

A two-stage suboptimal approximation for variable compliance and torque control

Perle Geoffroy, Olivier Bordron, Nicolas Mansard, M Raison, Olivier Stasse, T Bretl

▶ **To cite this version:**

Perle Geoffroy, Olivier Bordron, Nicolas Mansard, M Raison, Olivier Stasse, et al.. A two-stage suboptimal approximation for variable compliance and torque control. European Control Conference, Jun 2014, Strasbourg, France. pp.1151 - 1157, 10.1109/ECC.2014.6862557 . hal-01114610

HAL Id: hal-01114610
https://hal.science/hal-01114610

Submitted on 9 Feb 2015

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A two-stage suboptimal approximation for variable compliance and torque control

P. Geoffroy[*,†], O. Bordron[*,△], N. Mansard[*], M. Raison[†], O. Stasse[*], T. Bretl[*,□]

***Abstract*— Variable-stiffness actuator is a very appealing mechatronic design that combines the efficiency of stiff actuator in free space with the consistency of elastic actuation in contact. The control of such an actuation system remains a challenge due to its non-linearity and by the fact that it doubles the number of control inputs. In this paper, we propose an original control strategy to compute the whole-body movement of a complex variable-stiffness robot during dynamic task execution. Operational space control is first used to compute both the joint torque and stiffness from operational references. A non-linear model-predictive controller is then proposed to track at higher frequency these references on each joint separately. The effectiveness of this approach is then validated on two models of real actuator with adjustable stiffness, and finally on an explosive motion to make a humanoid robot jump.**

I. Introduction

This paper deals with the control of multiple body robot with compliant actuation. This paradigm of actuation originates from the series-elastic actuator [1]. An example of complete series-elastic actuation is the Coman robot [2]. The compliance is a very positive property when the robot works in contact with its environment, but deeply complicates the control in particular when performing fast and accurate movements in free space. Variable-stiffness actuators are a nice trade-off, offering compliancy when in contact or before an impact but being stiff when accuracy is required [3].

We are interested in the capability to generate and control complex movements, e.g. a humanoid robot grasping an object while keeping its balance [4] or several manipulator robots moving an object collaboratively. The brute-force solution is to model the movement to execute as an optimal motion problem, and use direct solution method to obtain an approximation of the optimum [5]. However, this solution is computationally expensive: it scales in the cube of the number of actuators. It is yet prohibitive for complex variable-stiffness robot that have two motors for each axis [6].

A classical solution to reduce the complexity is to consider only the instantaneous linearization of the system evolution [7], resulting in inverse kinematics when only the geometry is considered [8] or inverse dynamics when inertia and forces are considered. In particular, operational-space inverse dynamics, detailed below, proposes to compute the output motor joint torques in function of references forces and acceleration in some specific operational spaces [9]. The interest is that, when the proper operational space is chosen, the reference motion is generally very easy to specify.

Two major problems limit the generalization of operation-space inverse dynamics as the classical motion generation and method. First, joint torque is not a direct input of the motors, and a complex non-linear output as soon as reductors are introduced between the motor and the joint. Reductors are indeed difficult to properly model, which prevents feedforward control, while feedback control requires both an expensive torque sensor and a very fast (and very expensive) electronics [10]. Second, there is no sense to instantaneously linearize a spring, since any force at the input of a spring would instantaneously produce zero motion at its output.

However, the spring actuation can be considered as an advantage to apply a reference torque on a real robot, by using the spring as a cheap force sensor. A first level compute the reference joint torques from the operational specification. The torques are then tracked by an optimal-motion solver for each joint, acting at higher frequency and taking care of the spring dynamics. Compared to the whole-body optimal solver described for example in [6], this solution decouples the space complexity (solved by the whole-body inverse dynamics) from the temporal complexity (solved by each actuator optimal controller). The decoupled solution can be seen as a suboptimal approximation to the whole-body optimal problem.

With respect to this strategy, the paper proposes three contributions. We propose a method to compute the reference joint compliance from operation compliance specification, based on the operational-space formulation (Section II). We then propose a generic method to efficiently compute the optimal control law at the joint level (Section III). This method can be directly applied to any possibly non-linear joint model and various cost function, for tracking the output acceleration, torque or stiffness as shown in Section IV. Finally, we propose an implementation of the optimal controller, coupled with the operational-space solver, to produce complex movements. In particular, we demonstrate a whole-body jump in simulation with a modified stiffness-variable humanoid in Section V.

II. From whole-body objectives to actuator references

We first recall in Section II-A the generic model of a free-floating robot in contact with a known, typically rigid, environment. From this model, it is classical to deduce a torque control law based on task specifications, which is

[*] LAAS-CNRS, Univ. Toulouse, France (name.surname@laas.fr).
[†] École Polytechnique de Montréal, Montréal, Canada.
[△] École Normale supérieure de Bretagne, Ker Lann, France.
[□] University of Illinois, Urbana-Champaign, USA

also recalled in Section II-B. Based on this model and this procedure, we propose a solution to deduce the needed whole-body compliance to meet some reference compliance expressed in given task spaces in Section II-C.

A. Dynamic model

In general, the dynamics model of the robot whole body, considering all the bodies and the actuator to be stiff, is:

$$A(q)\ddot{q} + b(q,\dot{q}) = S^{\mathsf{T}}\tau - \sum_{k=1}^{n_c} J_k(q)^{\mathsf{T}} f_k \tag{1}$$

where A is the generalized inertia matrix, b is the dynamic drift (sum of Coriolis, centrifugal and gravity forces), τ are the joint torques (typically, the actuator output torques), S is the matrix that selects the actuated degrees of freedom (DOF), f_k are the n_c contact forces exerted by the robot on the environment at the contact points p_k and $J_k = \frac{\partial p_k}{\partial q}$ are the Jacobian of the contact points.

This equation represents the inverse dynamics: from a given joint acceleration given as a reference, it gives the forces that would lead to this acceleration. The three $\ddot{q}$, τ and $f_c = (f_1, ..., f_{n_c})$ are variable and evolve together. Very often in robotics, we want to impose a reference $\ddot{q}$ and search for a solution τ meeting this reference, while f_c is explicitly needed.

It is possible to get ride of f_c and then compute the needed joint torques if an interface model is given. For example, if assuming a rigid contact $J_k\ddot{q} + \dot{J}_k\dot{q} = 0$ [1], an acceleration-free relation can be obtained from (1):

$$J_c A^{-1}\tau = J_c A^{-1} J_c^T f_c + b_c$$

where $J_c = (J_1, ..., J_{n_c})$ and $b_c = J_c A^{-1} b - \dot{J}_c\dot{q}$. When J_c is full row rank, f_c can be immediately deduced from τ. This leads to equivalent dynamics in contact [11]:

$$A\ddot{q} + b_c = (SN_c)^T\tau \tag{2}$$

where $N_c = I - J_c^T(J_c A^{-1} J_c^T)^{-1} J_c A^{-1}$ is the projector in the null space of J alongside the direction A^{-1} and $b_c = N_c^T b - (J_c A^{-1} J_c^T)^{-1}\dot{J}_c\dot{q}$. This equation still holds when J_c is not full row rank, even if slightly less intuitively obtained [12].

In this last equation, the contact forces are considered as additional actuators to compensate the underactuation expressed by S^T. It is possible to separate the forces into support forces f_s, used as actuators, and the other forces, which would be typically used to accomplished a given force reference. In that case, the equation is written:

$$A\ddot{q} + b_S = (SN_S)^T\tau + \sum_{k=1}^{n_f} J_k f_k$$

[1]The contact typically also imposes the forces to stay in the friction, which comes as an additional constraint expressed in f_c.

B. Operational-space inverse dynamics

1) For one task: The task-function [13], or operational-space [9], approach was proposed to enable the user to define the objectives of the motion to be executed in a dedicated task, or operational, space rather than directly in the configuration space. The task is defined by a vector function $e(q,\Omega) \in \mathbb{R}^m$ of the robot configuration q and the rest of the universe Ω, whose image space is called the task space and whose Jacobian with respect to the configuration is denoted $J = \frac{\partial e}{\partial q}$. In addition to the task function, the control in the task space is given as a reference vector field $\ddot{e}^*$ in the tangent to the task space. The task acceleration $\ddot{e}$ can be linked to the control input τ by multiplying (2) by JA^{-1}:

$$\ddot{e} = JA^{-1}(SN_S)^T\tau + \mu$$

where μ is the task drift that collects the non linear terms. τ is obtained by solving this equation in the least-square sense:

$$\tau = (JA^{-1}(SN_S)^T)^{\#}\left(\ddot{e}^* - \mu\right)$$

where $.^{\#}$ denotes any reflexive generalized inverse, typically the inverse weighted on the left by $SA^{-1}(SN_S)^T$ [14].

2) For several tasks: This last form satisfies the task while minimizing the given norm, but the general solution is written:

$$\tau = (JA^{-1}(SN_S)^T)^{\#}\left(\ddot{e}^* - \mu\right) + P\tau_2$$

The second input τ_2 can be chosen arbitrarily ($\tau_2 = 0$ results in the least norm solution). This second input can be used to satisfy a secondary task, and recursively, any number of tasks set up in a hierarchy order. Denoting by e_1 ... e_p the hierarchy of tasks with J_i the task Jacobian, the general solution is:

$$\tau = \sum_i G_{i|i-1}^{\#}\left(\ddot{e_i}^* - \mu_{i|i-1}\right)$$

with $G_i = J_i A^{-1}(SN_S)^T$, $G_{i|i-1} = G_i P_{i-1}$, $P_0 = I$, $\forall i > 0, P_i = P_{i-1} - G_{i|i-1}^{\#} G_{i|i-1}$ and $\mu_{i|i-1}$ collects all the non linear terms. More details are given in [15]. In particular, the hierarchy can be implemented on an efficient manner and be coupled with an active-set search if some task references are given as bounds rather that equality.

C. Operational-space inverse compliance

Our objective in this section is to define a similar scheme to compute the configuration compliance, being given some reference compliance in dedicated task spaces. We first consider the fully actuated case, i.e. when SN_S is invertible (typically, a fixed manipulator whose only considered contact forces are at the interface with the ground).

1) Direct compliance model: We denote by Γ the compliance of the axes, supposed constant (linear springs). A variation of the torques then induces a variation of the axis position Δq given by

$$\Delta q = {}^{\Delta}\Gamma\tau$$

Γ can be seen both as a diagonal matrix (make explicit by the left exponent ${}^{\Delta}\Gamma$) or as a vector (denoted ${}^{\vee}\Gamma$).

Consider now a slight change of configuration Δq from the spring equilibrium. We search for the equivalent spring reflected at a given contact point. The change Δq leads both to a change of contact position and contact forces. The change of position is directly:

$$\Delta x = J\Delta q$$

The change of force is expressed from the steady-state force-to-torque equation $\tau = J^T f$. Plugging both equations, we get:

$$\Delta x = J\Delta q = J\,{}^{\Delta}\Gamma\tau = J\,{}^{\Delta}\Gamma J^T f$$

The apparent compliance can then be defined by:

$$\gamma = J\,{}^{\Delta}\Gamma J^T$$

where γ is a square symmetric matrix.

2) Quadratic solution: Now assume that we require a specific apparent compliance γ^*. The configuration compliance that best matches this reference can be expressed as the solution of the Frobenius problem:

$$\min_{{}^{\Delta}\Gamma\text{diagonal}} \|J\,{}^{\Delta}\Gamma J^T - \gamma^*\|_F$$

where $\|.\|_F$ denotes the Frobenius norm. This is a quadratic problem, since the square Frobenius norm is the sum of square of the matrix coefficients and since $J\,{}^{\Delta}\Gamma J^T$ is linear if Γ.

We denote by ${}^{\vee}.$ the matrix-to-vector operator stacking all the columns of the input matrix[2]. The Frobenius problem can be explicitly rewritten as a quadratic problem:

$$\min_{{}^{\vee}\Gamma} \|\mathcal{J}\,{}^{\vee}\Gamma - {}^{\vee}\gamma^*\|_2$$

where $\mathcal{J}$ is the unique matrix such that $\mathcal{J}\,{}^{\vee}\Gamma = {}^{\vee}(J\,{}^{\Delta}\Gamma J^T)$. This operator has the following properties:

$$\mathcal{J}\,{}^{\vee}\Gamma = J\,{}^{\Delta}\Gamma J^T \tag{3}$$

$$\mathcal{J}^T\,{}^{\vee}\gamma = {}^{\Delta}(J^T\gamma^* J) \tag{4}$$

where ${}^{\Delta}.$ denotes the matrix-to-vector operator that selects the diagonal elements of the input matrix. Finally, the square of $\mathcal{J}$ is the Hadamard square of the square of J:

$$\mathcal{J}^T\mathcal{J} = [\![J^T J]\!] \tag{5}$$

where $[\![A]\!] = A \circ A$ denotes the element-wise (Hadamard) square of A.

The solution to the unconstrained quadratic problem is given by the pseudo-inverse:

$${}^{\vee}\Gamma = \mathcal{J}^+\,{}^{\vee}\gamma^*$$

Since for any A, $A^+ = (A^T A)^+ A^T$ and using (5), this last form can be rewritten:

$${}^{\vee}\Gamma = [\![J^T J]\!]^+\,{}^{\Delta}(J^T\gamma^* J) \tag{6}$$

[2]The operators ${}^{\vee}.$ is abusively applied to pass from the diagonal matrix ${}^{\Delta}\Gamma$ to the vector of the diagonal elements ${}^{\vee}\Gamma$, to keep the notations simple.

3) Hierarchical solution: Among all the solutions that fit at best in the least-norm-2 sense the reference γ^*, this last solution is the one of least norm-2 Γ. As usual for quadratic system, the redundancy can be made evident by expressing all the possible solutions using the projector on the null space of $\mathcal{J}$:

$${}^{\vee}\Gamma = [\![J^T J]\!]^+\,{}^{\Delta}(J^T\gamma^* J) + \mathcal{P}\,{}^{\vee}\Gamma_2$$

where $\mathcal{P} = I - [\![J^T J]\!]^+[\![J^T J]\!] = ker[\![J^T J]\!]$ and ${}^{\vee}\Gamma_2$ is any configuration compliance that can be used to satisfy a second objective.

Assume now that several compliance objectives $\gamma_1^*, \ldots, \gamma_p^*$ ordered in a hierarchy are given. Using the classical redundancy scheme detailed in [8], the configuration compliance matching at best in the last-square sense the hierarchy of the γ^* is Γ_p obtained by the following iteration:

$$\begin{aligned} {}^{\vee}\Gamma_k = {}& {}^{\vee}\Gamma_{k-1} \\ & + (\mathcal{P}_{k-1}[\![J_k^T J_k]\!]\mathcal{P}_{k-1})^+\,{}^{\Delta}\big(J_k^T(\gamma_k^* - J_k\,{}^{\Delta}\Gamma_{k-1}J_k^T)J_k\big) \end{aligned} \tag{7}$$

with $\mathcal{P}_k$ the projector onto the null space of the stacked k first Jacobian, that can be computed following the recurrence proposed in [16].

4) Constrained system: All the development above holds for a fully-actuated unconstrained system, and by direct generalization to the case when $SN_S S^T$ is invertible. In the general case, the relation between a change of elastic torques $\tau = {}^{\Delta}\Gamma\Delta q$ and contact force f is correlated with the support N_S:

$$(SN_S)^T\tau = J^T f$$

The displacement Δq cope with the rigid contact and thus is such that:

$$\Delta q = N_S\Delta q$$

since N_S is a projector onto the kernel of J_S. The compliance relation is then written:

$$\gamma = J^T(SN_S)^T\,{}^{\Delta}\Gamma SN_S J$$

The same previous developments then still hold, with direct adaptations. The operational inverse compliance is then:

$$\begin{aligned} {}^{\vee}\Gamma_k = {}& {}^{\vee}\Gamma_{k-1} \\ & + (\mathcal{P}_{k-1|S}[\![J_{k|S}^T J_{k|S}]\!]\mathcal{P}_{k-1|S})^+ \\ & \qquad {}^{\Delta}\big(J_k^T(k_k^* - J_k\,{}^{\Delta}\Gamma_{k-1}J_k^T)J_k\big) \end{aligned} \tag{8}$$

with $J_{k|S} = SN_S J_k$ and $\mathcal{P}_{k|S}$ the projector onto the stacked $J_{k|S}$ Hadamard squares.

III. Model predictive control

In this section, we quickly recall the optimal control method that we used to drive the compliant actuators.

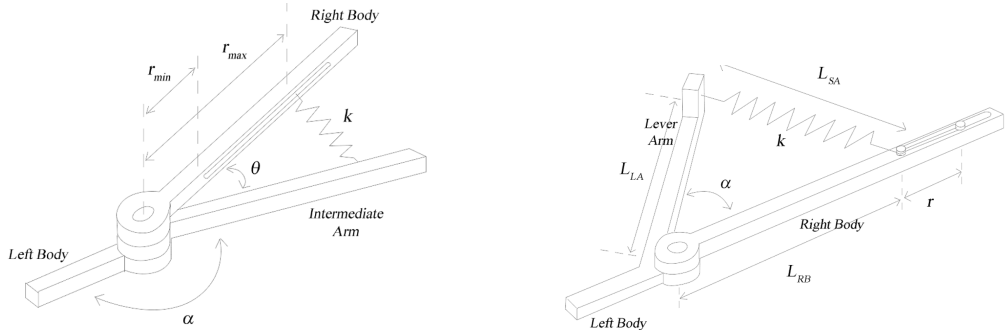


Fig. 1. Schema of the two actuator models used in the experiments.

A. Principles and model

Consider a generic system, with state variable x and control variable u, defined by its discrete time evolution function:

$$x_{t+1} = f(x_t, u_t, t) \tag{9}$$

where f is the system evolution function and the time variable t is considered discrete. Optimal control aims at computing the control and state trajectory by minimizing a given cost function:

$$\min_{X,U} \sum_{t=0}^{T-1} l_t(x_t, u_t) + l_T(x_T)$$

where T is the preview interval length (fixed), $U = (u_0 ... u_{T-1})$, $X = (x_0, ..., x_T)$ are the control and state trajectories and l_t and l_T are the running and terminal costs functions. The optimal control problem is to be solved under the constraint that (9) is satisfied. In practice, the problem is solved for X only or U only, the other variable being deduced from the dynamic equation. The solution is said explicit when computing X and implicit when computing U. The optimal solution for a linear dynamics $x_{t+1} = F_x x_t + F_u u_t$ and a quadratic cost is given by Riccati equations as a linear-quadratic regulator (LQR).

Model predictive control (MPC) is an advance control technique to control a given system by optimizing its predicted evolution. It relies on the systematic evaluation of the optimal control of the system with respect to a reference cost function, while only the first few steps of the optimal trajectory are actually executed by the actuators before its complete re-evaluation. The main interest of MPC is the ability of dealing with non-linear systems whose instantaneous linearization is not meaningful.

B. Differential dynamic programming

The Differential Dynamic Programming (DDP) is an iterative algorithm to solve a non-linear continuous optimal control problem using implicit formulation [5]. It is nearly equivalent to the application of a Newton descent algorithm. As in the Newton descent, the main idea is to approach a local optimum by iteratively modifying a candidate solution until stabilization of the evaluation of the cost function. It starts with initial state and control trajectories, typically obtained from the integration of the zero control if no other prior is available. The algorithm then iterates in two stages. It first computes an approximate quadratic model of the current candidate trajectory and compute the corresponding LQR (backward loop). Then the candidate is modified following the LQR (forward loop).

IV. Variable Stiffness Actuators

The previous section recalled a generic MPC framework. We now propose a solution to implement it, to efficiently control the behavior of a variable-stiffness actuator.

A. Principles

The operational-space approach provides the reference joint behavior (torques and/or stiffness) from operational references set by the user. The computation is typically expensive, since many DOF are involved and the whole-body operations scale with the cube of the number of joints. We can expect them to be computed at a middle dynamics (e.g. 100Hz). For in contact movements, the actuators have to react much faster to track this reference behavior (typically, 1kHz is advised [17], 3kHz is implemented on the LWR robot [10] while 10kHz are available on the iCub [2]). The joint references are then tracked by a DDP-based MPC whose cost function is described below.

B. Actuator generic model

We consider electrical-based variable stiffness actuators, where two motors in parallel are connected in series with some mechanical springs. This generic hypothesis is valid for a large class of actuators [18], [19] (see Fig. 1). While the general approach is still valid, this precise hypothesis cannot cover for example pneumatic actuators [20], [21]. The state variables are then reduced to the spring positions and velocities. Typically, two motors are involved:

$$x = (x_1, x_2, \dot{x}_1, \dot{x}_2)$$

If assuming linear (or at least static) models of the spring, the output torques can be deduced from the state.

We additionally make the hypothesis that the motor is capable of tracking acceleration references. While the electrical motors are typically driven in electrical power (current

or voltage), a first close loop is typically configured on the feedback of a output position (angular) sensor. Moreover, a mechanical reductor is often added in series with the motor, which acts as a dynamic screen preventing the motor to feel the joint dynamics. These two facts make the acceleration an easy but realistic control input. The control is then simply:

$$u = (\ddot{x}_1, \ddot{x}_2)$$

C. Cost function

We consider the following terminal cost function:

$$l_T(x_T) = w_T(\tau(x_T) - \tau_T^*)^2$$

The current torque is a function of the spring states, while τ_T^* is given by the operation-space computations. The running cost function is:

$$l_t(x_t, u_t) = w_\tau(\tau(x_t) - \tau_t^*)^2 + ||u_t||_Q^2 + w_j j(x_t)$$

where τ_t^* is typically equals to τ_T^*, j is a barrier function enforcing the joint limits and w_f, w_τ, w_j and Q are arbitrary gains that encompass the unit differences and the relative importance of the terms. The first term enforces the MPC to track the reference. The second term penalizes strong internal movements and regularizes the numerical solver behavior. The last term enforces any given constraints[3]. The barrier function is constructed to be positive, continuous and twice derivable, to value 0 on at ϵ from the limit and $+\infty$ on the limits.

An additional term $w_\Gamma(\Gamma(x_t) - \Gamma_t^*)^2$ can be added in both l_t and l_T to track a reference output stiffness, also function of the state. If no good reference stiffness is available, w_Γ is set to 0. The stiffness is then locally tuned by the optimal controller in answer to the actuator demanded dynamics. A comparison with and without reference stiffness is given in the next section.

D. Discussion

The obtained control scheme may be compared to the optimal controller described in [22]. In this paper, the authors control a torque-driven actuator to enforce an output position on an AwAS actuator [18]. They deployed a LQR by linearizing their actuator. This is possible since the torque-input position-output AwAS is linear for small spring deflection. For that mean, they have to separate the model between one main actuator driven the output motion and one stiffness actuator tuning the apparent stiffness. They finally proved the stability of their control scheme and proposed a theoretical study of its behavior that was used to automatically adapt the cost-function weights.

Our approach is more generic since no separation is needed and non-linear actuators may be considered, for example the MACCEPA. Moreover, if the stiffness is free, the controller will use it to reduce the oscillations of the output movement. Using the LQR formulation behind the DDP, the stability of our MPC can be derived using the same reasoning than in [22]. It seems more difficult to adapt the pole study for tuning the weights in general, since the poles vary with the system non-linearities. However, the same study can be directly applied when the non-linearities can be neglected (e.g. for the AwAS at steady-state).

[3]The DDP is efficient partly because unconstrained. The barrier function is an efficient solution to enforce important terms to be treated as constraints.

V. Results

We present in this section two set of simulations. In a first time, we checked the behavior of the MPC for a single actuator using the two actuator models AwAS and MACCEPA. The AwAS has a nearly linear dynamics, which is interesting to validate the LQR behavior, while the MACCEPA is less linear, which validates the capabilities of the DDP to handle a more complex cost landscape. We then produce a complex whole-body movement for a humanoid robot whose joint references are tracked by the MPC.

A. Single actuator

1) Set up: The simulations are made on the software Matlab. The model dimension are chosen to fit the requirements of a full size humanoid robot (for typical movements recorded on HRP-2). The control sampling frequency is 10kHz: the DDP is recomputed at 10kHz and the first sample is applied on the motor models. The preview windows lasts 2.5ms (that is to say 25 samples). The number of DDP iteration is bounded to 50 (while AwAS generally converges in less than 25). The MPC is used for tracking a smooth and a pulse-train reference without disturbance, and the behavior is then checked for external disturbances.

2) Torque reference tracking: DDP control has been tested for two scenarios : a pulse train and a polynomial. The response plot are given in Figures 2 and 3.

The first reference torque is a polynomial of degree three (Fig. 2). For both actuators, output torque is properly tracked. In the absence of discontinuity, DDP provides excellent control, no matter with the complexity of cost function.

The pulse-train amplitude is set to 10 Nm (calibrated on the knee torques during humanoid dynamics locomotion task). The AwAS is very reactive, reaching quickly 97% of the desired value and is stable. MACCEPA model, results are more mixed, with slower convergence time. These differences are explained by the different level of complexity of the two dynamics, excited by the train pulse that generates singularities. It is representative of a shock during an explosive task. The MACCEPA behavior might be improved by adjusting the cost weights, for example using a deeper analysis of the close-loop system.

3) Robustness: The robustness is tested for a constant reference. We let the system converge and then apply a static force of 5 Nm at the actuator output. The result is summarized in Fig. 4. The response time before new stability of the torque generated by the spring tension is of the same order as in the case of train pulse. The system naturally comes back to the reference position after a small adaptation time.

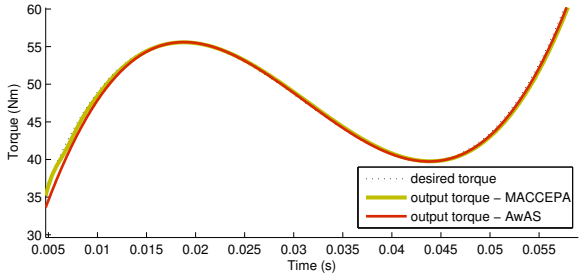


Fig. 2. Simulation A.1: Output torques with AwAS and MACCEPA actuators for a polynomial control

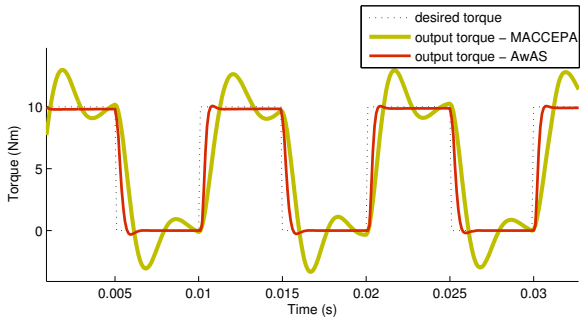


Fig. 3. Simulation A.2: Output torques with AwAS and MACCEPA actuators for a pulse train control

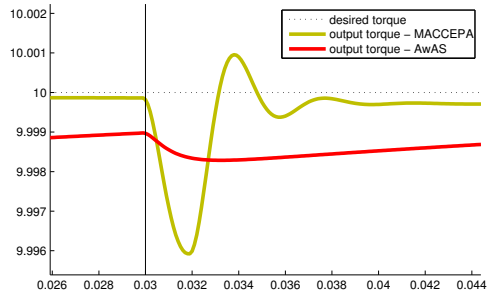


Fig. 4. Simulation A.3: Effect of 5 Nm perturbation torque at time t = 30 ms on a constant control

B. Whole-body movement

1) Set up: We consider a jumping movement with a model created from the HRP-2 robot [23]: the body specifications have been kept has the actuator dimensions, but we have replaced in simulation the stiff actuation by a set of variable stiffness actuators. The movement is defined by a sequence of tasks using the jumping methodology proposed in [24]. The motion is composed of three phases: starting from a low squatting posture, the robot violently accelerates its center of mass (CoM) until the legs are stretched while keeping a controlled (typically zero) angular momentum. The robot then floats while its CoM decelerates due to the gravity. During the flight phase, only the orientations of the feet are controlled to anticipate the landing. The robot finally impacts the ground and gently decelerates its CoM to recover its rest posture. The joint references are computed at 200Hz. One MPC at each joint then tracks this reference at 10kHz following the same set-up as in the previous section.

2) Torque tracking: The motion is summarized in Figures 5 to 8. Snapshots of the movement are gathered in Fig. 5. The whole body movement is not detailed by lack of space. The joint torques are tracked by an AwAS on the hip and a MACCEPA on the knee (the choices are arbitrary). We spotted torques on joints of a humanoid robot during a simulation of jump on site. The tracking performances are displayed in Fig. 6. The explosive movement is properly tracked. The reference torque at the impact is 3 Nm for the hip, and the MPC produces 3.3 Nm (91%). At the knee, the reference torque is 10 Nm while the MPC produces 10.8 Nm (108%). The delay is 15 ms which corresponds to the spring dynamics. This result proves the interest of using the spring actuation with MPC for achieving complex whole-body dynamic movements.

3) Stiffness selection: The joint compliance is computed from the a reference operational compliance. We set up a compliance of 2000 N/m on the z-axis of each foot, with a compliance of 40000 N/m on the two lateral rotations and stiff coupling between the translation and the rotations. The obtained joint stiffness of one leg is plotted in Fig. 7. It is perfectly tracked by the MPC. When the stiffness is let free, the MPC uses this additional DOF to optimize the output torque while minimizing the input. It is evident on the single-actuator pulse train displayed in Fig. 8. In particular, the stiffness changes at each new pulse and constantly decreases (the spring position increases) when the actuator position stabilizes.

VI. Conclusion

In this paper, we have proposed an original approach to control the whole-body movement of complex variable-stiffness robot. The operational-space approach is first used to compute the joint torques and stiffness from operational force, acceleration and compliance references. An MPC controller is then used to control the movement of each elastic actuator to track the joint references. The global approach can be seen as an efficient suboptimal to the yet untractable whole-body optimal control problem. It is very appealing to apply torque references on electrical actuators while avoid an excessive mechatronic cost.

More particularly, we have proposed an original solution to compute the joint compliance from operational references. We have proposed an efficient MPC solution to track the joint references, both torques and stiffness, using DDP. This method can be seen as a generalization to a large class of actuator and to non-linear situations of the LQR proposed in [22]. Finally, we have proposed a complete validation in simulation, by controlling the jump of a variable-stiffness full-size humanoid robot. Behind the application to a physical robot, the next step of our work is to exhibit the theoretical properties of our controller, in particular its stability.

Acknowledgments

The work has been partly supported by the grant ANR 13-CORD-002-01 Entracte.

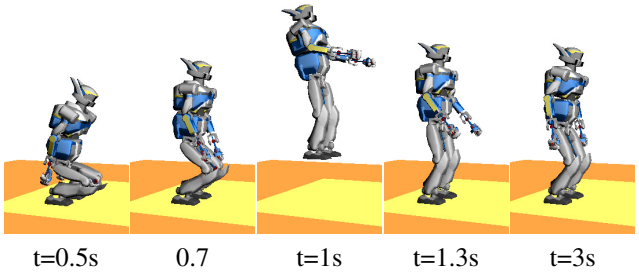


Fig. 5. Simulation B: Snapshots of the jumping movement.

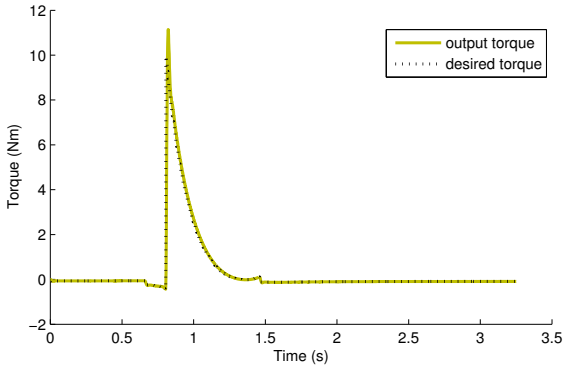


Fig. 6. Simulation B: Output torque with AwAS for a jump on site, on the hip

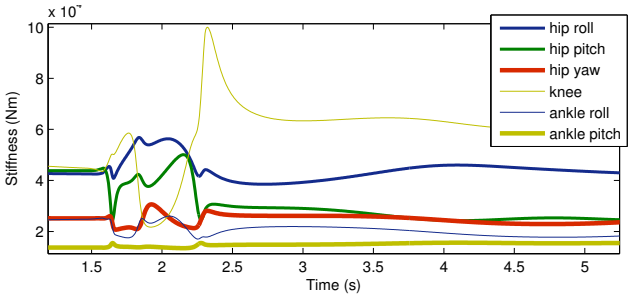


Fig. 7. Simulation B: Reference stiffness of the right leg during the jump.

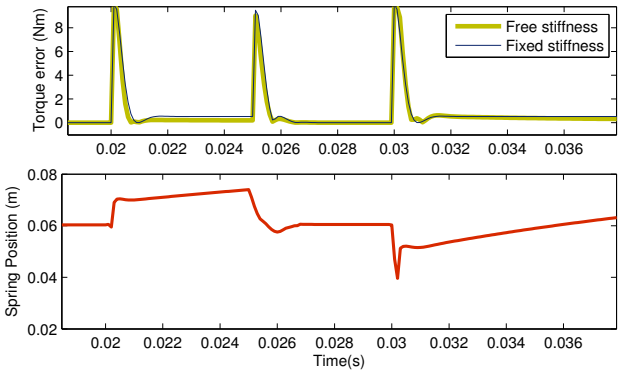


Fig. 8. Simulation B: Advantage of the free stiffness (top) Output torque error with free and constant stiffness. (bottom) Spring motion when the stiffness is free.

References

[1] G. Pratt and M. Williamson, "Series elastic actuators," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'95)*, 1995, pp. 399–406.

[2] N. Tsagarakis, Z. Li, J. Saglia, and D. Caldwell, "The design of the lower body of the compliant humanoid robot iCub," in *IEEE Int. Conf. on Robotics and Automation (ICRA'11)*, Shangai, China, May 2011.

[3] C. English and D. Russell, "Implementation of variable joint stiffness through antagonistic actuation using rolamite springs," *Mechanism and Machine Theory*, vol. 34, no. 1, pp. 27–40, 1999.

[4] N. Mansard, O. Stasse, F. Chaumette, and K. Yokoi, "Visually-guided grasping while walking on a humanoid robot," in *IEEE Int. Conf. on Robotics and Automation (ICRA'07)*, Roma, Italia, April 2007.

[5] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'12)*, Lisbon, Portugal, 2012.

[6] A. Enoch, A. amd Sutas, S. Nakaoka, and V. S., "Blue: A bipedal robot with variable stiffness and damping," in *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoid'12)*, Osaka, Japan, November 2012.

[7] D. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, no. 2, pp. 47–53, 1969.

[8] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *IEEE Int. Conf. on Advanced Robotics (ICAR'91)*, Pisa, Italy, June 1991.

[9] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *International Journal of Robotics Research*, vol. 3, no. 1, pp. 43–53, February 1987.

[10] A. Albu-Schaffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimbock, and G. Hirzinger, "The dlr lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: An International Journal*, vol. 34, no. 5, pp. 376–385, 2007.

[11] J. Park and O. Khatib, "Contact consistent control framework for humanoid robots," in *IEEE Int. Conf. on Robotics and Automation (ICRA'06)*, Orlando, USA, May 2006.

[12] N. Mansard, "A dedicated solver for fast operational-space inverse dynamics," in *IEEE ICRA'12*, St Paul, USA, May 2012.

[13] C. Samson, M. Le Borgne, and B. Espiau, *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.

[14] J. Park, "Control strategies for robots in contact," Ph.D. dissertation, Stanford University, USA, March 2006.

[15] L. Saab, O. Ramos, N. Mansard, P. Souères, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transaction on Robotics*, no. 2, pp. 346–362, April 2013.

[16] P. Baerlocher, "Inverse kinematics techniques for the interactive posture control of articulated figures," Ph.D. dissertation, EPFL, 2001.

[17] N. Hogan, "Impedance control," *Transaction of the ASME, Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1–24, 1985.

[18] A. Jafari, N. Tsagarakis, and D. Caldwell, "A novel intrinsically energy efficient development of a novel actuator with adjustable stiffness (AwAS)," *IEEE Transactions on Mechatronics*, vol. 18, no. 1, Jan. 2013.

[19] M. . C. A. used for Energy Efficient Hopping Robot Chobino1D, "Vanderborght b., tsagarakis n., van ham r., thorson i., caldwell d." *Autonomous Robots*, vol. 31, no. 1, pp. 55–65, 2011.

[20] B. Tondu and P. Lopez, "Modeling and control of mckibben artificial muscle robot actuators," *IEEE Control Systems*, vol. 20, no. 2, 2000.

[21] Y. Tassa, T. Wu, J. Movellan, and E. Todorov, "Modeling and identification of pneumatic actuators," in *IEEE International Conference Mechatronics and Automation*, August 2013.

[22] I. Sardellitti, G. Medrano-Cerda, N. Tsagarakis, A. Jafari, and D. Caldwell, "Gain scheduling control for a class of variable stiffness actuators based on lever mechanisms," *IEEE Trans. on Robotics*, vol. 29, no. 3, pp. 791–798, 2013.

[23] K. Kaneko, F. Kanehiro, S. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, S. Ota, and T. Isozumi, "Design of prototype humanoid robotics platform for hrp," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'00)*, Lausanne, Switzerland, October 2002.

[24] M. De Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," in *ACM SIGGRAPH'10*, July 2010. [Online]. Available: http://doi.acm.org/10.1145/1833349.1781157