) + \mathbf{Q}^T \boldsymbol{\tau}_d + \mathbf{Q}^T \tilde{\boldsymbol{\tau}}. \quad (17)$$

The details of the optimization problem will be presented later on. For now assume that $\boldsymbol{\tau}_d$ has been computed and consider the system conditionally to the convergence of the torque error, i.e. $\tilde{\boldsymbol{\tau}} = \mathbf{0}$. Therefore, the system to consider for the design of $\boldsymbol{\tau}_d$ is

$$\mathbf{M}(\mathbf{x})\dot{\mathbf{v}} + \mathbf{C}(\mathbf{x}, \mathbf{v})\mathbf{v} + \mathbf{g}(\mathbf{x}) - \mathbf{Q}^T \mathbf{K}(\boldsymbol{\eta} - \mathbf{q}) = \mathbf{Q}^T \boldsymbol{\tau}_d + \mathbf{J}_f^T(\mathbf{x})\mathbf{w}_f \quad (18a)$$

$$\mathbf{B}\ddot{\boldsymbol{\eta}} + K_H \tilde{\mathbf{H}}\mathbf{K}(\bar{\mathbf{q}}_y(\boldsymbol{\eta}) - \mathbf{q}) + \mathbf{D}_\eta \dot{\boldsymbol{\eta}} + \mathbf{K}_\eta \tilde{\boldsymbol{\eta}} = \mathbf{0}. \quad (18b)$$

In a first step, it is assumed that $\mathbf{Q}^T \boldsymbol{\tau}_d$ can be freely chosen and therefore, mutatis mutandis, from (6) and (14) it follows

$$\mathbf{Q}^T \boldsymbol{\tau}_d = \bar{\mathbf{J}}_N^T \bar{\boldsymbol{\Gamma}} \boldsymbol{\zeta} + \begin{pmatrix} \left[\begin{array}{c} \mathbf{Z}_1(\mathbf{g} - \mathbf{Q}^T \mathbf{K}(\boldsymbol{\eta} - \mathbf{q}) + \mathbf{J}_1^T \boldsymbol{\phi}_{d_1}) \\ \vdots \\ \mathbf{Z}_{r-1}(\mathbf{g} - \mathbf{Q}^T \mathbf{K}(\boldsymbol{\eta} - \mathbf{q}) + \mathbf{J}_{r-1}^T \boldsymbol{\phi}_{d_{r-1}}) \\ 0 \end{array} \right] \end{pmatrix}. \quad (19)$$

Compared to (6), one notices that there is no compensation of gravitational and elastic torques for the last task, i.e. the task responsible for the generation of the limit cycle. This is not surprising since the total potential energy is directly considered for the generation of the limit cycle. In (19), each $\boldsymbol{\phi}_{d_i}$ is a PD term that guarantees the regulation

of the task coordinates \mathbf{y}_i to their desired value \mathbf{y}_{d_i} . Both \mathbf{y}_i and \mathbf{y}_{d_i} are still to be defined in order to make the robot jump.

The exact definition of the constraints is exemplified, in the next subsection, for the robot in Fig. 1. Nevertheless, the choice is always based on similar tasks as in section 3, i.e. a task for the end effectors in contact with the environment, one for those free to move, one for the CoM and one for the posture. In particular, the choice of having the end effectors in contact with the environment as the first task is key to the design process. Because of this choice, $\mathbf{J}_1 = \mathbf{J}_f$ and it follows that the reaction wrench will only appear in the first task when performing the change of coordinates used in section 2. This is a consequence of the property

$$\mathbf{J}_1(\mathbf{q})\bar{\mathbf{J}}_N^{-1}(\mathbf{q}) = [\mathbf{E} \ \mathbf{O} \ \dots \ \mathbf{O}] ,$$

where \mathbf{E} is the identity matrix and \mathbf{O} a matrix of zeros. Therefore, as in [10], during the remainder of the design phase, the external wrenches are not taken into account⁴.

The last step of the design process consists in redistributing ϕ_d via an optimization problem as in section 3. In this way, the restrictions on the admissible $\mathbf{Q}^T \tau_d$ as well as the contact constraints can be taken into account. In particular, (19) can be rewritten as

$$\begin{bmatrix} \mathbf{0} \\ \tau_d \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{E}}_u \\ \bar{\mathbf{E}}_a \end{bmatrix} \phi_d + \bar{\mathbf{J}}_N^T \underbrace{\left(\bar{\Gamma} \zeta + \begin{bmatrix} \mathbf{Z}_1 \mathbf{g} - \mathbf{Q}^T \mathbf{K}(\eta - \mathbf{q}) \\ \vdots \\ \mathbf{Z}_{r-1} \mathbf{g} - \mathbf{Q}^T \mathbf{K}(\eta - \mathbf{q}) \\ 0 \end{bmatrix} \right)}_{[\mathbf{b}_u^T \ \mathbf{b}_a^T]^T} . \quad (20)$$

Finally, τ_d can be obtained solving the following optimization problem

$$\min_{\phi} (\phi - \phi_d)^T \mathbf{W} (\phi - \phi_d) \quad (21a)$$

$$\text{s.t. } \bar{\mathbf{E}}_u \phi + \mathbf{b}_u = \mathbf{0} \quad (21b)$$

$$f_{\min,i} \leq f_{i,\perp} \quad (21c)$$

$$\|f_{i,\parallel}\| \leq \mu_i f_{i,\perp} \quad (21d)$$

$$\text{CoP}_i(\mathbf{f}_i) \in \mathcal{S}_i \quad (21e)$$

$$\tau_{\min} - \mathbf{b}_a \leq \bar{\mathbf{E}}_a \phi \leq \tau_{\max} - \mathbf{b}_a \quad (21f)$$

and setting $\tau_d = \bar{\mathbf{E}}_a \phi^* + \mathbf{b}_a$, where ϕ^* is the optimal solution⁵.

Concerning the flight phase, it is important to remember that the CoM dynamics cannot be influenced by the motors. Similarly, not all the velocity coordinates can be freely influenced since they have to satisfy the conservation of the angular momentum

⁴This is justified by the fact that the optimization realizes a wrench at the feet, which does not violate the constraints of keeping them on the floor. If this is the case, the wrench \mathbf{w}_f will then be the reaction to the wrench exerted by the feet.

⁵The time differentiation of the signal τ_d can be obtained by filtering techniques.

[4]. The strategy consists, therefore, in switching the weights of the different tasks used in the cost function. A graphical representation of how the weights are updated for the C-Runner can be found in Fig. 6. Notice that while the number associated to the task defines how they are organized in the task hierarchy, i.e. defines the inertial decoupling of the tasks, the weights define which forces the optimization is more likely to modify from the desired values in order to satisfy the constraints of the problem.

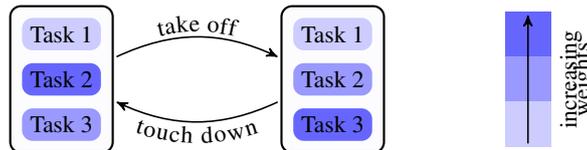


Fig. 6. The weights in the cost function are switched depending on the contact phase. The tasks are defined in section 6.1.

6.1 The C-Runner

The C-Runner, Fig. 1, is a two-legged planar testbed with linear series elastic actuation. The robot is equipped with torque and position sensors, as well as force sensors into each silicone half domes representing the contact points at the feet, to provide a full state measurement. The trunk is connected to a boom, which constrains the robot to move in the sagittal plane. Therefore, the link-side configuration of the robot is defined by nine parameters, three for the trunk and one for each joint. More details on the robot can be found in [11]. In order to achieve the desired jumping pattern, the first task is chosen to define the pose of the feet (6 coordinates), the second the horizontal position of the CoM and the trunk orientation (2 coordinates), the third the vertical position of the CoM (1 coordinate). A unique link-side configuration exists, which fulfills all the tasks.

Task one (feet pose) This task is responsible for generating the required interaction wrenches. The feet are asked to stay parallel to the floor and in contact with the ground. The weights of this task are the lowest in order to allow the optimization to use the feet for providing the necessary reaction wrenches by pushing into the floor. In section 7, it is discussed the necessity to have a precise foot placement for gaits like running and walking. In these cases, the weights of this task have to be the highest during the flight phase.

Task two (CoM horizontal position and trunk orientation) The position of the CoM is responsible for the balancing capabilities of the robot. The horizontal position of the CoM is chosen to be in the middle of the support polygon. The weights of this tasks are the highest during the contact phase, as it is fundamental to prevent the robot from falling. During the flight phase, instead, the weights are lowered since only the relative position between the CoM and the feet can be influenced. Finally, the trunk orientation completes the overall posture of the robot.

Task three (CoM vertical position) This task is responsible for the oscillatory behavior of the sequence of jumps. Assuming that all the other tasks are perfectly satisfied, it follows that the floor will exert a net force on the robot which is parallel to the weight. This is justified by the fact that a movement of the CoM along the horizontal position would be otherwise produced. Therefore, the vertical oscillation will continue until the energy is high enough to lead to lift off, similarly to section 5.

6.2 Evaluation

The proposed control law is evaluated in an experiment, for which the values of the proportional action for each PD term are listed in Table 1, while the correspondent coefficient of the derivative action was always chosen using a damping ratio of 0.2.

Table 1. Coefficients for the proportional action of the PD terms.

	Task one	Task two
Linear	600 N/m	2000 N/m
Angular	80 Nm/rad	800 Nm/rad

Starting from the initial configuration shown in Fig. 1, corresponding to the equilibrium, the desired energy is set to 10 J and then to zero, as shown in Fig. 7. The plot also shows the real value of the energy. Concerning the evolution of the energy, oscillations were expected if for no other reason than the presence of the impacts, as seen in section 5. It should be mentioned that there is no special requirement of reaching the exact desired energy, as long as the desired jumping behavior is obtained. If a higher jumping height is requested, one might increase the desired value of energy and/or the gain K_f , within the limits set by the hardware.

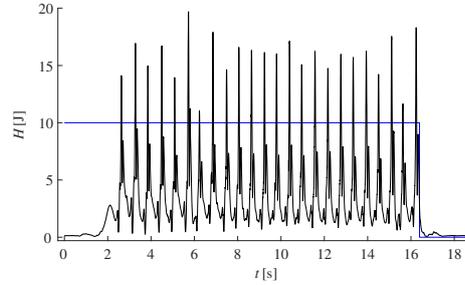


Fig. 7. Desired energy (blue) and actual value (black).

The resulting jumping height of the CoM is reported together with the height of one of the feet in Fig. 8. As it can be noticed, the robot starts from the equilibrium configuration, then the jumps are performed according to the desired energy and finally stops again when the energy goes back to zero.

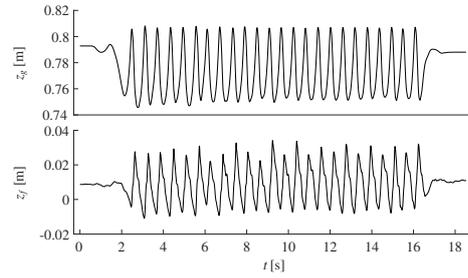


Fig. 8. Vertical position z_g of the CoM and height z_f of one of the feet.

Finally, in Fig. 9 one can see three entries of the inputs and outputs of the optimization problem. All the desired values of the PD terms are well tracked when the robot is at a rest or making small oscillations, except for the vertical force of the foot. The mismatch is actually the sign that the optimization is working properly, although one might think exactly the opposite. When the robot is at rest on the floor, the desired PD term is zero since the foot is exactly stopped at the required height. Nevertheless, the optimization “knows” that the gravity needs to be compensated and, as the foot task has a low weight in the cost function, commands the robot to push into the floor. On the other hand, when the robot starts jumping, deviations can be observed also for the horizontal position of the CoM. This is again a consequence of the constraints and how the weights are chosen in Fig. 6. The touch down events can be easily recognized by the jumps in the plots.

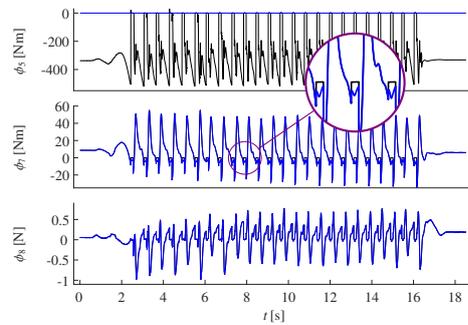


Fig. 9. Values of ϕ for the tasks: left foot height, trunk orientation and CoM horizontal position (from top to bottom). The desired values are in blue and the real in black.

7 Conclusion

Our previous controller for limit cycle generation was incorporated in an optimization framework (originally designed for balancing), in order to realize a jumping pattern. A toy model was used to motivate the use of energy regulation to achieve the desired jumps, showing that energy regulation is not beneficial just to exploit the presence of the elastic joints. The controller was evaluated in an experiment using a planar elastically actuated biped robot, which shows the effectiveness of the controller and a behavior similar to the toy example. This controller can be used as core for developing additional locomotion patterns. Crucial in these cases becomes the role of planning and the presence of a state machine. The position of the feet not in contact with the floor is, in fact, paramount for a successful stride and the state machine has to orchestrate the sequence of different phases, i.e. double support, single support and flight phase. These topics are part of future works.

Acknowledgment

The authors would like to thank Alexander Werner and Florian Loeffl for their constant support with the hardware.

References

1. Canudas-de-Wit, C., Siciliano, B., Bastin, G.: Theory of robot control. Springer-Verlag, London (1996)
2. De Luca, A.: Dynamic control of robots with joint elasticity. In: IEEE Int. Conf. on Robotics and Automation (ICRA). pp. 152–158. Philadelphia, USA (Apr 1988)
3. Garofalo, G., Engelsberger, J., Ott, C.: On the regulation of the energy of elastic joint robots: excitation and damping of oscillations. In: American Control Conference (ACC). pp. 4825–4831. Chicago, USA (Jul 2015)
4. Garofalo, G., Henze, B., Engelsberger, J., Ott, C.: On the inertially decoupled structure of the floating base robot dynamics. In: 8th Vienna International Conference on Mathematical Modelling (2015). pp. 322–327. Vienna, Austria (Feb 2015)
5. Garofalo, G., Ott, C.: Limit cycle control using energy function regulation with friction compensation. IEEE Robotics and Automation Letters (RA-L) 1(1), 90–97 (2016)
6. Garofalo, G., Ott, C.: Energy based limit cycle control of elastically actuated robots. IEEE Trans. on Automatic Control 62(5), 2490–2497 (2017)
7. Garofalo, G., Ott, C., Albu-Schäffer, A.: Walking control of fully actuated robots based on the bipedal SLIP model. In: IEEE Int. Conf. on Robotics and Automation (ICRA). pp. 1999–2004. Saint Paul, USA (May 2012)
8. Garofalo, G., Ott, C., Albu-Schäffer, A.: Orbital stabilization of mechanical systems through semidefinite Lyapunov functions. In: American Control Conference (ACC). pp. 5735–5741. Washington DC, USA (Jun 2013)
9. Gehring, C., Coros, S., Hutter, M., Bloesch, M., Hoepfner, M.A., Siegwart, R.: Control of dynamic gaits for a quadrupedal robot. In: IEEE Int. Conf. on Robotics and Automation (ICRA). pp. 3287–3292. Karlsruhe, Germany (May 2013)
10. Henze, B., Dietrich, A., Ott, C.: An approach to combine balancing with hierarchical whole-body control for legged humanoid robots. IEEE Robotics and Automation Letters (RA-L) 1(2), 700–707 (2016)

11. Loeffl, F.C., Werner, A., Lakatos, D., Reinecke, J., Wolf, S., Burger, R., Gumpert, T., Schmidt, F., Ott, C., Grebenstein, M., Albu-Schäffer, A.: The DLR C-Runner: Concept, design and experiments. In: IEEE/RAS Int. Conf. on Humanoid Robots. pp. 758–765. Cancun, Mexico (Nov 2016)
12. Ott, C., Dietrich, A., Albu-Schäffer, A.: Prioritized multi-task compliance control of redundant manipulators. *Automatica* 53, 416–423 (2015)
13. Ott, C., Kugi, A., Nakamura, Y.: Resolving the problem of non-integrability of nullspace velocities for compliance control of redundant manipulators by using semi-definite Lyapunov functions. In: IEEE Int. Conf. on Robotics and Automation (ICRA). pp. 1456–1463. Pasadena, USA (May 2008)
14. Park, H.W., Park, S., Kim, S.: Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D running of MIT Cheetah 2. In: IEEE Int. Conf. on Robotics and Automation (ICRA). pp. 5163–5170. Seattle, USA (May 2015)
15. Pratt, G.A., Williamson, M.M.: Series elastic actuators. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). pp. 399–406. Pittsburgh, USA (Aug 1995)
16. Sakka, S., Yokoi, K.: Humanoid vertical jumping based on force feedback and inertial forces optimization. In: IEEE Int. Conf. on Robotics and Automation (ICRA). pp. 3752–3757. Barcelona, Spain (Apr 2005)
17. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated (2008)
18. Spong, M.W.: Modeling and control of elastic joint robots. *the ASME Journal of Dynamic Systems, Measurement, and Control* 109, 310–318 (1987)
19. Wolf, S., Eiberger, O., Hirzinger, G.: The DLR FSJ: Energy based design of a variable stiffness joint. In: IEEE Int. Conf. on Robotics and Automation (ICRA). pp. 5082–5089. Shanghai, China (May 2011)