DOI 10.1007/s10514-007-9070-7

Reaching with multi-referential dynamical systems

Micha Hersch · Aude G. Billard

Received: 27 October 2006 / Accepted: 3 December 2007

Abstract We study a reaching movement controller for a redundant serial arm manipulator, based on two principles believed to be central to biological motion control: multi-referential control and dynamical system control. The resulting controller is based on two concurrent dynamical systems acting on different, yet redundant variables. The first dynamical system acts on the end-effector location variables and the second one acts on the joint angle variables. Coherence constraints are enforced between those two redundant representations of the movement and can be used to modulate the relative influence of each dynamical system. We illustrate the advantages of such a redundant representation of the movement regarding singularities and joint angle avoidance.

Keywords Bio-inspired reaching · Dynamical system control · Multi-referential control · DLS inverse · Redundant manipulator control · Joint limit avoidance · Singularity avoidance

1 Introduction

After many decades of progress and significant achievements in the field of robot control, biological movement control remains a model of efficient and adaptive control which we cannot compete with. The mechanisms underlying biological control and accounting for its extraordinary properties are still largely unknown, although many attempts have been made to uncover them. In this paper, we focus on two principles that have been suggested to be at the core of biological motion control, and apply them to control the reaching motions of a robotic arm. The first principle is multi-referential control, i.e., the idea that movements are represented and controlled in multiple frames of reference. The second principle is dynamical system control, the idea that control policies are best represented in terms of a dynamical system active in a given state space.

Taking those two principles drawn from the study of biological systems, we design a reaching controller composed of two dynamical systems simultaneously active in different yet redundant frames of reference. This enables the system to take advantage of the properties of each representation. The resulting system offers interesting characteristics in terms of robustness to perturbations and singularities.

Dynamical systems have been used successfully to control mobile robots (Schöner et al. 1995), robotic arms (Iossifidis and Schöner 2004), humanoid arms (Ijspeert et al. 2002) and walking movements (Righetti and Ijspeert 2006). This paradigm has been recognized as a powerful tool for robust and adaptive control. The novelty of our work lies in the combination of multiple and redundant dynamical systems for the control of reaching movements.

2 Human reaching movements

2.1 Multi-referential control

In the eighties, a lot of work has been devoted to determining the frame of reference in which human reaching movements are planned. Some studies (Morasso 1981;

M. Hersch (✉) · A.G. Billard
LASA Laboratory, School of Engineering, EPFL, Station 9,
1015 Lausanne, Switzerland
e-mail: micha.hersch@epfl.ch

A.G. Billard
e-mail: aude.billard@epfl.ch

Abend et al. 1982; Flash and Hogan 1985) suggested that movements were planned according to constraints expressed in a Cartesian body-centered frame of reference, while others (Atkeson and Hollerbach 1985; Lacquaniti et al. 1986) suggested that movements are planned according to intrinsic constraints expressed in joint angle coordinates. In all cases, invariant trajectory characteristics (in particular straight lines) in a given space were taken as evidence for that space being the planning space.

After a careful review of the experimental settings of previous experiments, Desmurget et al. (1997) showed that constrained settings, where subjects were forced by a manipulandum to move in an horizontal plane induced straight spatial trajectories, whereas free settings induced straight joint angle trajectories. This work, along with other evidence (see Desmurget et al. 1998b), made it quite convincing that movements were not planned in a single frame of reference, but that many different frames of reference were involved in the planning and control of reaching motions (Paillard 1991). In particular, a hybrid visuo-kinesthetic frame of reference was hypothesized in Carrozzo and Lacquaniti (1994).

2.2 Dynamical system control

Traditionally, reaching has been (explicitly or implicitly) considered as a two-stage process (Shadmehr and Wise 2005). The first stage is the planning stage, which is followed by an execution stage. According to this view, a reaching trajectory is computed during the planning stage, and this trajectory is actually tracked during the execution stage. This is in line with the usual robotics applications, where the planner and the controller are two separate components.

This traditional view has been challenged by the dynamical system approach to movement control (Bizzi et al. 1984; Kelso 1995; Todorov and Jordan 2002). According to this approach, there is no explicit trajectory planning, but rather an implicit set of trajectories made possible by a dynamical system. So there is no planned "preferred" trajectory that the system tries to match during execution, but a set of dynamic laws that move the system from one point to another, eventually leading it to the goal. In this framework, the target acts as an attractor for the arm. This view was supported by experimental evidence on frogs showing that an attractor could be created at a leg position by micro-stimulation of the spinal cord (Giszter et al. 1993).

The exact nature of such a dynamical system is still controversial, although some suggestions have been made, such as the equilibrium-point hypothesis (Bizzi et al. 1984), the λ model (Feldman and Levin 1995), the stochastic optimal feedback control law (Todorov and Jordan 2002) or the Vector Integration To Endpoint (VITE) (Bullock and Grossberg 1988). This model is described in the next paragraph.

2.2.1 *The VITE model*

The VITE model (Bullock and Grossberg 1988) describes the neural signals commanding a pair of agonist-antagonist muscles. The target limb position T is assumed to be known. The actual limb position is given by a signal $\mathbf{P}(t)$. The model hypothesizes the existence of a "difference vector population" of neurons with activity $\mathbf{V}$ and a "go signal" $G(t)$, which gates the execution of the movement. The VITE model for a single muscle is then described by the following equations:

$$\dot{\mathbf{V}} = \alpha(-\mathbf{V} + \mathbf{T} - \mathbf{P}), \tag{1}$$

$$\dot{\mathbf{P}} = G \cdot [\mathbf{V}]^{+}, \tag{2}$$

where α is a positive constant and $[\cdot]^{+}$ indicates the positive value function (i.e. 0 if the argument is negative).

Applying this model on a pair of muscles, agonist and antagonist, and taking a step-like go function yields

$$\ddot{\mathbf{r}} = \alpha(-\dot{\mathbf{r}} + \beta(\mathbf{r}_T - \mathbf{r})). \tag{3}$$

In this equation, $\mathbf{r}$ represents the limb position under the influence of both agonist and antagonist muscles, $\mathbf{r}_T$ the target position and $\alpha, \beta \in [0\ 1]$ are constant scalars. This is the equation of the spring-and-damper system. It is straightforward to show that this dynamical system has a stable attractor on $\mathbf{r}_T$. So, from any starting point, the system will be brought smoothly to the target and remain there. Moreover, for $\alpha > 4\beta$, there is no trajectory overshoot. A critically damped solution is obtained if $\alpha = 4\beta$. Trajectories for various values of α and β are illustrated in Fig. 1. The evolution of the position $\mathbf{r}$ given by this equation is illustrated in Fig. 1 for different parameters α and β.

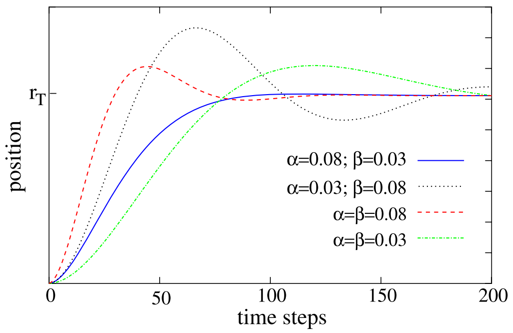


Fig. 1 The dynamics of the VITE model for various values of the parameters. Throughout this document the values $\alpha = 0.08$ and $\beta = 0.03$ were used. Refer to (3) for variable definitions

3 A multi-referential dynamical system

3.1 Overview

The idea driving the design of the robot arm controller described here is to unite a VITE-like dynamical system control with a multi-referential control. This is achieved by considering two VITE-like sub-controllers active in different spaces. The first sub-controller is active in the joint angle space (or arm configuration space) and the second sub-controller is active in the end-effector location space (or Cartesian space). Each sub-controller is specified by the VITE-like dynamical system described by (3). Those two dynamical systems concurrently act on two distinct sets of variables, robot joint angles for the first one, and robot end-effector location for the second one.

However, those sets of variables are not independent, as the end-effector location is uniquely determined by the robot arm configuration. Thus coherence constraints are enforced so that the joint angles specified by the first dynamical system correspond to the end-effector location specified by the second dynamical system. In other words, there are two redundant representations of the movement, each one with its own dynamics. Those two dynamics are coupled by coherence constraints (described in Sect. 3.3) that ensure that no contradiction between the two representations occur. The structure of this controller is shown in Fig. 2.

This coupling can then be modulated in order to adapt to the relative influence of the sub-controllers, thus allowing a smooth transition from one controller to the other. This amounts to adapting the control strategy to the situation at hand. In Sect. 3.6 this feature is exploited in order to obtain a very simple and effective joint limit avoidance mechanism.

In summary, our controller can be viewed as two linear dynamical systems coupled through non-linear coherence constraints. As such, the resulting global dynamical system is non-linear.

3.2 Setting and notations

We consider a robotic arm with n degrees of freedom (DOF) manipulating objects in a m-dimensional space. The arm configuration and the end-effector location are given by vectors $\theta \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^m$ respectively. (Here and throughout this paper, vectors are represented with bold lower case symbols.) Those vectors may be indexed by the time t, indicating that at that moment in time the manipulator is in configuration θ_t and the end-effector in location $\mathbf{x}_t$. The fact that a particular arm configuration θ_t corresponds to a particular end-effector location $\mathbf{x}_t$ is expressed by the following relationship.

$$\mathbf{x}_t = \mathbf{K}(\theta_t), \tag{4}$$

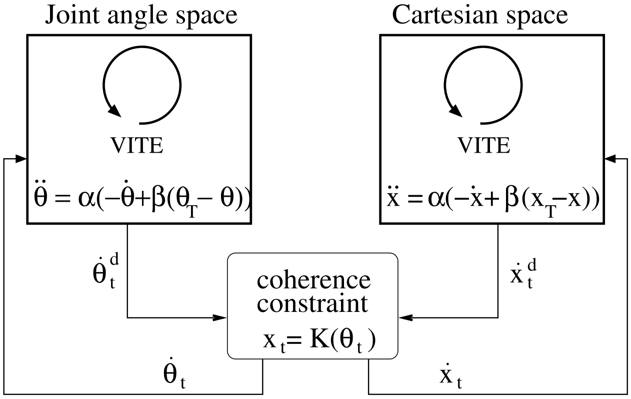


Fig. 2 A schematic representation of the system. A VITE-like dynamical system is applied to the joint angle variables θ and to the end-effector variables $\mathbf{x}$. Coherence constraints are enforced between the two sets of variables

where $\mathbf{K}$ is the kinematic function. Due to the redundancy of the manipulator, different values of θ may yield the same value $\mathbf{K}(\theta)$.

The aim of the controller is to smoothly bring the manipulator end-effector to a target location $\mathbf{x}_T$. To this location corresponds a set Θ_T of the manipulator's joint configurations:

$$\Theta_T = \{\theta | \mathbf{K}(\theta) = \mathbf{x}_T\}. \tag{5}$$

In the case of redundant manipulators, this set usually contains an infinite number of elements. However, one specific configuration θ_T must be given to the VITE-like dynamical system. We describe below (in Sect. 3.4) how this redundancy is handled.

In the remaining of this paper, the integration constant of the dynamical system is denoted by τ.

3.3 Coherence enforcement

It is clear that if one formally applies the dynamical system described in (3) to both the robot joint angles and the end-effector location, the result will bear no sense. This is due to the fact that both sets of variables (joint angles and end-effector location) will be brought to values where they do not correspond to each other. In order to avoid this, coherence constraints are enforced, which will force the joint angles and the end-effector location to be consistent with each other at all times. This is achieved by finding the position which is closest to the positions given by each of the dynamical systems, while remaining coherent. In other words, at every time step the two dynamical systems bring the manipulator to a desired position $(\mathbf{x}_t^d, \theta_t^d)$ which is not coherent. Coherence is then enforced by finding the coherent velocities $(\dot{\mathbf{x}}_t, \dot{\theta}_t)$ that will bring the system closest to $(\mathbf{x}_t^d, \theta_t^d)$.

Similarly to what was done in Billard et al. (2006), those velocities can be found by solving the following constrained optimization problem:

$$\min_{\dot{\theta}_t, \dot{\mathbf{x}}_t} \quad \frac{1}{2}((\dot{\theta}_t - \dot{\theta}_t^d)^T \mathbf{W}_\theta (\dot{\theta}_t - \dot{\theta}_t^d) + (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d)^T \mathbf{W}_{\mathbf{x}} (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d)) \tag{6}$$

$$\text{u.c.} \quad \dot{\mathbf{x}}_t = \mathbf{J}_t \dot{\theta}_t, \tag{7}$$

where $\mathbf{J}_t$ is the Jacobian of $\mathbf{K}$ at θ_t. Here the semi-definite positive diagonal matrices $\mathbf{W}_\theta \in \mathbb{R}^{n \times n}$ and $\mathbf{W}_{\mathbf{x}} \in \mathbb{R}^{m \times m}$ serve as coefficient indicating the respective weight that one should give to the desired joint angles or end-effector location, and account for the unit difference between $\mathbf{x}$ and θ. Because it is a positive quadratic optimization problem subject to linear constraints, this problem has a single minimum. It can be found using Lagrange multipliers. The quantity to be minimized can be expressed as

$$\begin{aligned} L(\dot{\theta}_t^d, \dot{\theta}_t, \dot{\mathbf{x}}_t^d, \dot{\mathbf{x}}_t) = \frac{1}{2}\big((\dot{\theta}_t - \dot{\theta}_t^d)^T \mathbf{W}_\theta (\dot{\theta}_t - \dot{\theta}_t^d) \\ + (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d)^T \mathbf{W}_{\mathbf{x}} (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d)\big) \\ - \lambda^T (\dot{\mathbf{x}}_t - \mathbf{J}_t \dot{\theta}_t), \end{aligned} \tag{8}$$

where λ is the vector of Lagrange multipliers. After differentiating with respect to $\dot{\theta}_t$ and $\dot{\mathbf{x}}_t$ and setting to zero, one gets

$$\frac{\partial L}{\partial \dot{\theta}_t} = \mathbf{W}_\theta (\dot{\theta}_t - \dot{\theta}_t^d) + \mathbf{J}_t^T \lambda = 0, \tag{9}$$

$$\frac{\partial L}{\partial \dot{\mathbf{x}}_t} = \mathbf{W}_{\mathbf{x}} (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d) - \lambda = 0 \tag{10}$$

and thus

$$\mathbf{W}_\theta (\dot{\theta}_t - \dot{\theta}_t^d) + \mathbf{J}_t^T \mathbf{W}_{\mathbf{x}} (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d) = 0. \tag{11}$$

After inserting (7) into this last equation, we can infer

$$\mathbf{W}_\theta (\dot{\theta}_t - \dot{\theta}_t^d) + \mathbf{J}_t^T \mathbf{W}_{\mathbf{x}} (\mathbf{J}_t \dot{\theta}_t - \dot{\mathbf{x}}_t^d) = 0 \tag{12}$$

and hence

$$\dot{\theta}_t = \big(\mathbf{W}_\theta + \mathbf{J}_t^T \mathbf{W}_{\mathbf{x}} \mathbf{J}_t\big)^{-1} \big(\mathbf{W}_\theta \dot{\theta}_t^d + \mathbf{J}_t^T \mathbf{W}_{\mathbf{x}} \dot{\mathbf{x}}_t^d\big). \tag{13}$$

We can notice that this expression is a generalization of the Moore-Penrose pseudo-inverse and also of the Damped Least Square (DLS) inverse. Indeed, by setting $\mathbf{W}_\theta$ to zero and $\mathbf{W}_{\mathbf{x}}$ to identity, one gets the original resolved rate control method presented in Whitney (1969), which makes use of the Moore-Penrose pseudo-inverse of the Jacobian to compute the joint velocities that produce given end-effector velocities. Moreover setting $\dot{\theta}_t^d$ to zero and $\mathbf{W}_\theta$ to identity, one gets the DLS inverse method (Chiaverini et al. 1994), which avoids singularities.

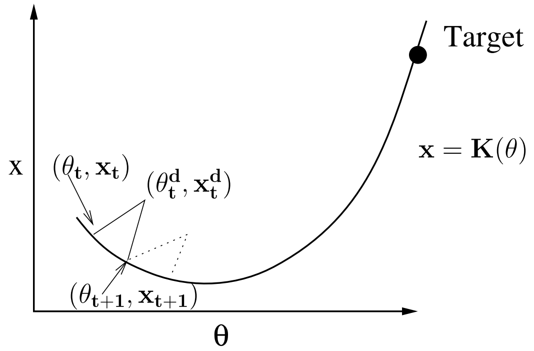


Fig. 3 Coherence enforcement between the two dynamical systems. At time t, the system is in position $(\theta_t, \mathbf{x}_t)$ which is coherent $(\mathbf{x}_t = \mathbf{K}(\theta_t))$. Each of the sub-controllers bring the system to an incoherent position $(\theta_t^d, \mathbf{x}_t^d)$, which is then projected on the closest coherent position $(\theta_{t+1}, \mathbf{x}_{t+1})$

The parameters $\mathbf{W}_\theta$ and $\mathbf{W}_{\mathbf{x}}$ control the influence of each of the sub-controllers. By setting $\mathbf{W}_{\mathbf{x}}$ to zero, one obtains a pure joint angle controller and by setting $\mathbf{W}_\theta$ to zero, the result is a pure end-effector location controller.

An alternate representation of (13) can be formulated as (see Appendix for the proof)

$$\dot{\theta}_t = \dot{\theta}_t^d + \mathbf{W}_\theta^{-1} \mathbf{J}_t^T (\mathbf{W}_{\mathbf{x}}^{-1} + \mathbf{J}_t \mathbf{W}_\theta^{-1} \mathbf{J}_t^T)^{-1} (\dot{\mathbf{x}}_t^d - \mathbf{J}_t \dot{\theta}_t^d). \tag{14}$$

Although maybe not as simple as (13), this formulation is advantageous from an implementation perspective. Using $\mathbf{W}_\theta^{-1}$ and $\mathbf{W}_{\mathbf{x}}^{-1}$ instead of $\mathbf{W}_\theta$ and $\mathbf{W}_{\mathbf{x}}$ allows to avoid infinity when dealing with a pure angular controller. It is indeed equivalent to have $\mathbf{W}_\theta^{-1}$ (respectively $\mathbf{W}_{\mathbf{x}}^{-1}$) or $\mathbf{W}_{\mathbf{x}}$ (respectively $\mathbf{W}_\theta^{-1}$) equal to zero. Moreover, this formulation is faster to compute, because it requires a matrix inversion of degree m, whereas (13) requires the inversion of a matrix of degree $n > m$ (for redundant manipulators).

Another, perhaps more intuitive, way of understanding the coherence constraints enforcement is to consider the joint angles and the end-effector location in a joint space of dimension $n + m$. In this space, (4) defines a n-dimensional manifold of consistent positions. Thus the system can be seen as a VITE-like dynamical system in this joint space, whereby the position is constantly projected on the manifold of consistent positions, as illustrated by Fig. 3. In this view, the parameters $\mathbf{W}_\theta$ and $\mathbf{W}_{\mathbf{x}}$ determine the direction of the projection.

3.4 Target configuration redundancy

As mentioned above, the joint angle dynamical system takes a target arm configuration θ_T as input. In case of a redundant manipulator, there can be an infinite number of such $\theta_T \in \Theta_T$ corresponding to a desired target location. In order to decide which one to choose, we take (at each time step) the one that is closest (in the standard Euclidean norm) to the actual arm configuration. This way, the joint displacement prescribed by the joint angle controller is minimized. If we already have a $\theta_T \in \Theta_T$, it is possible to update this value by $\tau\dot{\theta}_T$, which will bring it closer to the actual arm configuration θ.[1] Note that the target itself may be moving with velocity $\dot{\mathbf{x}}_T$. The update value $\tau\dot{\theta}_T$ can be found by solving the following constrained optimization problem

$$\min_{\dot{\theta}_T} \quad \frac{1}{2}(\theta_T + \tau\dot{\theta}_T - \theta)^T(\theta_T + \tau\dot{\theta}_T - \theta) \tag{15}$$

$$\text{u.c.} \quad \tau\mathbf{J}_T\dot{\theta}_T = \tau\dot{\mathbf{x}}_T + \mathbf{x}_T - \mathbf{K}(\theta_T). \tag{16}$$

Here, $\mathbf{J}_T$ is the Jacobian of the kinematic function at θ_T. The term $\mathbf{x}_T - \mathbf{K}(\theta_T)$ is there to avoid a numerical drift, which would bring $\mathbf{K}(\theta_T)$ away from $\mathbf{x}_T$. Again, it is a positive quadratic optimization problem subjected to linear constraints and thus has a single minimum. Using the Lagrange multipliers for solving this problem, we have

$$\begin{aligned}&\frac{d}{\partial\dot{\theta}_T}\frac{1}{2}(\theta_T + \tau\dot{\theta}_T - \theta)^T(\theta_T + \tau\dot{\theta}_T - \theta)\\ &\quad - \lambda^T\big(\tau\mathbf{J}_T\dot{\theta}_T - \tau\dot{\mathbf{x}}_T - \big(\mathbf{x}_T - \mathbf{K}(\theta_T)\big)\big) = 0.\end{aligned} \tag{17}$$

Thus

$$\begin{aligned}0 &= \theta_T + \tau\dot{\theta}_T - \theta - \mathbf{J}_T^T\lambda\\ \Rightarrow &\quad \tau\dot{\theta}_T = \mathbf{J}_T^T\lambda - \theta_T + \theta\\ \Rightarrow &\quad \mathbf{J}_T(\mathbf{J}_T^T\lambda - \theta_T + \theta) = \tau\dot{\mathbf{x}}_T + \mathbf{x}_T - \mathbf{K}(\theta_T)\\ \Rightarrow &\quad \lambda = (\mathbf{J}_T\mathbf{J}_T^T)^{-1}\big(\tau\dot{\mathbf{x}}_T + \mathbf{x}_T - \mathbf{K}(\theta_T) + \mathbf{J}_T(\theta_T - \theta)\big)\\ \Rightarrow &\quad \tau\dot{\theta}_T = \mathbf{J}_T^T(\mathbf{J}_T\mathbf{J}_T^T)^{-1}\big(\tau\dot{\mathbf{x}}_T + \mathbf{x}_T - \mathbf{K}(\theta_T)\\ &\qquad\qquad + \mathbf{J}_T(\theta_T - \theta)\big) - \theta_T + \theta.\end{aligned}$$

This results in

$$\begin{aligned}\tau\dot{\theta}_T &= \big(\mathbf{J}_T^T(\mathbf{J}_T\mathbf{J}_T^T)^{-1}\mathbf{J}_T - \mathbf{I}_n\big)(\theta_T - \theta)\\ &\quad + \mathbf{J}_T^T(\mathbf{J}_T\mathbf{J}_T^T)^{-1}\big(\tau\dot{\mathbf{x}}_T + \mathbf{x}_T - \mathbf{K}(\theta_T)\big),\end{aligned} \tag{18}$$

where $\mathbf{I}_n$ denotes the identity matrix of size $n \times n$.

This amounts to performing a gradient descent on the squared Euclidean distance to the actual arm configuration in angle space. As such, θ_T may end up in a local minimum if Θ_T is disjoint (see Burdick 1989) and if the initial θ_T is too far from the optimal one. Thus, in order to find an initial value for θ_T we sample Θ_T using a geometrical inverse kinematics algorithm and take the value closest to θ. This initialization is performed again when a sudden target displacement occurs.

3.5 Summary

Putting together the elements described above results in the following dynamical system. At each time step inconsistent velocities $\dot{\theta}_t^d$ and $\dot{\mathbf{x}}_t^d$ are obtained by each of the sub-controllers, using the Euler approximation of the VITE-like system:

$$\ddot{\theta}_t^d = \alpha(-\dot{\theta}_t + \beta(\theta_T - \theta_t)), \tag{19}$$

$$\dot{\theta}_t^d = \dot{\theta}_t + \tau\ddot{\theta}_t^d, \tag{20}$$

$$\ddot{\mathbf{x}}_t^d = \alpha(-\dot{\mathbf{x}}_t + \beta(\mathbf{x}_T - \mathbf{x}_t)), \tag{21}$$

$$\dot{\mathbf{x}}_t^d = \dot{\mathbf{x}} + \tau\ddot{\mathbf{x}}_t^d. \tag{22}$$

Coherence constraints are then enforced using (14):

$$\dot{\theta}_{t+1} = \dot{\theta}_t^d + \mathbf{W}_\theta^{-1}\mathbf{J}^T(\mathbf{W}_\mathbf{x}^{-1} + \mathbf{J}\mathbf{W}_\theta^{-1}\mathbf{J}^T)^{-1}(\dot{\mathbf{x}}_t^d - \mathbf{J}\dot{\theta}_t^d), \tag{23}$$

$$\dot{\mathbf{x}}_{t+1} = \mathbf{J}\dot{\theta}_{t+1} \tag{24}$$

$$\theta_{t+1} = \theta_t + \tau\dot{\theta}_{t+1}, \tag{25}$$

$$\mathbf{x}_{t+1} = \mathbf{K}(\theta_{t+1}). \tag{26}$$

Finally, the target arm configuration θ_T is updated using (18):

$$\begin{aligned}\theta_T &= \theta_T + \big(\mathbf{J}_T^T(\mathbf{J}_T\mathbf{J}_T^T)^{-1}\mathbf{J}_T - \mathbf{I}_n\big)(\theta_T - \theta_{t+1})\\ &\quad + \mathbf{J}_T^T(\mathbf{J}_T\mathbf{J}_T^T)^{-1}\big(\tau\dot{\mathbf{x}}_T + \mathbf{x}_T - \mathbf{K}(\theta_T)\big).\end{aligned} \tag{27}$$

Those three sets of equations are iterated until the target is reached.

3.6 Joint limit avoidance

In this section we explore an application offered by an online modulation of the sub-controller weights. We show how interesting properties can be obtained by a judicious weight modulation policy. In particular, we focus on joint limit avoidance. When controlling a robotic arm, it is always important to avoid bumping into the joint boundaries. Indeed, this is needed to avoid jerky movements and to avoid restraining the movements of the arm.

The solution presented here can be applied when the joint angle workspace is convex. This is the case when the joint angle workspace is specified by a fixed upper and lower bound for each of the joint angles. The hyper-volume defined in this way is a parallelepiped, and is hence convex. This implies that when reaching a target from a starting position, the joint angle sub-controller will never bring the system to a joint angle boundary because there is no trajectory overshoot. This property can be exploited to effectively avoid the joint boundaries, by switching to the joint angle sub-controller when approaching the joint limit. So joint angle limits can be easily avoided by making the weights

[1] Here we omit the time index t to lighten the notation.

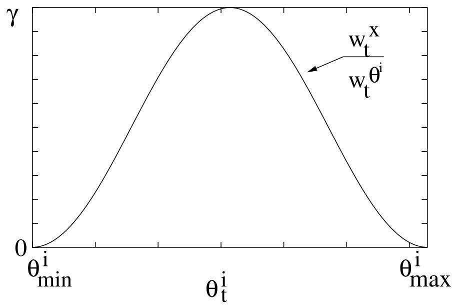


Fig. 4 The behavior of the function described in (28)

$\mathbf{W}_\theta$ and $\mathbf{W}_\mathbf{x}$ dependent on the robotic arm configuration. As the arm gets closer to one of the joint limits, say θ^i, the corresponding element $w^{\theta,i}$ of the matrix $\mathbf{W}_\theta$ gets bigger to eventually have a ratio $w^x/w^{\theta,i}$ equal to zero, which amounts to have a pure joint angle controller and thus avoid the joint limit. This can be achieved by making the weight ratio $w_t^x/w_t^{\theta,i}$ depend on the joint angle position θ_t^i as follows.

$$\frac{w_t^x}{w_t^{\theta,i}} = \frac{1}{2}\gamma\left(-\cos\left(2\pi\cdot\frac{\theta_t^i-\theta_{min}^i}{\theta_{max}^i-\theta_{min}^i}\right)+1\right), \tag{28}$$

where θ^i_{min} and θ^i_{max} are the joint angle boundaries and γ is a constant setting the maximum value for $w^x/w^{\theta,i}$. The right-hand side of this equation is plotted in Fig. 4. By applying this equation, the control is purely angular ($w^x=0$) when the system is approaching the joint boundary, thus avoiding it.

For the implementation of this modulation, it is again simpler to use (14) instead of (13). In fact, working with the inverse of the weight matrices, one can set the $\mathbf{W}_\mathbf{x}^{-1}$ to identity and the diagonal elements of $\mathbf{W}_\theta^{-1}$ can be computed according to the right-hand side of (28).

This method is quite different from other joint limit avoidance methods such as in Liégeois (1977), Fung Chan and Dubey (1995) or Chaumette and Marchand (2001). Those authors typically consider a main task, given by a end-effector trajectory and use the redundant degrees of freedom to optimize a secondary task, in this case joint limit avoidance. This secondary task is only performed if it does not disrupt the main task. Because we use the dynamical system approach, we do not have such a constraint. In fact, our joint angle avoidance method does influence the end-effector trajectory, but this is of no concern to us, as the target will nevertheless be reached, thanks to the attractor attached to the target.

3.7 Robustness to singularities

We have mentioned before that our controller can be seen as a generalization of the Damped Least-Squares (DLS) inverse method (Chiaverini et al. 1994). The DLS inverse $\mathbf{J}^*$ of the Jacobian matrix $\mathbf{J}$ is defined by

$$\mathbf{J}^* = (\epsilon\mathbf{I}_n + \mathbf{J}^T\mathbf{J})^{-1}\mathbf{J} = \mathbf{J}^T(\epsilon\mathbf{I}_m + \mathbf{J}\mathbf{J}^T)^{-1}, \tag{29}$$

where $\epsilon > 0$, $\mathbf{I}_n$ and $\mathbf{I}_m$ are the identity matrices of dimension n and m respectively. This inverse was introduced by Wampler (1986) and Nakamura and Hanafusa (1986) as an alternative to the Moore-Penrose pseudo-inverse solution (Whitney 1969). The advantage of this method is that it effectively avoids singularities because $\mathbf{J}^*$ is always defined. This comes at the cost of precision in the tracking of a desired end-effector velocity.

Equations (13) and (14) make essentially use of this DLS inverse (up to a diagonal matrix multiplication) and therefore our controller also avoids singularities. Because the matrices

$$\mathbf{J}_n^\# = \mathbf{W}_\theta + \mathbf{J}^T\mathbf{W}_\mathbf{x}\mathbf{J}, \tag{30}$$

$$\mathbf{J}_m^\# = \mathbf{W}_\mathbf{x}^{-1} + \mathbf{J}\mathbf{W}_\theta^{-1}\mathbf{J}^T \tag{31}$$

are positive definite (as long as $\mathbf{W}_\theta$ and $\mathbf{W}_\mathbf{x}^{-1}$ are positive definite), they are never singular. Moreover, an upper bound on the condition number of $\mathbf{J}_m^\#$ is given by:

$$\operatorname{cond}(\mathbf{J}_m^\#) \le 1 + \gamma\cdot\max_\theta(\sigma_1^2(\theta)), \tag{32}$$

where $\sigma_1(\theta)$ is the biggest singular value of $\mathbf{J}(\theta)$. This formula can give a maximal value for the variable γ for a given kinematic function $\mathbf{K}$, over which computational problems may arise. This is an additional advantage of using (14) and the corresponding weight parametrization.

3.8 Convergence

As mentioned in Sect. 2.2.1, each of the sub-controllers has one single fixed point. This fixed point is the target location and acts as an attractor. This, however, does not ensure that the controller as a whole, i.e. the combination of the two sub-controllers, also has a unique fixed point. In fact, the nonlinear interaction between the two sub-controllers may give rise to the appearance of spurious fixed points. Those appear when the two controllers exactly cancel each other. As can be seen in (13), this is the case when

$$\mathbf{W}_\theta\dot{\theta}_t^d + \mathbf{J}_t^T\mathbf{W}_\mathbf{x}\dot{\mathbf{x}}_t^d = \mathbf{0}, \tag{33}$$

$$\dot{\theta}_t = \mathbf{0}, \tag{34}$$

$$\dot{\mathbf{x}}_t = \mathbf{0} \tag{35}$$

which means that

$$\theta_T - \theta_t = -\mathbf{W}_\theta^{-1}\mathbf{J}_t\mathbf{W}_\mathbf{x}\big(\mathbf{K}(\theta_T) - \mathbf{K}(\theta_t)\big). \tag{36}$$

The existence of configurations θ_t and θ_T satisfying this equation depends on the function $\mathbf{K}$ and on the weights $\mathbf{W}_\mathbf{x}$

and $\mathbf{W}_\theta$. For $\mathbf{W_x} = \mathbf{0}$, there are no such points apart those for which $\theta = \theta_T$, i.e., the only fixed point is the target. But for other values of the weights, those spurious fixed points may well appear. Hence the parameter γ of (28) is a bifurcation parameter of the system. Section 5.1 presents simulation results confirming this hypothesis.

The presence or absence of cycles in the system is harder to prove, especially considering the weight dynamics. Simulation results seem to indicate that there are no such cycles.

Both spurious attractors and cycles can be avoided by making γ decay slowly. This way, the system will necessarily cross the bifurcation threshold at some point and reach the target.

3.9 Stability

The VITE-like dynamical system is a linear system with a single attractor. As such it is asymptotically stable. Because the system has no singularity, the velocities remain bounded. The joint angles also remain bounded by the joint limit avoidance mechanism. The whole system thus remains bounded, as long as the input, i.e., the target location is bounded. Since the system needs an input, it must be provided with a default input in case the target is not tracked. It can be for example the last target position, the current manipulator configuration or a "rest" position.

4 Implementation

This controller was implemented for the control of the arms of the Hoap2 humanoid robot of Fujitsu (see Fig. 5). Those arms have four DOFs assembled in a kinematic chain depicted in Fig. 6. We describe the end-effector location by its three spatial coordinates and discard its orientation. We thus have $n = 4$ and $m = 3$, which means that we are dealing with a redundant manipulator.

The experimental setting comprises a stereo-vision system composed of two cameras, the robot and two personal computers running Linux. The stereo-vision system and the controller run on the first computer, while the second computer is used as an interface with the robot. Using color recognition, the stereo-vision system gives the controller the target location. The controller updates the simulated joint angle trajectories and gives it to the second computer at a fixed rate of 20 Hz. The second computer, which runs Real-Time Linux, performs a linear interpolation and sends the robot the positions at a rate of 1 kHz. The robot has an on-board high-gain feedback controller, i.e. we control the robot in position and the on-board controller computes the corresponding torques.

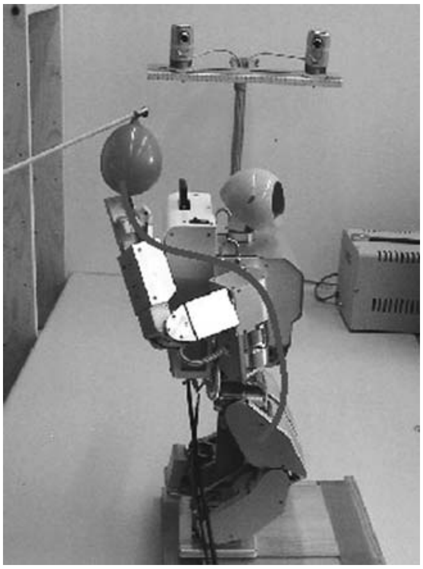


Fig. 5 The robot reaching for a target tracked by a stereo-vision system

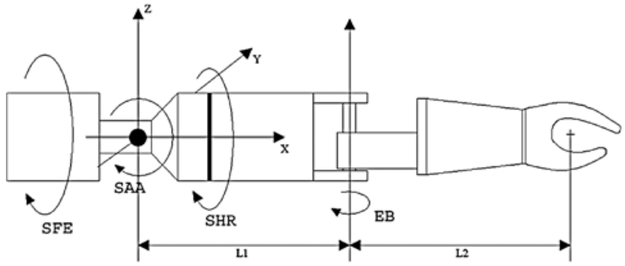


Fig. 6 A schematic representation of the robotic arm. It has four DOFs: SFE, SAA, SHR, EB

5 Results

5.1 Convergence properties

In order to see when and where spurious attractors appear, we performed a simulative analysis of the system. We simulated two millions trajectories given by randomly and uniformly sampled starting configurations and target locations. Each trajectory was simulated for various values of γ. The ratio of trajectories that did not reach the target is plotted in Fig. 7. One can see that for small γ, there are no spurious attractors and all trajectories reach their target. But when γ gets bigger, spurious attractors appear and some trajectories cannot reach their goal. Figure 8 shows the location of the targets that could not be reached. They almost all lie in the vicinity of lower boundary of the workspace. The region of unreachable targets is centered around a fully downward stretched vertical arm position. As illustrated in Fig. 8, this region grows with γ. It must be noted that the points shown by this figure are not always unreachable, they are unreachable by a non-empty set of starting configurations. Figure 9

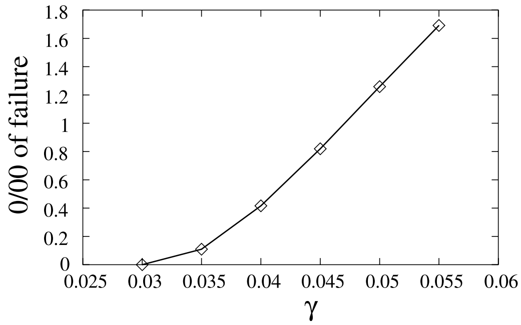


Fig. 7 The ratio of reaching failure for increasing values of γ. For a very small γ all reaching tasks could be successfully performed. But when $\gamma = 0.035$, spurious attractors appear

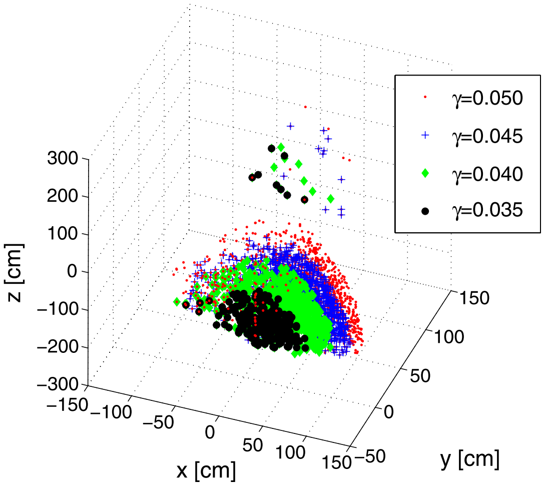


Fig. 8 The set of unreachable targets, in Cartesian space for different values of γ. The origin lies on the shoulder

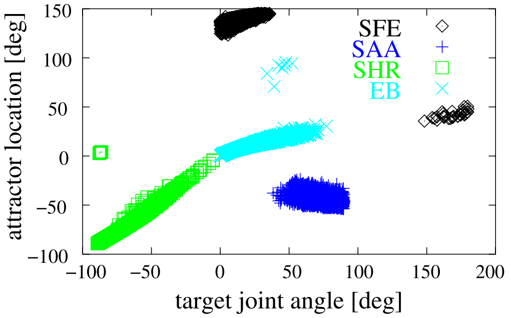


Fig. 9 The attractor location depending on the joint angle target. This graph represents an eight-dimensional space, because there are two dimensions for each angle (the target and the attractor value). SFE, SAA, SHR and EB correspond to the four DOFs of the arm. Here, $\gamma = 0.05$

shows the location of the unreachable target and the spurious attractor where the trajectory is trapped in joint angle space. One sees that for particular locations of the target joint angle (x-axis), the trajectory might be brought to a spurious attractor (y-axis).

5.2 Point-to-point reaching trajectories

Our controller can perform accurate point-to-point reaching motions. An example is shown in Fig. 10 which shows a reaching movement in three dimensions, performed in simulation and with the real robot. One sees that the robot matches well the simulated trajectory. Thanks to this close match between the simulated and the real trajectory, one can assume that the results obtained through simulations carry over to the real robot trajectories. Consequently, unless specified otherwise, the experiments presented in the rest of this paper are done in simulations.

Figure 11 shows additional properties of the movement displayed in Fig. 10. On the right, one sees the bell-shaped end-effector velocity profile and the joint angle velocity profiles. On the left, one sees how the system handles the target arm configuration redundancy. In joint space, the system aims at the closest joint angle configuration that corresponds to the target in Cartesian space.

Quasi-horizontal movements in front of the robot are depicted in Fig. 12. The controller produces quasi-straight trajectories, when the reaching is performed in the workspace center. This is clearly because the controller produces trajectories which are some sort of compromise between straight lines in end-effector location space (produced by the end-effector location sub-controller) and straight lines in joint angle space (produced by the joint angle sub-controller). Moreover, the velocity profiles are bell-shaped and smooth, due to the use of the VITE-like dynamical system. Bell-shaped velocity profiles and quasi-straight hand paths are typical of human reaching movements (Morasso 1981). Moreover, it has been suggested that human movements are the result of “a compromise between a straight line in workspace and a straight line in joint space” (Cruse and Brüwer 1987).

5.3 Singularity avoidance

As explained in Sect. 3.7, the controller presented here avoids singularities. By this, we do not merely mean that the controller avoids singular configurations (as in Baillieul 1985), but that there are no singularities for our controller, whatever the configuration. This fact is illustrated here on a task involving reaching to and from points liable to produce a singularity in traditional controllers. This happens when the Jacobian matrix of the kinematic function is degenerate, i.e when its rank is smaller than m. The starting position is set as the arm fully stretched downward and the target position is with the arm fully and horizontally stretched to the side. For those two positions, the Moore-Penrose pseudo-inverse of the Jacobian matrix cannot be

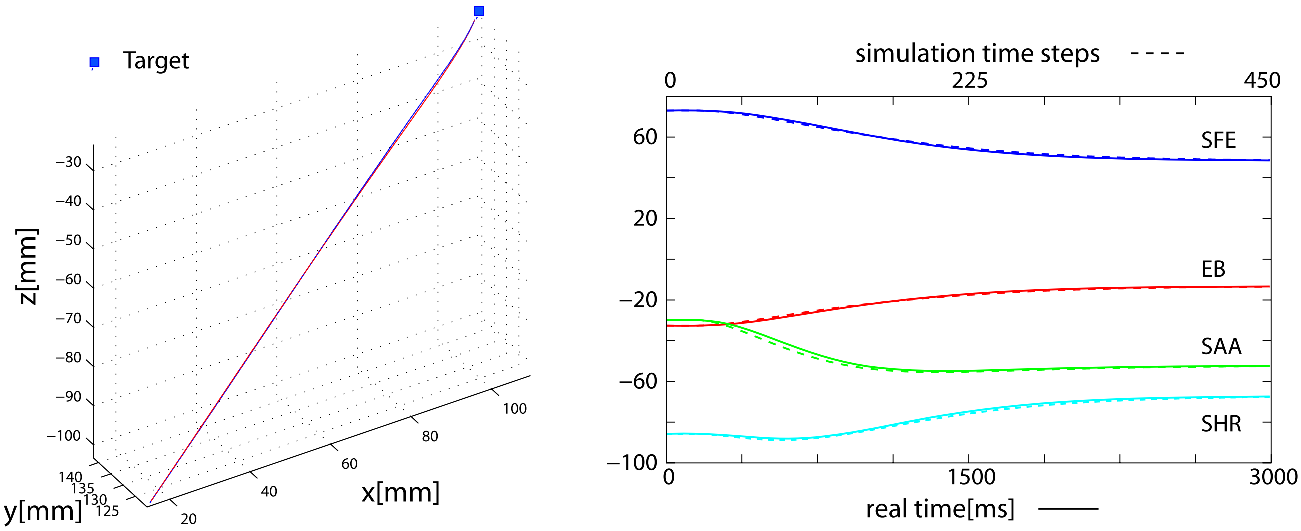
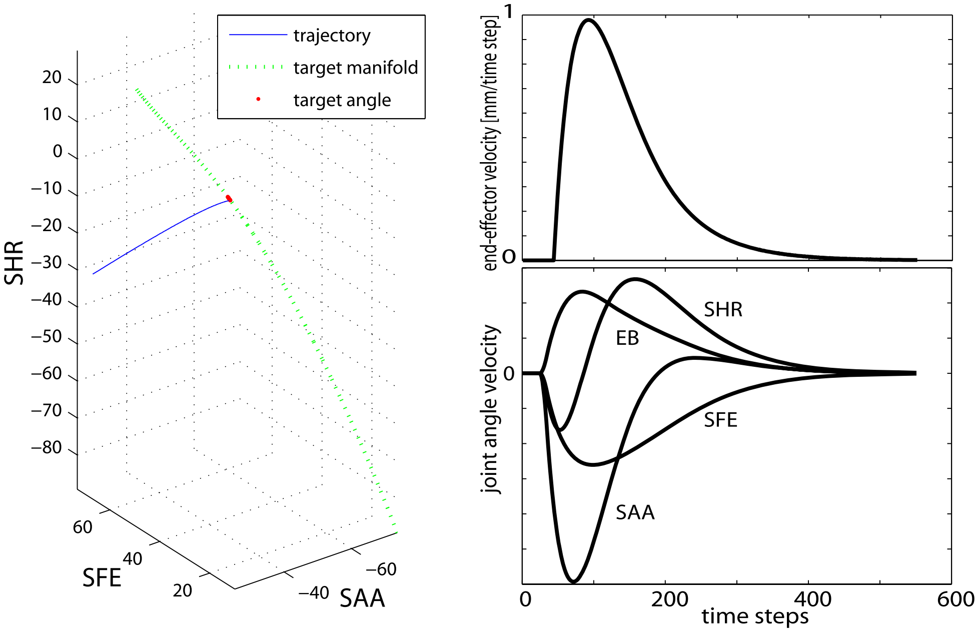


Fig. 10 Comparison between simulated movement and real movements. *Left*: a 3D reaching trajectory, in simulation (*dashed line*) and on the robot (*solid line*). *Right*: the corresponding joint angle trajectories in simulation (*dashed*) and as measured on the robot sensors (*solid*). Angles are given in degrees. The delay was removed to ease the comparison. One simulation time step corresponds to roughly 7 milliseconds in real time

Fig. 11 *Left*: the redundancy resolution for the movement depicted in Fig. 10. The target manifold Θ_T is represented by the *dotted line* and one can see that the joint angle trajectory (*solid line*) aims at the closest point of Θ_T. This point (shown with the *dot*) moves on Θ_T according to (18). *Right*: the end-effector velocity (*above*) and the joint angle velocities (*below*) for the same movement



computed and hence the task cannot be accomplished using that method. This limitation, however, does not apply to the hybrid controller presented here, as can be seen in Fig. 13. This figure shows the hand path trajectory obtained when performing this task and the condition number of the matrix which is inverted at every time step. One sees that this condition number remains in the first order of magnitude, which means that the inversion can be reliably performed without numerical instability.

5.4 Joint limit avoidance

The joint limit avoidance method described above prevents that the robotic arm reaches its joint angle boundaries. An example is given in Fig. 14, which shows the end-effector and joint angle trajectories, with and without the joint limit avoidance mechanism. In this example the robot is asked to reach from behind his neck to the front. One sees that the weight modulation forces the system to remain within the

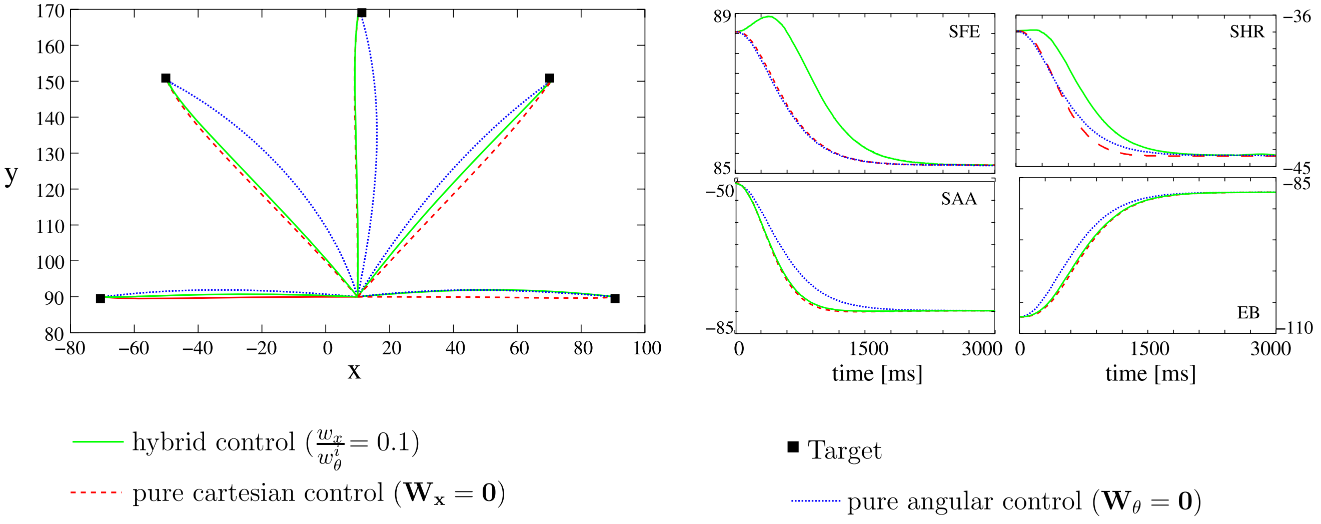


Fig. 12 *Left*: reaching trajectories for various weight configurations. The trajectories lie in the center of the workspace. *Right*: the joint angle trajectories (measured on the robot sensors) for the rightmost movement

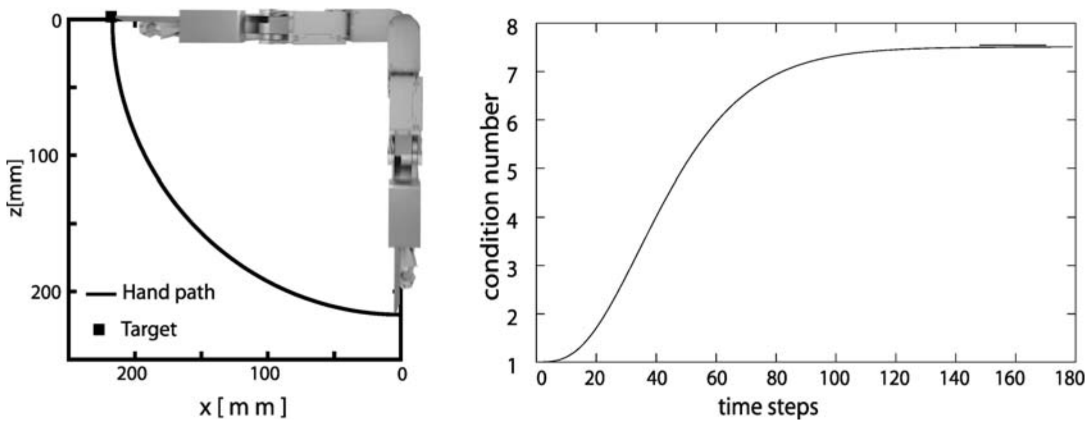


Fig. 13 *Left*: the trajectory for reaching to a potentially singular point. There is no singularity. *Right*: the evolution of matrix condition number corresponding to this trajectory. The inverted matrices remain well-conditioned throughout the movement

Table 1 Joint limit avoidance results

Joint	SFE	SAA	SHR	EB
Lower boundary	−90	−180	−90	−115
Without avoidance	−90	−180	−90	−115
With avoidance	−83.5	−164.1	−89.9	−107.2

joint boundaries. Without the joint limit avoidance mechanism, the boundaries are reached for all four joint angles. Table 1 shows how close the trajectories get from the workspace boundaries.

For the kind of tasks displayed in Fig. 14, the joint limit avoidance mechanism has a significant influence on the end-effector trajectories. This may be a disadvantage in some cases, but if there is no particular constraint on the end-effector it seems adequate to take advantage of this freedom.

5.5 Robustness to perturbations

The controller described here can essentially be understood as a dynamical system having the target as an attractor. It therefore comes as no surprise that the system is robust to perturbations. Here, two kinds of perturbations are considered: a sudden target displacement and a transient external force applied on the manipulator. The target displacement example is illustrated in Fig. 15. One sees that by moving the target, one simply moves the attractor of the dynamical system. Hence, the system reaches the target while retaining a continuous velocity, because it is a second-order system.

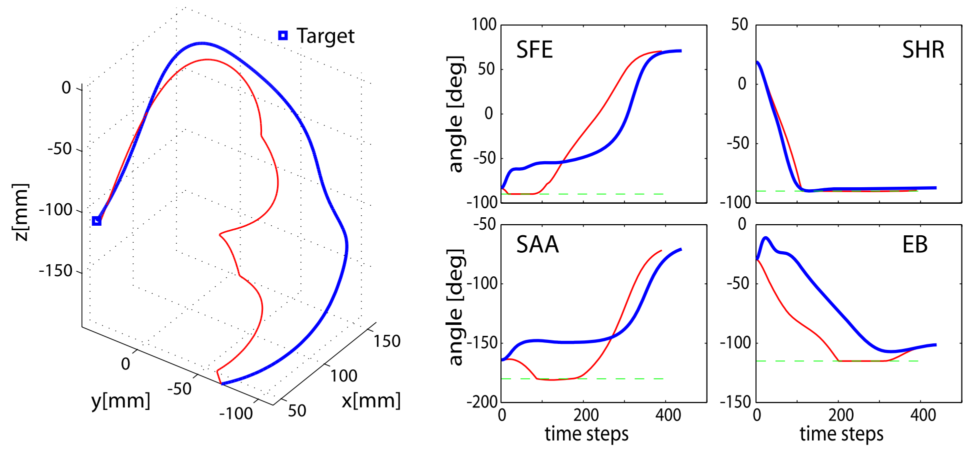


Fig. 14 A reaching trajectory, with and without the joint limit avoidance mechanism. *Left*: end-effector trajectories. *Right*: Joint angle trajectories. Without the joint limit avoidance mechanism (*thin line*) the system bumps into the joint limits (shown with *dashed lines*). This does not occur with the joint limit avoidance mechanism (*thick lines*). The weight ratio (and γ respectively) is 0.05. SFE, SAA, SHR and EB refer to the four DOFs, as displayed in Fig. 6

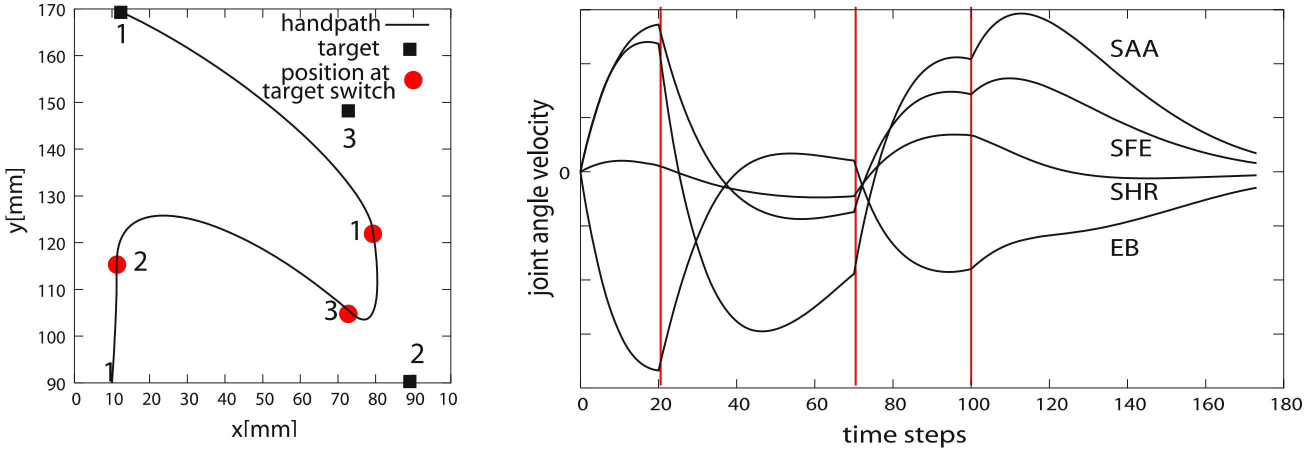


Fig. 15 The end-effector trajectory altered by sudden target displacements. *Left*: end-effector trajectory. The system is at the location indicated by the *circles* when the target suddenly moves to *square* with the corresponding number. *Right*: the corresponding joint angle velocities. The *vertical bars* indicate the occurrences of a target displacement

The effect of an external perturbation acting on the manipulator is depicted in Fig. 16. Again, due the attractor nature of the system, this perturbation is handled on-line in an appropriate manner. The system smoothly adapts its trajectory to reach the target.

6 Discussion

In the preceding sections, we described a robotic manipulator controller based on two basic principles drawn from the studies of human reaching movements, multi-referential movement representation and dynamical system control. The controller comprises two VITE dynamical systems, one acting on the end-effector described in a body-centered Cartesian frame of reference and the other acting on the joint angle arm configuration. The controller is interesting both for robotics and for biological control modeling.

6.1 Strengths of the controller

From a pure roboticist perspective, the multi-referential controller presented here has several advantages over classical controllers. First, it does not have any singularity because it uses a generalized version of the DLS inverse which has been shown to avoid the singularity problem (Wampler 1986). The inverse in (13) can always be computed as long as $\mathbf{W}^{\theta}$ is positive definite because $\mathbf{J}_t^T \mathbf{W}_{\mathbf{x}} \mathbf{J}_t$ is positive semi-definite. Another advantage of the controller is that it allows a simple and elegant solution to the joint limit avoidance problem. Our results show that this method is effective and yields smooth and short end-effector trajectories. Note that this method makes the assumption that the joint angle workspace is convex, which is generally the case in the absence of obstacles. If there are obstacles, or in order to prevent self collisions, a classical potential field method (Khatib 1985) can easily be (and has been) integrated into our dynamical system framework.

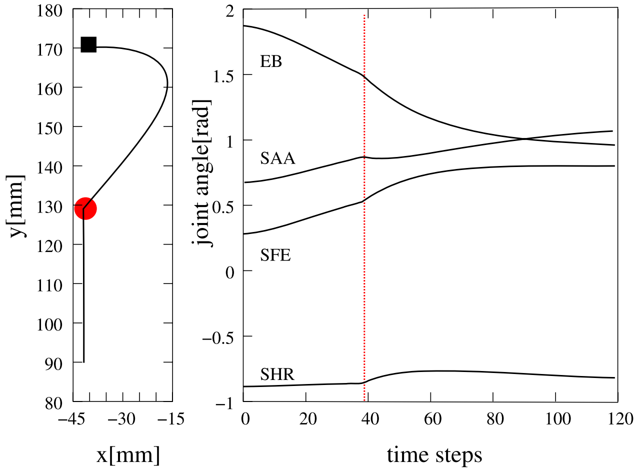


Fig. 16 Robustness to external perturbations. The end-effector trajectory is depicted on the *left* and the joint angle trajectories appear on the *right*. The location and time of the perturbation are indicated by a *circle* and a *vertical bar* respectively

Finally, the controller is robust to unexpected changes in the target location and smoothly adapts its trajectory accordingly. This is due to the robustness of the dynamical system underlying the controller.

6.2 Biological inspiration

This robotic controller draws its inspiration from two principles purportedly underlying biological arm control. Those principles are on one hand a dynamical system control and on the other hand a multi-referential control. They have been argued to lie at the core of human movement control (Kelso 1995; Paillard 1991). The particular choice of dynamical system (the VITE-like system) and of frames-of-reference (joint angles and end-effector locations) do have some biological plausibility and have also been suggested to underlie human reaching control (Bullock and Grossberg 1988; Cruse and Brüwer 1987; Carrozzo and Lacquaniti 1994). The resulting movements share some of their properties with human reaching movements, either due to the VITE model such as the speed-accuracy trade-off and the speed-to-distance proportionality, or due to the multi-referential control, such as the quasi-straight paths. The way the target redundancy is handled (see Sect. 3.4) results in a dependence between final and initial postures, which can also be found in human reaching motions (Soechting et al. 1995; Desmurget et al. 1998a).

The controller presented here is kinematic only and does not take the dynamical properties of the robot limbs into account. This is why in the present implementation the specified position is fed into a PID-like controller. The exact nature of the relationship between the controller and the actual limbs in biological systems is still unclear, but the use of forward models to account for the limb dynamics has been hypothesized (Wolpert et al. 1995; Ariff et al. 2002). This could be a further extension of our system.

7 Conclusion

The idea of concurrent dynamical systems (or controllers) interacting with each other and leading to the emergence of a global behavior, is interesting for robot control as it provides robustness to failures of the individual controller and allows a smooth switching from one controller to the other, thus allowing multiple behaviors. It is also an appealing paradigm under which to study and model biological movement control, as it well suits the distributed computing performed in biological systems.

The controller presented here hints at the possibilities offered by multi-referential dynamical system control, but it also highlights the challenges related to the design of such controllers. Indeed it remains to be further investigated under what conditions dynamical systems can be combined so that a global coherent behavior emerges and how spurious attractors or limit cycles can be avoided.

Acknowledgements This work was funded by the Swiss National Science Foundation and EU Project IST-2004-004370 RobotCub.

Appendix: Proof of (14)

Starting from (9), we have

$$\dot{\theta}_t = \dot{\theta}_t^d + \mathbf{W}_\theta^{-1}\mathbf{J}_t^T\lambda, \tag{37}$$

$$\dot{\mathbf{x}}_t = \dot{\mathbf{x}}_t^d - \mathbf{W}_\mathbf{x}^{-1}\lambda. \tag{38}$$

Inserting those equations into (7) yields

$$\mathbf{J}_t(\dot{\theta}_t^d + \mathbf{W}_\theta^{-1}\mathbf{J}_t^T\lambda) = \dot{\mathbf{x}}_t^d - \mathbf{W}_\mathbf{x}^{-1}\lambda, \tag{39}$$

$$\lambda = \left(\mathbf{W}_\mathbf{x}^{-1} + \mathbf{J}_t\mathbf{W}_\theta^{-1}\mathbf{J}_t^T\right)^{-1}(\dot{\mathbf{x}}_t^d - \mathbf{J}_t\dot{\theta}_t^d). \tag{40}$$

This equation is then used to replace λ in (37).

$$\dot{\theta}_t = \dot{\theta}_t^d + \mathbf{W}_\theta^{-1}\mathbf{J}_t^T\left(\mathbf{W}_\mathbf{x}^{-1} + \mathbf{J}_t\mathbf{W}_\theta^{-1}\mathbf{J}_t^T\right)^{-1}(\dot{\mathbf{x}}_t^d - \mathbf{J}_t\dot{\theta}_t^d). \tag{41}$$

References

Abend, W., Bizzi, E., & Morasso, P. (1982). Human arm trajectory formation. *Brain*, *105*, 331–348.

Ariff, G., Donchin, O., Nanayakkaran, T., & Shadmehr, R. (2002). A real-time state predictor in motor control: Study of saccadic eye movements during unseen reaching movements. *The Journal of Neuroscience*, *22*, 7721–7729.

Atkeson, C. G., & Hollerbach, J. M. (1985). Kinematic features of unrestrained vertical arm movements. *The Journal of Neuroscience*, *5*(9), 2318–2330.

Baillieul, J. (1985). Kinematic programming alternatives for redundant manipulators. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 722–728).

Billard, A., Calinon, S., & Guenter, F. (2006). Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, *54*(5), 370–384.

Bizzi, E., Accornero, N., Chapple, W., & Hogan, N. (1984). Posture control and trajectory formation during arm movement. *The Journal of Neuroscience*, *4*, 2738–2744.

Bullock, D., & Grossberg, S. (1988). Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, *95*(1), 49–90.

Burdick, J. W. (1989). On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds. In: *Proceedings of the IEEE international conference on robotics and automation* (pp. 264–270).

Carrozzo, M., & Lacquaniti, F. (1994). A hybrid frame of reference for visuo-manual coordination. *Neuroreport*, *5*, 453–456.

Chaumette, F., & Marchand, É. (2001). A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing. *IEEE Transactions on Robotics and Automation*, *17*(5), 719–730.

Chiaverini, S., Siciliano, B., & Egeland, O. (1994). Review of the damped leat-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology*, *2*(2), 123–134.

Cruse, H., & Brüwer, M. (1987). The human arm as a redundant manipulator: The control of path and joint angles. *Biological Cybernetics*, *57*, 137–144.

Desmurget, M., Jordan, M., Prablanc, C., & Jeannerod, M. (1997). Constrained and unconstrained movements involve different control strategies. *Journal of Neurophysiology*, *77*, 1644–1650.

Desmurget, M., Gréa, H., & Prablanc, C. (1998a). Final posture of the upper limb depends on the initial position of the hand during prehension movements. *Experimental Brain Research*, *119*, 411–516.

Desmurget, M., Pélisson, D., Rosseti, Y., & Prablanc, C. (1998b). From eye to hand: Planning goal-directed movements. *Neuroscience and Biobehavioural Reviews*, *22*(6), 761–788.

Feldman, A. G., & Levin, M. F. (1995). The origin and use of positional frames of reference in motor control. *Behavioral Brain Sciences*, *18*, 723–806.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, *5*(7), 1688–1703.

Fung Chan, T., & Dubey, R. V. (1995). A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulator. *IEEE Transactions on Robotics and Automation*, *11*(2), 286–292.

Giszter, S. F., Mussa-Ivaldi, F. A., & Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord. *The Journal of Neuroscience*, *13*(2), 467–491.

Ijspeert, A. J., Nakanishi, J., & Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 1398–1403).

Iossifidis, I., & Schöner, G. (2004). Autonomous reaching and obstacle avoidance with the anthropomorphic arm of a robotic assistant using the attractor dynamics approach. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 4295–4300).

Kelso, J. A. S. (1995). *Dynamic patterns: The self-organization of brain and behavior*. Cambridge: MIT Press.

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 500–505).

Lacquaniti, F., Soechting, J. F., & Terzuolo, S. A. (1986). Path constraints on point-to-point arm movements in three-dimensional space. *Neuroscience*, *17*(2), 313–324.

Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, *7*(12), 868–871.

Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research*, *42*, 223–227.

Nakamura, Y., & Hanafusa, H. (1986). Inverse kinematics solutions with singularity robustness for robot manipulator control. *ASME Journal of Dynamic Systems, Measurement, and Control*, *108*, 163–171.

Paillard, J. (Ed.). (1991). *Brain and space*. London: Oxford University Press. Chaps. from Arbib, Berthoz and Paillard.

Righetti, L., & Ijspeert, A. J. (2006). Programmable central pattern generators: an application to biped locomotion control. In *Proceedings of the 2006 IEEE international conference on robotics and automation* (pp. 1585–1590).

Schöner, G., Dose, M., & Engels, C. (1995). Dynamics of behavior: theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, *16*, 213–245.

Shadmehr, R., & Wise, S. P. (2005). *The computational neurobiology of reaching and pointing*. Cambridge: MIT Press.

Soechting, J. F., Buneo, C.A., & Flanders, M. (1995). Moving effortlessly in three dimensions: Does Donder's law apply to arm movement? *Journal of Neuroscience*, *15*(9), 6271–6280.

Todorov, E., & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, *5*(11), 1226–1235.

Wampler, C. W. (1986). Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, *16*(1), 93–101.

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prosteses. *IEEE Transactions on Man-Machine Systems*, *10*(2), 47–52.

Wolpert, D., Ghahramani, Z., & Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, *269*(5232), 1880–1882.

Micha Hersch graduated in Computer Science at the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland in 2003. He also obtained a DEA (M.Sc.) in Cognitive Science at the Université Pierre et Marie Curie, Paris in 2004. He is currently working towards his Ph.D. under the supervision of Professor Billard at Learning Algorithms and Systems Laboratory (LASA) at EPFL, within the EU RobotCub project. His research interests include bio-inspired and developmental robotics, and embodied cognition at large.

Aude G. Billard is an Associate Professor and head of the Learning Algorithms and Systems Laboratory at the school of Engineering at EPFL, which she joined in 2002. She was a Research Assistant Professor at the Computer Science department at the University of Southern California (USC) from 00 to 02, where she remains an adjunct faculty member. Dr. Billard serves as Administrative Committee member for the IEEE Robotics and Automation Society. She received her B.Sc. (1994) and M.Sc. (1995) in Physics from the EPFL, with a specialization in Particle Physics at the European Center for Nuclear Research (CERN). She received her MSc in Knowlegde Based System (1996) and her Ph.D. in Artificial Intelligence (1998) from the department of Artificial Intelligence at the University of Edinburgh.