Neurocomputing 72 (2009) 3624–3630

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Emerging motor behaviors: Learning joint coordination in articulated mobile robots ☆

Diego Pardo *, Cecilio Angulo, Sergi del Moral, Andreu Català

CETpD, Technical Research Center for Dependency Care and Autonomous Living, ESAII-UPC, Automatic Control Department, Technical University of Catalonia, Vilanova i la Geltrú, Barcelona, Spain

A R T I C L E I N F O

Available online 23 June 2009

Keywords:
Articulated mobile robots
Motor behaviors
Robot learning

A B S T R A C T

In this paper, we analyze the insights behind the common approach to the assessment of robot motor behaviors in articulated mobile structures with compromised dynamic balance. We present a new approach to this problem and a methodology that implements it for motor behaviors encapsulated in rest-to-rest motions. As well as common methods, we assume the availability of kinematic information about the solution to the task, but reference is not made to the workspace, allowing the workspace to be free of restrictions. Our control framework, based on local control policies at the joint acceleration level, attracts actuated degrees of freedom (DOFs) to the desired final configuration; meanwhile, the resulting final states of the unactuated DOFs are viewed as an indirect consequence of the profile of the policies. Dynamical systems are used as acceleration policies, providing the actuated system with convenient attractor properties. The control policies, parameterized around imposed simple primitives, are deformed by means of changes in the parameters. This modulation is optimized, by means of a stochastic algorithm, in order to control the unactuated DOFs and thus carry out the desired motor behavior.

1. Introduction

Articulated mobile robots (AMRs) are autonomous systems constructed to accomplish generic tasks. These platforms offer features that allow the generation of diverse types of motor behaviors, but also place restrictions on these behaviors. An initial example of this scenario is provided by the RoboCup competition [8], where a standard robot is used to play football. Research teams try to maximize the benefits provided by the kinematics and dynamics of the structure in order to perform motor behaviors better, such as running, kicking, heading the ball and goalkeeping. Hence, improving and generating motor behavior capacities is a research challenge. Some questions are: (i) Is the default gait of the robot efficient or fast enough? (ii) Is it able to jump? (iii) Can it lift a weight heavier than the factory-specified limit? Challenging questions arise about the optimization of default motor behaviors, the design of new motor behaviors and the overcoming of body constraints and limitations.

AMRs are often *underactuated* systems, i.e. not all of the degrees of freedom (DOFs) are actuated, and therefore the global dynamic balance of the system is constantly compromised. Additionally, they also are *redundant*, i.e. they have more DOFs than those needed for representing the position and orientation of the controlled element of the robot (the workspace). It is not evident how to control this kind of mechanism, and this is still an open area of research. Motivated by this challenge, we address a methodology to synthesize motor behaviors in AMRs in this paper, where we understand *motor behavior* as a human interpretation of the motions of a robot and their consequences (e.g. sitting, throwing an object or walking).

In order to restrict the problem, we shall focus on motor behaviors that may be encoded by the realization of rest-to-rest motions, i.e. motions defined by an initial and a final state with a velocity equal to zero. Nevertheless, many of the motor behaviors of AMRs can be understood as a consequence of rest-to-rest motions [3], for example reaching [5], throwing [10] and simple posture transitions. Moreover, cyclic and composed tasks may be decomposed into sequences of motions of this type [1].

Procedures that synthesize behaviors rely on the availability of specifications in terms of the workspace, which may include a complete path or simply the initial and final positions of the controlled element. These specifications may be derived from a direct human imposition or by path-planning methods, often offering kinematic solutions to the problem of achieving the

☆ This work was partly supported by the Spanish MEC project ADA (DPI2006-15630-C02-01).

* Corresponding author.
E-mail addresses: diego.pardo@upc.edu (D. Pardo), cecilio.angulo@upc.edu (C. Angulo), sergi.delmoral@upc.edu (S. del Moral), andreu.catala@upc.edu (A. Català).
URLS: http://www.upcnet.es/~upc26174 (D. Pardo), http://www.upcnet.es/~upc15838 (C. Angulo).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.01.015

desired motor behavior. Subsequently, robot control methods are required to compute torque-level actions that drive the mechanism according to the given specifications. As an inheritance from experience in the control of robot manipulators, where the element that defines the workspace is the end-effector of a joint chain with a limited motion domain, most of the approaches to robot control consider a workspace reference as input data, computed off-line or generated within the methodology of the relevant approach, starting from pure kinematic information about the solution to the task. When workspace references are followed, the torque control actions are computed with the purpose of achieving desired joint accelerations, which implies compensation of rigid-body dynamics, canceling the effects of gravity, and other nonlinearities.

Some of the approaches described in [9] consider the kinematics and rigid-body dynamics of the robot in the generation of the workspace reference, but this is usually done to establish sufficient conditions for the accomplishment of the behavior, rather than to take advantage of the kinematics and rigid-body dynamics. An example of this is evidenced in the gait control system used in the ASIMO humanoid (which has one of the best humanoid gaits developed so far), where control forces are computed to maintain balance stability during gait execution, i.e. the effects of gravity are canceled while suitable accelerations are imposed to accomplish the motion; consequently, the energy consumption is more than 15 times (scaled) the amount required during human gait [18]. However, it has been demonstrated that during human gait, not only are the dynamic effects of gravity not always canceled but also they are actually employed [14]. It seems that the current strategies to carry out a given motor behavior are well-suited to obtaining a particular solution of the problem. Thus, the space of behavior solutions is narrowed by the approach used rather than by the capacities of the robot.

However, some results using new perspectives show evidence of alternative solutions, ones that favor the execution of the motion and expand the capacities of the robot. For instance, results in [19] show that the given factory-maximum payload of an industrial manipulator (a 6-DOF PUMA-762) can be greatly increased by exploring new zones of the solution space with suitable control policies. The approach used was the formulation of a parameterized optimal control problem, where body dynamics and time ranges were stated as restrictions. Torque-level actions were found such that the payload lifted by the manipulator was much more (six times) than the load reachable by the default aggregation of path planning, workspace reference and torque control. Surprisingly, contrary to standard procedures, the resulting trajectories included singularities, letting the robot rest the payload against its structure on its way to the goal. Along the same lines, a similar result was later presented in [15], where a simple manipulator (2D, 3-DOF) accomplished a weightlifting behavior, avoiding workspace restrictions in the formulation. Besides maximizing the payload lifted, the results included quite different workspace trajectories that accomplished the same behavior.

The key attribute in both approaches was the direct connection between the desired behavior and the torque commands, i.e. the workspace requirements were almost null, leaving the system free to be modulated in order to fulfill the behavior, i.e. lift a defined weight. Both approaches use optimization as the main route; nevertheless, the analytical solution in [19] implies a detailed formulation of the problem and its restrictions, which is perfectly viable for manipulators in structured environments, but this is not the case for AMRs. On the other hand, the solution given in [15] is not analytical but numerical; it searches in the solution space using a learning algorithm, i.e. a numerical optimization of policy parameters by means of iterative evaluation of experiences. Nevertheless, its control framework, based on the coordination of lower-level PID controllers, cannot be directly extrapolated to more complex problems.

Recently, the attention given to the use of learning as a paradigm to exploit the capacity of robots has being growing. The latest publications on learning of motions by robots [7] revolve around early results on imitation [6], where the initial solution in the workspace is directly guided by a human, and afterwards the robot joints are controlled by parameterized policies that are intended to accomplish the behavior. The type of the functions used as control policies is that of dynamical systems (DSs). The optimal parameters of the policy are found using reinforcement learning (RL) [17] algorithms. Extensive work on RL algorithms suitable for computing robot control policies has been presented in [13].

In the methodology presented in this paper, we assume the availability of kinematic information equivalent to the initial and final states of the desired behavior. In contrast to the imitation approach, a reference in the workspace is not specified. Our control framework, based on local control policies at the joint acceleration level, attracts actuated DOFs to the desired final configuration; meanwhile, the resulting final states of the unactuated DOFs are viewed as a consequence of the actuated acceleration profiles. DSs are used as acceleration controllers, providing the system with these attractor properties. Additionally, the control policies are parameterized around imposed simple primitives, which may by deformed by means of changes in the parameters in order to obtain complex accelerations.

Subsequently, we present an example that provides a qualitative description of the type of problems that this paper addresses. The *standing-up* behavior illustrates those motor behaviors of underactuated systems in which dynamic balance is compromised. Fig. 1 shows the initial and final states for this behavior. Note that the behavior is enclosed by a motion where the initial and final velocities are equal to zero. The robot starts in a lying-down posture and should stand up, ending up as shown in Fig. 1b. However, gravity and other nonlinearities can influence the behavior in such a way that the robot ends up in a different state (see Fig. 1c). The achievement of desired values for the actuated DOFs is not enough for the desired behavior to be the result.

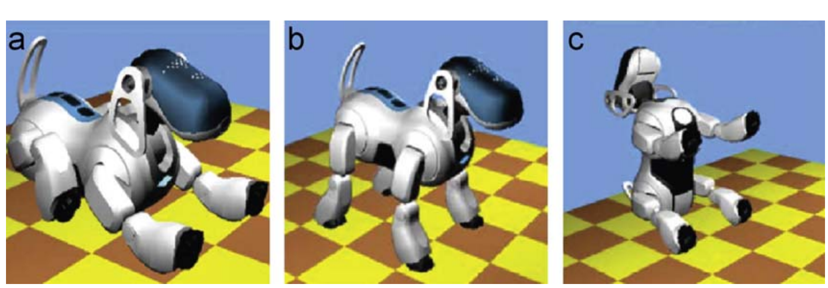

Fig. 1. (a) Initial state of the robot for the standing-up behavior. (b) Desired final configuration. (c) Undesired final configuration, where motor behavior has failed.

We present the basic definitions and a formal formulation of the problem in Section 2, and then the methodology for computing controllers is described in Section 3. A demonstration motor behavior in a simulated humanoid is synthesized by applying this methodology in Section 4. Finally, the conclusions are gathered together in Section 5.

2. Controlling robot motor behaviors

The configuration of a robot specifies the location of all parts of the mechanism with respect to a reference frame, and is given by a vector of independent position variables (generalized coordinates) $\mathbf{q} \in \mathbb{R}^n$; the number of generalized coordinates defines the number of degrees of freedom of the mechanism. The joint space is the set of possible values for the joints of the robot $\mathbf{\Theta} \in \mathbb{R}^b, b \leq n$. The state of the robot is given by the set formed by the positions and velocities of the generalized coordinates, i.e. $\mathbf{z} = [\mathbf{q}\ \dot{\mathbf{q}}] \in \mathbb{R}^{2n}$.

An element controlled by a robot also has a configuration that determines its position and orientation. This configuration defines the workspace, denoted by $\mathbf{x} \in \mathbb{R}^m$. The geometrical relation that maps the generalized coordinates to the configuration of the controlled element is known as the forward kinematics

$$\mathbf{x} = \mathbf{f_{kinem}}(\mathbf{q}) \tag{1}$$

In those cases where the number of DOFs of a robot is greater than that of the controlled element, i.e. $n > m$, the system is called *redundant* owing to the existence of more DOFs than are required for the control of the operational space.

The relation between the velocities and accelerations in the operational space and those in the configuration space is obtained from the derivative and second derivative of (1), leading to

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \tag{2}$$

where $\mathbf{J}(\mathbf{q})$ is the Jacobian matrix of $\mathbf{f_{kinem}}(\mathbf{q})$.

Now that the elements that describe the motion of a robot have been established, we now focus on the forces that generate the motion. The relation between the control vector (the applied forces) $\mathbf{u} \in \mathbb{R}^b$ and the resulting accelerations $\ddot{\mathbf{q}} \in \mathbb{R}^n$ is given by the robot dynamics, and may be written in the constrained form

$$\ddot{\mathbf{q}} = \mathbf{f_1}(\mathbf{q}, \dot{\mathbf{q}}, t) + \mathbf{f_2}(\mathbf{q}, \dot{\mathbf{q}}, t)\mathbf{u} \tag{3}$$

The system is called *underactuated* when the configuration is not able to command an instantaneous acceleration in an arbitrary direction of $\mathbf{q}$. In the case of an AMR, the assumption that all joints are actuated is coherent, but, because they are not secured to the ground, the AMR can move in directions other than those allowed by its actuators, and therefore it has more DOFs than joints, i.e. $\text{rank}[\mathbf{f_2}] = b < n$. The nonactuated DOF may be represented as virtual joints that correspond to the motion of a reference frame at the robot base with respect to the inertial frame [16], here this virtual joints are denoted as $\ddot{\mathbf{q}}_\mathbf{c} \in \mathbb{R}^c$, where $c = n - b$.

Provided that articulated robots are rigid bodies, their dynamics can be described as second-order systems where torque commands (control actions) interact with the rest of the forces acting on the mechanism. A well-known model describing this interaction is given by

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\dot{\mathbf{q}}, \mathbf{q}) \tag{4}$$

where $\mathbf{u} \in \mathbb{R}^n$ represents the vector of joint torques, $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the mass matrix that describes the inertial properties of the robot, and $\mathbf{N} \in \mathbb{R}^n$ contains nonlinear terms such as Coriolis forces and gravity.

The accelerations of the actuated DOFs $\ddot{\mathbf{q}}_\mathbf{b} \in \mathbb{R}^b$ and the unactuated DOFs $\ddot{\mathbf{q}}_\mathbf{c} \in \mathbb{R}^c$, may be considered separately without loss of generality [2]. We can expand the dynamics of the system in (4) as

$$\begin{bmatrix} \mathbf{u_b} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{M_{bb}} & \mathbf{M_{cb}} \\ \mathbf{M_{bc}} & \mathbf{M_{cc}} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_\mathbf{b} \\ \ddot{\mathbf{q}}_\mathbf{c} \end{bmatrix} + \begin{bmatrix} \mathbf{N_b} \\ \mathbf{N_c} \end{bmatrix} \tag{5}$$

where $\mathbf{u_b} \in \mathbb{R}^b$ is the vector of actuation torques, and $[\mathbf{N_b}, \mathbf{N_c}]^T$ gather together the projections of other forces (gravitational forces, Coriolis forces and ground reactions) in the corresponding subspace. The inertia matrix $\mathbf{M}$ is divided into four matrices $(\mathbf{M_{bb}}, \mathbf{M_{cb}}, \mathbf{M_{bc}}, \mathbf{M_{cc}})$, consequently relating causes and effects for actuated and unactuated accelerations.

We denote an element of the given robot posture set by $\mathscr{P}$, i.e. $\mathscr{P} \in$ {sitting, standing, arm raised, etc.}. Let us assume that a certain function P maps robot configurations $\mathbf{q} \in \mathbb{R}^n$ to the corresponding robot postures, i.e. $P : \mathbf{q} \to \mathscr{P}$. Let us denote by $\mathscr{Q}_\mathscr{P}$ the set of all configurations that represent the posture $\mathscr{P}$, i.e. $\mathscr{Q}_\mathscr{P} = \{\mathbf{q} | P(\mathbf{q}) = \mathscr{P}\}$. We define as *motor behavior* the process of taking the robot from an initial robot posture $(\mathscr{Q}_0)$ to a final posture $(\mathscr{Q}_\mathscr{F})$ within a delimited amount of time $0 < t < t_f$. It is assumed that representative elements of the initial and final posture sets are known, i.e. $\mathbf{q_0} \in \mathscr{Q}_0, \mathbf{q_f} \in \mathscr{Q}_\mathscr{F}$, including actuated and unactuated DOFs, i.e. $\mathbf{q_0} = (\mathbf{q_{0b}}, \mathbf{q_{0c}})$, $\mathbf{q_f} = (\mathbf{q_{fb}}, \mathbf{q_{fc}})$. It is assumed that all joints are actuated, i.e. $\mathbf{q_b} = \mathbf{\Theta}$.

We control actuated joints at the acceleration level using dynamical systems as policies, which allows on-line modification of acceleration commands. Provided the DSs have attractor properties, the policies are designed to attract each joint to its corresponding final state $\mathbf{q_{fb}} = \mathbf{\Theta^g}$, while it follows a smooth trajectory. In order to select an appropriate policy, we first define the following term as a *local primitive*:

$$p_i(t) = \alpha_i[-\dot{\theta}_i(t) + \beta_i(\theta_i^g - \theta_i(t))], \quad i \in \{1, \ldots, b\} \tag{6}$$

This represents the state of the velocity and position error of joint i at instant t. The parameters (α_i, β_i) locally encode the properties of the primitive. We define as a *local control policy* the combination of the local primitives involved

$$\ddot{q}_{b,i}^d(t) = \sum_{j=1}^{b} \omega_{ij} \cdot p_j(t) \tag{7}$$

where the ω_{ij} are scaling factors for the influence of the primitives j on the desired accelerations $\ddot{q}_{b,i}^d$ of individual joints. In this paper, we assume that there exists a lower-level controller that converts the desired accelerations into suitable torque commands (e.g. by the computed torque method). Therefore we assume that the actual accelerations of the actuated DOFs are given by (7).

Before continuing, and for clarification purposes, we define as a *basic policy* the case where the scaling factors are given by

$$\omega_{i,j} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \tag{8}$$

When a basic policy is used, the position and velocity errors of the joints behave as a simple damped harmonic system with an attractor in $(\theta_i^g, 0)$. The dynamics of this system are modulated when the nondiagonal weighting factors are not zero. This allows the generation of complex forms in the actual profiles followed by the joints.

The whole-body policy, generically denoted by $\ddot{\mathbf{q}}_\mathbf{b} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$, may be explicitly presented, dropping the time dependency notation, in a single expression

$$\ddot{\mathbf{q}}_b = \mathbf{W} \cdot \mathbf{A} \begin{bmatrix} \dot{\mathbf{q}}_\mathbf{b} \\ \mathbf{q_b} \end{bmatrix} \tag{9}$$

where the matrix $\mathbf{A} \in \mathbb{R}^{b \times 2b}$ is formed by the set of (α, β)-factors from the local primitives represented in (6), and the matrix $\mathbf{W} \in \mathbb{R}^{b \times b}$ gathers together the scaling factors of the policy.

With the intention of analyzing the consequences of the joint policies for the whole-body behavior, using (5) we write the resulting effects on the unactuated DOFs, generically denoted by $\ddot{\mathbf{q}}_\mathbf{c} = \mathbf{g}(\ddot{\mathbf{q}}_\mathbf{b})$, explicitly as

$$\ddot{\mathbf{q}}_\mathbf{c} = \mathbf{M}_\mathbf{cc}^{-1}[-\mathbf{M}_\mathbf{bc}\ddot{\mathbf{q}}_\mathbf{b} - \mathbf{N}_\mathbf{c}] \tag{10}$$

Note that the external forces $\mathbf{N}_\mathbf{c}$ interact with the forces related to $\ddot{\mathbf{q}}_\mathbf{b}$ and together modify the motor behavior, i.e. they affect (2) and generate workspace trajectories from the internal dynamics of the robot.

The problem of completing a motor behavior can be solved if a function $\mathbf{h}^*$ is found such that

$$\ddot{\mathbf{q}}_\mathbf{c}^* = \mathbf{g}\circ\mathbf{h}^*(\mathbf{q}, \dot{\mathbf{q}}) \tag{11}$$

accomplishing

$$\int_0^{t_f}\left(\int_0^{t_f} \mathbf{g}\circ\mathbf{h}^*(\mathbf{q}, \dot{\mathbf{q}})\, dt\right)dt \approx \mathbf{q}_\mathbf{fc} \tag{12}$$

The parameters $w_{i,j}$ in (7) must be such that above condition is satisfied. An optimization framework is presented in the following section for the establishment of a numerical methodology to find the optimal parameters of the acceleration policies and therefore solve for the motor behavior.

3. Learning control policies

The problem of finding the parameters of the acceleration policies that fulfill the motor behavior has the necessary elements for it to be formulated as a parametric optimization problem. We evaluate the quality of a given set of parameters as the objective function, by measuring the performance of the robot attempting to accomplish the task

$$R(\mathbf{W}) = \mathbf{H}(\mathbf{q}_\mathbf{b}, \dot{\mathbf{q}}_\mathbf{b}, \dot{\mathbf{q}}_\mathbf{c}, \mathbf{q}_\mathbf{c}, \ddot{\mathbf{q}}_\mathbf{b}) \tag{13}$$

where $R \in \mathbb{R}$ is a scalar that represents how the state and actions of the robot proceed for some particular parameters in the policy. Without loss of generality, we may specify a type of relation for (13). Given that the objective of the task is to take the robot's states from an initial to a final value, the euclidian distance from the resulting final states $\mathbf{q}(t_f)$ to the desired states $\mathbf{q}_\mathbf{f}$ is a valid representation of the quality of the controllers

$$R_1(\mathbf{W}) = ([\mathbf{q}_\mathbf{f} - \mathbf{q}(t_f)]^T[\mathbf{q}_\mathbf{f} - \mathbf{q}(t_f)])^{1/2} \tag{14}$$

Moreover, as the unactuated DOFs are a virtual concept used for representation purposes, an equivalent measure of the distance function is obtained if the actual and desired values of the workspace vector are measured

$$R(\mathbf{W}) = ([\mathbf{x}_f - \mathbf{x}(t_f)]^T[\mathbf{x}_\mathbf{f} - \mathbf{x}(t_f)])^{1/2} \tag{15}$$

This implies that the workspace variables $\mathbf{x}$ must be observable.

If we aim to minimize this function, the computation of the optimal parameters may be performed by iteratively calculating the steepest descent for the distance function, i.e.

$$\mathbf{W}_{\mathbf{r}+1} = \mathbf{W}_\mathbf{r} + \gamma[\nabla_\mathbf{W} R]_{\mathbf{W}=\mathbf{W}_\mathbf{r}} \tag{16}$$

where $\mathbf{r}$ denotes the iteration number, and $0<\gamma<1$ is a learning rate. The gradient of R with respect to $\mathbf{W}$ evaluated in $\mathbf{W}_\mathbf{r}$ is written as $[\nabla_\mathbf{W} R]$. However, because an analytical computation of this gradient is not viable in the current framework, we approximate the gradient based on data collected from the robot's experience, i.e. we use an instance of the Policy Gradient Reinforcement Learning (PGRL) algorithms (see e.g. [13]). The stochastic approximation of the gradient proposed in [4] is given by the expected value of a special term

$$\mathbf{G}_\mathbf{APP}(\mathbf{W}_\mathbf{r}) = \mathrm{E}\left[(b^2)\cdot\frac{R(\mathbf{W}_\mathbf{r} + \mu v_j)}{\mu}\cdot v_j\right] \tag{17}$$

where $\mathbf{G}_\mathbf{APP}(\mathbf{W}_\mathbf{r})$ represents the estimated value of the gradient in the point $\mathbf{W}_\mathbf{r}$, whereas v_j is a uniformly random unitary vector in the search space and μ is a size factor; the term (b^2) is the size of the vector of parameters. It has been demonstrated in [4] that with a suitable selection of these parameters, the gradient approximation leads to convergence at least to a local minimum.

In order to solve (17), attempts, also known as roll-outs, to perform the task must take place (with $0<j\leq J$). The resulting performance of a single roll-out is obtained using an acceleration policy with perturbed parameters, i.e. $R(\mathbf{W}_\mathbf{r} + \mu v_j)$. In practice, a single roll-out may be enough for an unbiased gradient estimate [4]. Here we present the corresponding algorithm

input: $\mu, b, \mathbf{x}_\mathbf{f}$

repeat
- select $v_j \in \mathbb{R}^{b^2}$, $\|v_j\| = 1, v_j \sim U(0,1)$
- perform roll-out with perturbed parameters $(\mathbf{W} + \mu v_j)$
- sense controlled element configuration $\mathbf{x}(t_f)$
- compute the performance $R(\mathbf{W} + \mu v_j) = ([\mathbf{x}_\mathbf{f} - \mathbf{x}(t_f)]^T[\mathbf{x}_\mathbf{f} - \mathbf{x}(t_f)])^{\frac{1}{2}}$
- estimate the gradient vector $\mathbf{G}_\mathbf{APP}(\mathbf{W}_\mathbf{r}) = \left\langle (b^2)\cdot\frac{R(\mathbf{W}_r+\mu v_j)}{\mu}\cdot v_j\right\rangle$

until gradient estimate $\mathbf{G}_\mathbf{APP}(\mathbf{W}_\mathbf{r})$ converged.

return: gradient estimate $\mathbf{G}_\mathbf{APP}(\mathbf{W}_\mathbf{r})$

where $\langle \cdot \rangle$ denotes the average of the gradient estimators.

4. Results

A simple motor behavior, named "humanoid equilibrium", was tested in a simulated humanoid (Fujitsu Hoap-2). Starting from a standing-up position, the goal of this task is to reach an equilibrium point where the humanoid stands using just one leg. The initial and final configurations of the task are shown in Fig. 2.

The humanoid uses four joints during the motion: left ankle, right ankle, hip and knee, i.e. $b = 4$. (see Fig. 3 for details). However, owing to the dynamic balance condition of the humanoid, the size of the vector of generalized coordinates needed to determine its posture and orientation (assuming no absolute displacement) is $\mathbf{q} \in \mathbb{R}^n$, $n = 7$. As mentioned in the

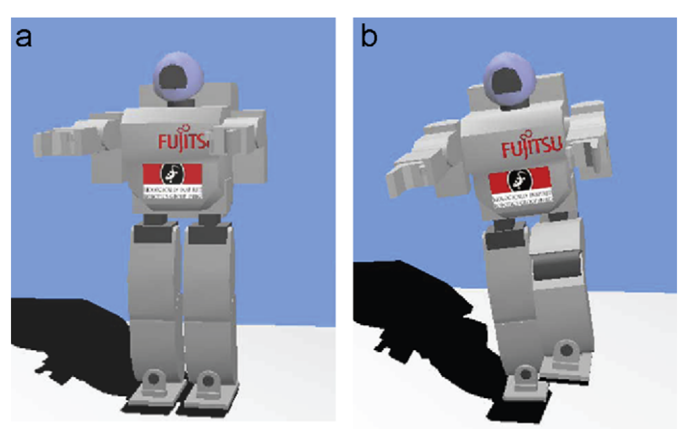

Fig. 2. Fujitsu Hoap-2 simulated humanoid used as a test bed for motion behavior synthesis. (a) Initial posture; and (b) final posture.

description of the methodology in Section 2, a goal is manually established for each actuated DOF $\mathbf{q}_{\mathbf{f,b}}$. Parameters for the acceleration primitives of joint i, i.e. α_i, β_i, are imposed to generate standard exponential trajectories, such as the one shown (dotted line) in Fig. 4. Using the basic policy defined in (8) to generate accelerations, despite the fact that the joints arrive at their desired positions, the unactuated DOFs finish in a completely different state, i.e. the robot falls down.

Using the final position of the center of mass, i.e. the workspace final position $\mathbf{x}_\mathbf{f}$, an indirect measure of the final configuration of the unactuated DOFs can be obtained. Thus, the cost function that measures whether the motor behavior is accomplished or not is given by the Euclidian distance between the initial and desired final values of the position vector of the controlled element as in (15).

One set of weights is found using the gradient-descent methodology presented in Section 3 such that the trajectories for the virtual DOFs minimize (15), and then the behavior emerges: the robot complete the motion without falling. Velocity profiles for the four actuated DOFs are shown in Fig. 5.

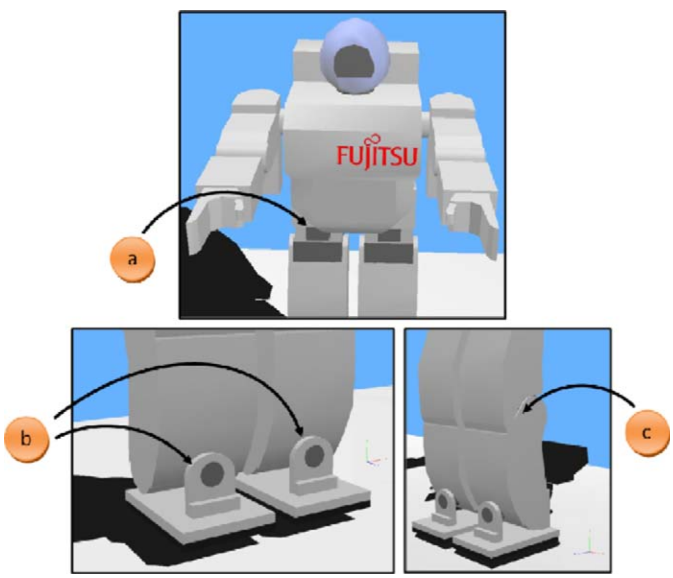

Fig. 3. Details of the joints used by the humanoid during the motor behavior. (a) hip; (b) ankles; and (c) knee.

5. Discussion and conclusions

With the increasing availability of complex mobile robots, with powerful processors and user-friendly interfaces, it seems that robots are ready to leave the research laboratory to move into the space of human daily life. Locomotion systems that do not use wheels are now being offered more frequently in standard platforms available on a large scale. This presents a challenge for the generation of goal-oriented motions.

Those motions which may be categorized by an interpretation in terms of their intention have been labeled here as motor behaviors. Examples of this type of motion include weightlifting, high-jumping, hammer throwing and sitting. Usually, robot motions are controlled to solve problems with reference to the workspace, not to solve problems of behaviors. Reference to the workspace seems to break the relation between the capacities of robots and the objectives of motions. Here we have analyzed this phenomenon and present a different perspective on the solution of the problem of motor behaviors.

In our perspective, we claim that restrictions in the workspace are eliminated, and that simple kinematic information can serve as a unique guide for the actuated DOFs of the robot. Nevertheless, the nonactuated DOFs depend directly on the accelerations generated by the powered DOFs, i.e. the joints.

Starting from the premise that the final position of the actuated joints is known, we generate accelerations such that each joint arrives at its final destination. A dynamical system with convenient attractor properties becomes very well-suited for this objective; therefore we define control policies, at the acceleration level, which are a set of individual DSs that take individual joints to the desired state. However, knowing that a basic joint acceleration profile is not enough to generate the nonactuated accelerations that solve the problem of motor behavior, we parameterize the local DSs; the parameters modulate the shape of the acceleration without dropping the attractor properties. The modulation is imposed in the form of a weighted interconnection between the outputs of the DSs involved. That is, the type of shape that a local acceleration policy may provide depends on the current state of the other policies. We call this effect "coordination" of the control policies.

The control framework has been designed such that the quality of the global control policy may be measured by the distance between the desired final state of the unactuated joints and the resulting state obtained using the policy that is being evaluated.

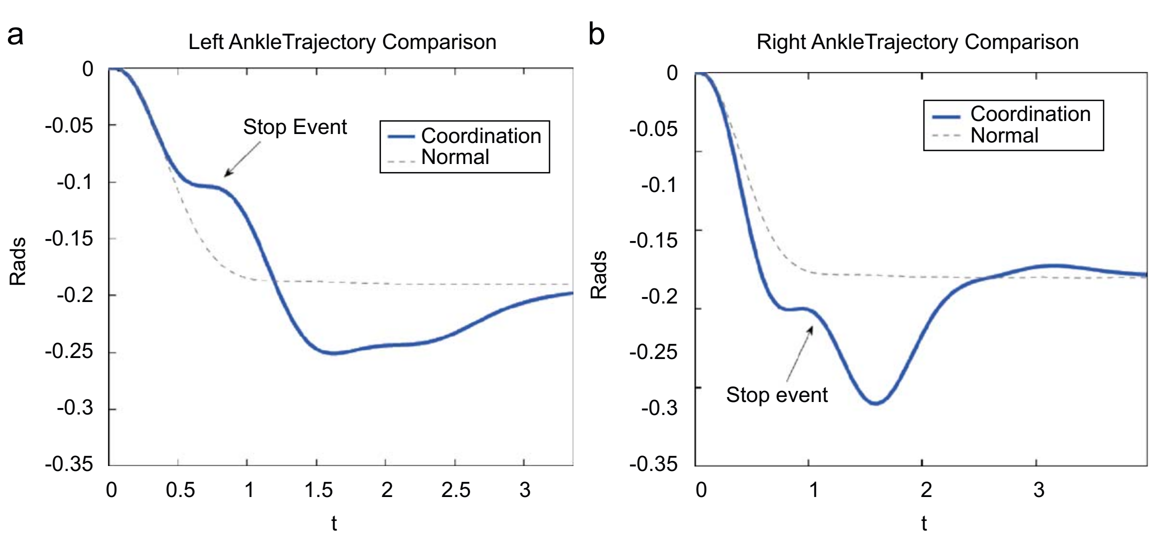

Fig. 4. (a) Left and (b) right ankle joints. Trajectories generated by the primitives (dashed lines), and trajectories deformed by the optimized weights (solid lines).

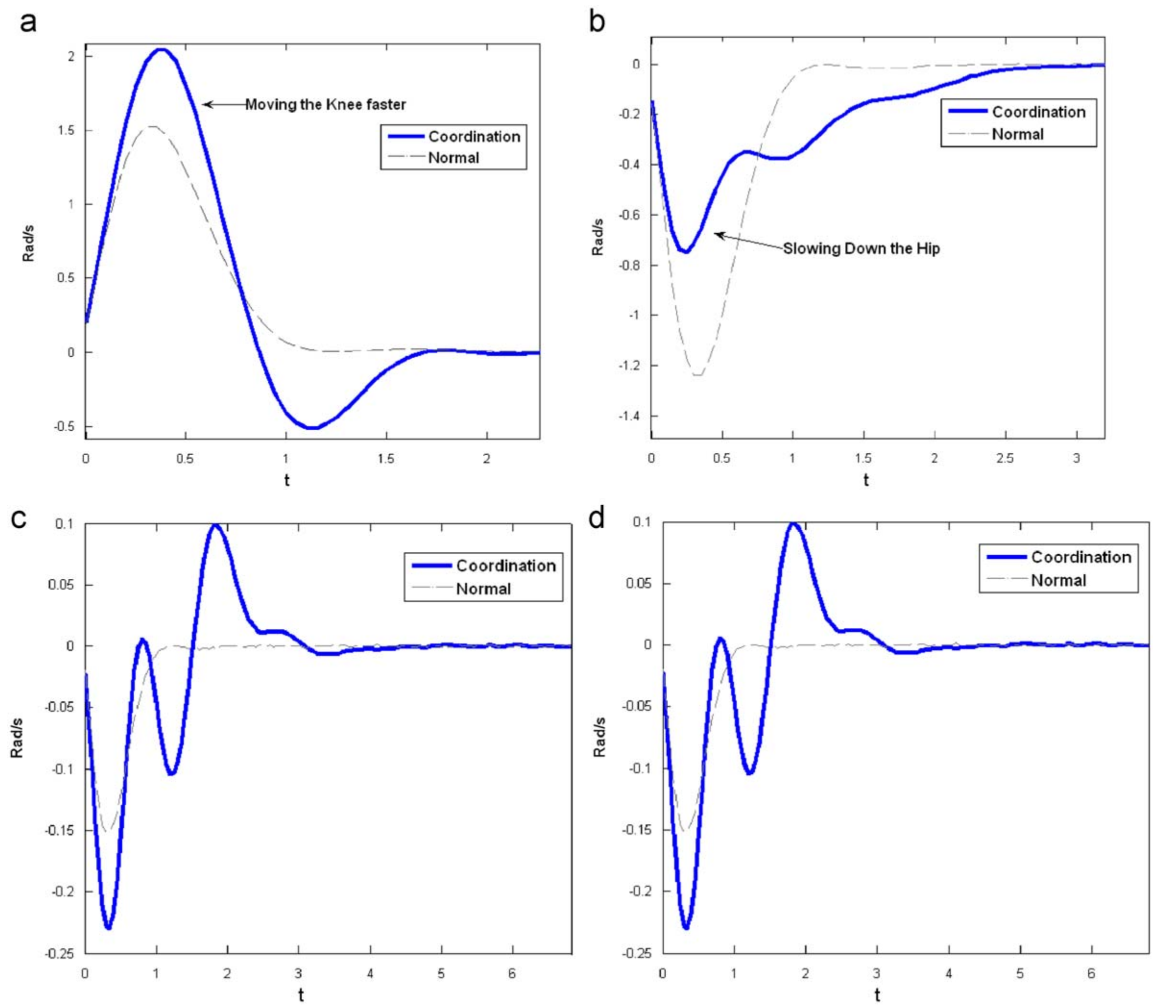

Fig. 5. Control policy output. (a) Knee; (b) hip; (c) left ankle; and (d) right ankle. Primitives (dashed lines) and deformed trajectories (solid lines).

Using this performance measurement, the policy is optimized using an iterative gradient-descent minimization scheme. The key point in this scheme is that the quality of the policy has a direct relation to its parameters; if the gradient of the distance function is known at every iteration, the direction in which the parameters should change to improve the performance is known. Unfortunately, this relation is not explicit, and an analytical solution for the gradient is not available. We therefore use a known stochastic approximation to the gradient, based on an iterative measure of the performance of the system. We have given evidence that this is a machine extrapolation of learning: in this case, learning to synthesize a motor behavior.

References

[1] D. Bullock, R.M. Bongers, M. Lankhorst, P.J. Beek, A vector-integration-to-endpoint model for performance of viapoint movements, Neural Networks 12 (1999) 1–29.

[2] S. Chun-Yi, Y. Stepanenko, Adaptive variable structure set-point control of underactuated robots, IEEE Transactions on Automatic Control 44 (1999) 2090–2093.

[3] A. De Luca, G. Di Giovanni, Rest-to-rest motion of a two-link robot with a flexible forearm, in: Proceedings of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, vol. 2, 2001, pp. 929–935.

[4] A.D. Flaxman, A.T. Kalai, B.H. Mcmahan, Online convex optimization in the bandit setting: gradient descent without a gradient, in: SODA'05: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005, pp. 385–394.

[5] S.C.F. Guenter, M. Hersch, A. Billard, Reinforcement learning for imitating constrained reaching movements, in: RSJ Advanced Robotics, Special Issue on Imitative Robots, 2007, pp. 1521–1544.

[6] A. Ijspeert, J. Nakanishi, S. Schaal, Trajectory formation for imitation with nonlinear dynamical systems, in: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Malu, Hawaii, 2001.

[7] J. Kober, B. Mohler J. Peters, Learning perceptual coupling for motor primitives, in: International Conference on Intelligent Robot Systems (IROS), 2008.

[8] E. Lakemeyer, G. Sklar, D. Sorrenti, T. Takahashi (Eds.), RoboCup 2006: Robot Soccer World Cup X, in: Lecture Notes in Artificial Intelligence, vol. 4434, Springer, Berlin, 2007.

[9] S.M. LaValle, Planning Algorithms, Cambridge University Press, Cambridge, 2006.

[10] G. Lawrence, N. Cowan, S. Russel, Efficient gradient estimation for motor control learning, in: Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI), 2003.

[13] J. Peters, Machine Learning for Robotics: Learning Methods for Robot Motor Skills, VDM Verlag Dr. Müller, Saarbrücken, 2008.

[14] H. Pontzer, A new model predicting locomotor cost from limb length via force production, Journal of Experimental Biology 208 (8) (2005) 1513–1524.

[15] M.T. Rosenstein, Learning to exploit dynamics for robot motor coordination, Ph.D. Thesis, University of Massachusetts, Amherst, May 2003.

[16] L. Sentis, Synthesis and control of whole-body behaviors in humanoid systems, Ph.D. Thesis, Stanford University, July 2007.

[17] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.

[18] R. Tedrake, Underactuated robotics: learning, planning, and control for efficient and agile machines, Course Notes, Massachusetts Institute of Technology, 2008.

[19] C.-Y. Wang, W. Timoszyk, J. Bobrow, Weightlifting motion planning for a puma 762 robot, in: Proceedings of Conference on International Robotics and Automation (ICRA), 1999.

Diego Pardo received the B.Sc. degree in Electrical Engineer from the Colombian School of Engineers in Bogota, Colombia (1996), and the M.Sc. degree in Electrical Engineering from the Universidad de los Andes, Bogota (2002). Since 2004, he is a Ph.D. student in Control, Vision and Robotics at the Technical University of Catalonia (UPC)—Automation Control Department (ESAII).

Cecilio Angulo has a B.Sc./M.Sc. in Mathematics from the University of Barcelona (1993) and a Ph.D. in Sciences from the Department of Automatic Control of the Technical University of Catalonia (UPC) (2001, with Dr. Andreu Català). In September 1994, I was appointed Assistant Professor in the department of Informatics and Applied Mathematics at University of Girona. In September 1998, I joined the Department of Automatic Control at UPC as an Assistant Professor. I carried out a predoc at LP2A - Université de Perpignan (1999) with Dr. Monique Polit, a postdoc at IIIA-CSIC (2004) with Dr. Pere Garcia and a postdoc at DIBE Univeristà degli Studi di Genova (2006) with Dr. Davide Anguita. Professional position was promoted to Associate professor in October 2007.

Sergi del Moral received the B.Sc. degree in Mathematics from the Universitat Autònoma de Barcelona (UAB), Spain, in 2007, and currently he is working toward the Ph.D. degree in the Automatic Control Department from the Technical University of Catalonia (UPC), Spain (http://www.epsevg.upc.edu/smoral/).

Andreu Catala received the M.Sc. degree in Physics from the University of Barcelona (1980) and a Ph.D. in Sciences from the Department of Automatic Control of the Technical University of Catalonia (1993). Associated professor at the same university from 1997. Head of the Knowledge Engineering Research Group (GREC) and co-director of the Technological Research Center for Dependency Care and Autonomous Living (CETpD). Main research interest are computational intelligence and qualitative reasoning applied to the improvement of the quality of life of people and communities (http://www.upc.edu/cetpd).