SAPHARI

SAFE AND AUTONOMOUS PHYSICAL HUMAN-AWARE ROBOT INTERACTION

SEVENTH FRAMEWORK PROGRAMME

COGNITIVE SYSTEMS & ROBOTICS

Project funded by the European Community's 7th Framework Programme (FP7-ICT-2011-7)
Grant Agreement ICT–287513

Deliverable D6.2.1

Report on Multimodal reactive motion generation (trajectories of motion, force, and impedance patterns)

Deliverable due date: 30 April 2015	**Actual submission date:** 15 June 2015
Start date of project: 1 November 2011	**Duration:** 48 months
Lead beneficiary: TUM	**Revision:** DRAFT

Nature: R	**Dissemination level:** CO

R = Report
P = Prototype
D = Demonstrator
O = Other
	PU = Public
	PP = Restricted to other programme participants (including the Commission Services)
	RE = Restricted to a group specified by the consortium (including the Commission Services)
	CO = Confidential, only for members of the consortium (including the Commission Services)

Executive Summary

This deliverable of WP6 deals with the problem of multimodal reactive motion generation, considering not only trajectories of motion but also force and impedance patterns. This is a key aspect in all the tasks that require the robot to physically interact with the environment.

TUM improved the dynamical system (DS) modulation technique for reactive collision avoidance to guarantee the Lyapunov stability of the modulated DS. The original formulation, in fact, do not consider a full stability analysis of the modulated dynamical system. Despite it is possible to show that the equilibrium points of the modulated DS are not affected by the modulation matrix, the presence of periodic orbits cannot be excluded. The stability, as well as the avoidance of possible obstacles, are guaranteed in our novel formulation by a customized control input computed solving a constrained optimization problem. Indeed, stability and collision avoidance are considered as constraints of the optimization problem for which an analytical solution exists. Hence, the algorithm can be implemented and applied in real-time. Preliminary simulation results show the effectiveness of the proposed approach.

Regarding force and impedance patterns generation, TUM proposed a novel approach, based on reinforcement learning (RL), to learn impedance behaviors by robot self practise. The proposed approach combines the benefits of two state-of-the-art approaches, giving the possibility to learn full stiffness matrices. Hence, the novel approach takes into account the interdependency among different degrees-of-freedom (or different directions in the Cartesian space). Moreover, the external applied torque is explicitly considered in the reward function and minimized during the learning procedure. The effectiveness of the proposed approach is demonstrated with an experiment on a 7 degrees-of-freedom KUKA LWR IV+ manipulator.

Table of contents

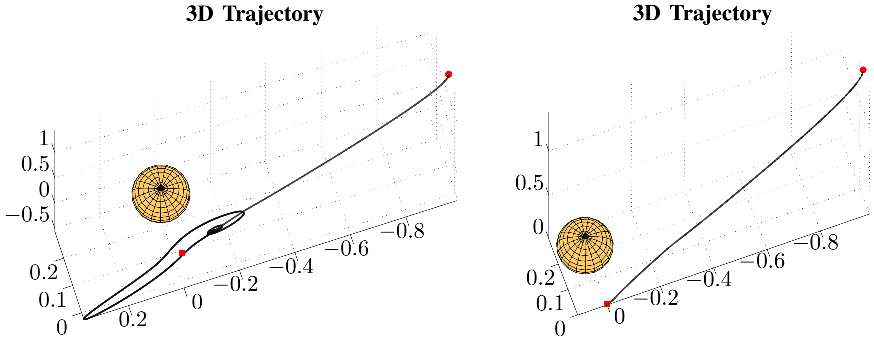
1 Optimal modulation of dynamical systems

In the first period of the project TUM developed a reactive collision avoidance algorithm, namely the distance-based Dynamical System (DS) modulation, capable to locally modify the dynamics of a first order system to avoid possible collisions with fixed [1] and moving obstacles [2]. The algorithm extends the preliminary work in [3] to the case of obstacles represented as point clouds. A suitable *Modulation Matrix* $\boldsymbol{M}(\boldsymbol{p})$ is used to modulate the DS, i.e. $\dot{\boldsymbol{p}} = \boldsymbol{M}(\boldsymbol{p})\boldsymbol{f}(\boldsymbol{p})$, where $\boldsymbol{p}$ represents the current robot position. The modulation does not affect the equilibrium points of the modulated DS and it guarantees the obstacle surface will not be penetrated during the motion.

The modulation matrix depends on the state (position) of the modulated DS. The state dependency of the modulation matrix makes the modulated DS non-linear also if $\boldsymbol{f}$ is linear. In general, non-linear systems are not always guaranteed to reach their equilibria, also when those equilibrium points are stable. As an example, consider the linear time invariant system:

$$\dot{\boldsymbol{p}} = \boldsymbol{A}\boldsymbol{p} = \begin{bmatrix} -8.2 & 1 & 1 \\ -45 & -3 & 0 \\ 0 & 8.3 & 5.3 \end{bmatrix} \boldsymbol{p} \tag{1}$$

that has a globally asymptotically stable (GAS) equilibrium at the origin (the eigenvalues of $\boldsymbol{A}$ are all negative). Putting a spherical object of radius $r = 0.1$, centered at $\boldsymbol{c} = [0.2\ 0.01\ 0.1]$, the modulated system follows a periodic orbit without converging to the equilibrium (see Figure 1(a)).



(a) DS Modulation (b) Optimal DS Modulation

Figure 1: (a) The modulated system follows a periodic orbit without converging to the equilibrium point. (b) The optimization of the modulated velocity guarantees the convergence to the equilibrium point.

1.1 Modulated Velocity Optimization

TUM proposes an approach to avoid the robot follows unexpected periodic trajectories, by computing a suitable velocity solving a constrained optimization problem. Let us assume that the autonomous system $\dot{\boldsymbol{p}} = \boldsymbol{f}(\boldsymbol{p})$ has a globally asymptotically stable (GAS) equilibrium $\hat{\boldsymbol{p}}$, and let's call $V_f(\boldsymbol{p})$ a Lyapunov function for that system. A velocity $\boldsymbol{f}'$ that guarantees the convergence to the goal and the impenetrability can be calculated solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{f}'} \quad & \frac{1}{2}\|\boldsymbol{f}' - \boldsymbol{M}\boldsymbol{f}\|^2 \\ \text{s. t.} \quad & \dot{V}_f = \nabla V_f \boldsymbol{f}' \leq -\gamma,\ \gamma > 0,\ \gamma(\boldsymbol{0}) = 0 \\ & \dot{V}_f = 0,\ \boldsymbol{p} = \hat{\boldsymbol{p}} \\ & \hat{\boldsymbol{n}}^T \boldsymbol{f}' = 0,\ \forall \bar{\boldsymbol{p}} \end{aligned} \tag{2}$$

where the first two constraints guarantee that $\hat{\boldsymbol{p}}$ is GAS. The third equality constraint is the impenetrability condition, i.e. the normal component of the velocity vanishes on the obstacle surface.

Assuming that $\boldsymbol{f}' = \boldsymbol{M}\boldsymbol{f} + \boldsymbol{u}$, the equality constraint in (2) can be satisfied putting $\hat{\boldsymbol{n}}^T\boldsymbol{u} = 0,\ \forall \boldsymbol{p}$. The function $\gamma(\boldsymbol{p})$ is chosen as[1] $\gamma(\boldsymbol{p}) = min(\gamma_1, a(1 - e^{-b||\boldsymbol{p}||}))$, $\gamma_1 > 0$, where a and b are tunable parameters. Hence, $\gamma(\boldsymbol{p})$ is positive definite and it vanishes only at the origin. The maximum value of $\gamma(\boldsymbol{p})$ is bounded to γ_1 to avoid that, far from the equilibrium ($||\boldsymbol{p}|| \gg 0$), a big $\boldsymbol{u}$ ($||\boldsymbol{u}|| \gg 0$) is required to satisfy $\dot{V}_f(\boldsymbol{p}) \leq -\gamma(\boldsymbol{p})$. Under those assumptions, the Karush-Kuhn-Tucker (KKT) conditions for the optimization problem in (2) can be written as:

$$\begin{cases} \boldsymbol{u} + \mu(\nabla V_f)^T + \delta\hat{\boldsymbol{n}} = 0 \\ \nabla V_f \boldsymbol{u} \leq -\gamma - \nabla V_f \boldsymbol{M}\boldsymbol{f} \\ \mu\left(\nabla V_f \boldsymbol{f}' + \gamma\right) = 0 \\ \nabla V_f \boldsymbol{f}' = 0,\ \boldsymbol{p} = \hat{\boldsymbol{p}} \\ \boldsymbol{n}^T\boldsymbol{u} = 0 \end{cases} \tag{3}$$

Solving the system of KKT conditions (3) it is possible to compute the optimal solution $\boldsymbol{u}^\star$ of (2). It is straightforward to verify that the optimal control input $\boldsymbol{u}^\star$ can be expressed as:

$$\boldsymbol{u}^\star = \begin{cases} -\boldsymbol{M}\boldsymbol{f} & \boldsymbol{p} = \hat{\boldsymbol{p}} \\ \boldsymbol{0} & \dot{V}_f \leq -\gamma \\ -\mu(\nabla V_f)^T - \delta\hat{\boldsymbol{n}} & \text{otherwise} \end{cases} \tag{4}$$

where

$$\begin{cases} \mu = \dfrac{\gamma + \nabla V_f \boldsymbol{M}\boldsymbol{f} - (\nabla V_f \hat{\boldsymbol{n}})(\hat{\boldsymbol{n}}^T\boldsymbol{M}\boldsymbol{f})}{||\nabla V_f||^2 - |\nabla V_f \hat{\boldsymbol{n}}|^2} \\ \delta = \hat{\boldsymbol{n}}^T\boldsymbol{M}\boldsymbol{f} - \mu\nabla V_f \hat{\boldsymbol{n}} \end{cases} \tag{5}$$

As shown in Figure 1(b), the computed optimal control input is able to drive the robot toward the goal position while avoiding possible collisions.

2 Reinforcement learning of impedance behaviors

TUM extended the standard[2] PI^2 to learn couplings between joints (full stiffness matrices) rather than gain schedules for each joint individually as in [6]. The benefits of taking into account such coupling information in off-diagonal elements of stiffness matrices have been highlighted in several research works [7–9] which imply that the association of full stiffness matrices enables to learn synergies, coordination and couplings in motor control. A further novel idea TUM proposed is to consider external force-torque sensing as learning stimuli in the reward function of the RL algorithm. This idea allows to compensate for unknown contact forces and instability in a human-like manner [10] by increasing contact forces trial after trial until a primary task goal can be accomplished, e.g. the minimization of positional deviations imposed by interaction disturbances, which is also defined in the reward function.

[1] When the DS has a GAS equilibrium in $\hat{\boldsymbol{p}} \neq \boldsymbol{0}$ we simply chose $\gamma(\boldsymbol{p}) = min(\gamma_1, a(1 - e^{-b||\boldsymbol{p}-\hat{\boldsymbol{p}}||}))$.

[2] Compared to other approaches, like *Policy learning by Weighting Exploration with the Returns* (*PoWER*) [5], PI^2 has no limitation on the choice of the reward function.

2.1 Correlated Dynamic Movement Primitives (DMPs)

In Correlated DMPs [11], training data are given in terms of end-effector position $\boldsymbol{x} \in \mathbb{R}^m$, velocity $\dot{\boldsymbol{x}}_t \in \mathbb{R}^m$ and acceleration $\ddot{\boldsymbol{x}}_t \in \mathbb{R}^m$, and the desired acceleration command $\ddot{\boldsymbol{x}}_{t,d} \in \mathbb{R}^m$ is generated as:

$$\ddot{\boldsymbol{x}}_{t,d} = \sum_{j=1}^{p} h_{t,j}\left[\boldsymbol{K}_j^{\mathcal{P}}(\boldsymbol{\mu}_j^{\mathcal{X}} - \boldsymbol{x}_t) - \kappa^{\mathcal{V}}\dot{\boldsymbol{x}}_t\right] \tag{6}$$

The dynamical system in (6) can be considered as a spring-damper system with attractor vectors $\boldsymbol{\mu}_j^{\mathcal{X}} \in \mathbb{R}^m$, full stiffness matrices (i.e. coordination matrices) $\boldsymbol{K}_j^{\mathcal{P}} \in \mathbb{R}^{m\times m}$ and damping gain $\kappa^{\mathcal{V}} \in \mathbb{R}$. The attractor vectors $\left\{\boldsymbol{\mu}_j^{\mathcal{X}}\right\}_{j=1}^{p}$ and full coordination matrices $\left\{\boldsymbol{K}_j^{\mathcal{P}}\right\}_{j=1}^{p}$ are learnable policy parameters. In order to reproduce a desired path $\boldsymbol{x}_{t_i,d}, \dot{\boldsymbol{x}}_{t_i,d}, \ddot{\boldsymbol{x}}_{t_i,d} \in \mathbb{R}^m$ in each time step $t_i,\ i = 0, 1, \ldots, N-1$, the trajectory can be computed by summarizing the weighted attractor points over all basis functions to

$$\boldsymbol{x}_{t_i,d} = \sum_{j=1}^{p} h_{t_i,j}\boldsymbol{\mu}_j^{\mathcal{X}}\ , \tag{7}$$

with temporal weighting basis functions

$$h_{t_i,j} = \frac{\psi_{t_i,j}}{\sum_{l=1}^{p}\psi_{t_i,l}}\ , \quad \psi_{t_i,j} = \mathcal{N}(t_i; \mu_j^{\mathcal{T}}, \Sigma_j^{\mathcal{T}}) \tag{8}$$

composed of equally in time distributed Gaussians $\mathcal{N}(\mu_j^{\mathcal{T}}, \Sigma_j^{\mathcal{T}})$ with centers $\mu_j^{\mathcal{T}}$ and variances $\Sigma_j^{\mathcal{T}}$. The Gaussians are activated by the canonical system

$$\frac{1}{\tau}\dot{\nu}_{t_i} = -\alpha_\nu \nu_{t_i} \longrightarrow t_i = -\frac{\ln(\nu_{t_i})}{\alpha_\nu \tau} \tag{9}$$

where the time constant α_ν and the temporal scaling factor τ determine the movement duration. ν_t is set to $\nu_t = 1$ to initiate the movement and then converges to zero. Accordingly, the temporal varying stiffness matrix $\boldsymbol{K}_{t_i}^{\mathcal{P}}$ can be estimated by

$$\boldsymbol{K}_{t_i}^{\mathcal{P}} = \sum_{j=1}^{p} h_{t_i,j}\boldsymbol{K}_j^{\mathcal{P}} \tag{10}$$

to generate the PD motor command

$$\ddot{\boldsymbol{x}}_{t_i,d} = \boldsymbol{K}_{t_i}^{\mathcal{P}}(\boldsymbol{x}_{t_i,d} - \boldsymbol{x}_{t_i}) - \kappa^{\mathcal{V}}\dot{\boldsymbol{x}}_{t_i}\ . \tag{11}$$

2.2 Coordination Policy Improvement with Path Integrals (C-PI^2)

Previous approaches to learn DMP parameters through PI^2 have considered each DoF independently, thus the motor control variable of each joint or task space dimension is estimated individually. In order to additionally learn the couplings between motor control variables, the PI^2 algorithm must be viewed from a different perspective.

A trajectory can be learned by parameterizing the attractors in equation (7) in the policy form

$$\boldsymbol{x}_{t_i,d} = \sum_{j=1}^{p} h_{t_i,j}(\boldsymbol{\mu}_j^{\mathcal{X}} + \boldsymbol{\epsilon}_{t_i,j}^{\mathcal{X}}) = \sum_{j=1}^{p} h_{t_i,j}(\boldsymbol{\theta}_j^{\mathcal{X}} + \boldsymbol{\epsilon}_{t_i,j}^{\mathcal{X}})\ , \tag{12}$$

with parameter vectors $\boldsymbol{\theta}_j^{\mathcal{X}} \in \mathbb{R}^m$ and exploration noise vectors $\boldsymbol{\epsilon}_{t_i,j}^{\mathcal{X}} \in \mathbb{R}^m$. To learn full stiffness matrices, the policy (10) must be parameterized by $m \times m$ parameter matrices for each of the p basis functions

$$\boldsymbol{K}_{t_i}^{\mathcal{P}} = \sum_{j=1}^{p} h_{t_i,j}(\boldsymbol{K}_j^{\mathcal{P}} + \boldsymbol{\epsilon}_{t_i,j}^{\mathcal{P}}) = \sum_{j=1}^{p} h_{t_i,j}(\boldsymbol{\theta}_j^{\mathcal{P}} + \boldsymbol{\epsilon}_{t_i,j}^{\mathcal{P}}) \tag{13}$$

with parameter matrices $\boldsymbol{\theta}_j^{\mathcal{P}} \in \mathbb{R}^{m\times m}$ and exploration noise matrices $\boldsymbol{\epsilon}_{t_i,j}^{\mathcal{P}} \in \mathbb{R}^{m\times m}$. These parameters $\boldsymbol{\theta}_j^{\mathcal{X}} = \boldsymbol{\mu}_j^{\mathcal{X}}$ and $\boldsymbol{\theta}_j^{\mathcal{P}} = \boldsymbol{\mu}_j^{\mathcal{P}}$ form in combination the policy output $\ddot{\boldsymbol{x}}_{t_i,d}$ in equation (11) which can be interpreted as motor command for the PI^2 algorithm. As a consequence the immediate cost can be expressed, as for the PI^2 algorithm, in the form [4]

$$r_{t_i} = q_{t_i} + \frac{1}{2}\ddot{\boldsymbol{x}}_{t_i,d}^T \boldsymbol{R}\, \ddot{\boldsymbol{x}}_{t_i,d} \tag{14}$$

with an arbitrary state-dependent cost function q_{t_i} and a quadratic control weight matrix $\boldsymbol{R}$.

The exploration vector for the trajectory in equation (12) is drawn from a zero-mean Gaussian distribution $\boldsymbol{\epsilon}_{t_i,j}^{\mathcal{X}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\epsilon,j}^{\mathcal{X}})$ with variance $\boldsymbol{\Sigma}_{\epsilon,j}^{\mathcal{X}}$ for each basis function. Similarly, the exploration matrix in equation (13) is drawn from $\boldsymbol{\epsilon}_{t_i,j}^{\mathcal{P}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\epsilon,j}^{\mathcal{P}})$, despite the variance $\boldsymbol{\Sigma}_{\epsilon,j}^{\mathcal{P}}$ have to be chosen to guarantee that the initialized coordination matrices $\boldsymbol{K}_j^{\mathcal{P}} \in \mathbb{R}^{m\times m}$ in (6) are symmetric and positive-semidefinite. Recalling that the sum of two symmetric and positive-semidefinite matrices is a symmetric and positive-semidefinite matrix, a symmetric and positive-semidefinite full stiffness matrix $\boldsymbol{K}_{t_i}^{\mathcal{P}}$ is obtained in (13) if the exploration matrices are all symmetric and positive-semidefinite. These properties are also retained when applying PI^2, where the parameters are temporal averaged, due to the fact that a weighted averaging over positive-semidefinite matrices yields a positive-semidefinite matrix [12].

Considering equation (14), the generalized cost term from PI^2 algorithm in [4] can be expressed as

$$S(\boldsymbol{\tau}_i) = \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} + \frac{1}{2}\sum_{j=i+1}^{N-1} \ddot{\boldsymbol{x}}_{t_j,d}^T \boldsymbol{R}\, \ddot{\boldsymbol{x}}_{t_j,d} \tag{15}$$

where ϕ_{t_N} is the terminal cost. The generalized cost of each rollout path defines a probability of a path $\boldsymbol{\tau}_i^k$ as

$$P(\boldsymbol{\tau}_i^k) = \frac{E_S(\boldsymbol{\tau}_i^k)}{\sum_{k=1}^{K} E_S(\boldsymbol{\tau}_i^k)} \tag{16}$$

with automatic sensitivity regulation term

$$E_S(\boldsymbol{\tau}_i^k) = \text{exp}\left(-h_\lambda \frac{S(\boldsymbol{\tau}_i^k) - \text{min}\, S(\boldsymbol{\tau}_i^k)}{\text{max}\, S(\boldsymbol{\tau}_i^k) - \text{min}\, S(\boldsymbol{\tau}_i^k)}\right) \tag{17}$$

that maximizes the discrimination between experienced paths for every time step i with sensitivity regulation constant h_λ [4]. Probability-weighted averaging over K rollouts yields the trajectory and stiffness parameter updates at each time step

$$\begin{aligned} \delta\boldsymbol{\theta}_{t_i,j}^{\mathcal{X}} &= \sum_{k=1}^{K} P(\boldsymbol{\tau}_i^k)\boldsymbol{\epsilon}_{t_i,j}^{\mathcal{X},k}\,, \\ \delta\boldsymbol{\theta}_{t_i,j}^{\mathcal{P}} &= \sum_{k=1}^{K} P(\boldsymbol{\tau}_i^k)\boldsymbol{\epsilon}_{t_i,j}^{\mathcal{P},k} \end{aligned} \tag{18}$$

and temporal weighted averaging over N time steps

$$\begin{aligned} \delta\boldsymbol{\theta}_j^{\mathcal{X}} &= \frac{\sum_{i=0}^{N-1}(N-i)\psi_{t_i,j}\delta\boldsymbol{\theta}_{t_i,j}^{\mathcal{X}}}{\sum_{i=0}^{N-1}\psi_{t_i,j}(N-i)} \, , \\ \delta\boldsymbol{\theta}_j^{\mathcal{P}} &= \frac{\sum_{i=0}^{N-1}(N-i)\psi_{t_i,j}\delta\boldsymbol{\theta}_{t_i,j}^{\mathcal{P}}}{\sum_{i=0}^{N-1}\psi_{t_i,j}(N-i)} \end{aligned} \tag{19}$$

leads eventually to parameter updates

$$\begin{aligned} \boldsymbol{\theta}_j^{\mathcal{X}} &\leftarrow \boldsymbol{\theta}_j^{\mathcal{X}} + \delta\boldsymbol{\theta}_j^{\mathcal{X}} \, , \\ \boldsymbol{\theta}_j^{\mathcal{P}} &\leftarrow \boldsymbol{\theta}_j^{\mathcal{P}} + \delta\boldsymbol{\theta}_j^{\mathcal{P}} \end{aligned} \tag{20}$$

individually performed for each basis function $j = 1, 2, \ldots, p$. As suggested in [4], update equations (19) take the activation of the j-th basis function $\psi_{t_i,j}$ from equation (8) into account. The pseudocode of the resulting C-PI^2 algorithm is given in Algorithm 1.

2.3 Learning Force Profiles using External Stimuli

Interactions with unfamiliar environments require to compensate for instability and unknown forces. Studies on human subjects have revealed in [13] that the central nervous system (CNS) reduces motion errors when interacting in novel environments trial after trial by adapting feedforward control to overcome environment forces. On the other hand, it has been observed that slightly perturbed arm motions tend to return to the undisturbed trajectory driven by spring-like muscle viscoelasticity and the stretch reflex [14]. Thus, a restoring force acts towards the undisturbed trajectory, whereby muscles and reflexes impose stiffness and damping that provide feedback during motions and can be adapted to compensate for dynamic environments [15]. To sum up, the human strategy is to adapt endpoint forces and viscoelasticity by minimizing error and effort as well as ensuring a constant stability margin [10].

To adopt this human movement behaviors, a RL reward/cost function can be designed to compensate for interaction forces in unknown environments by increasing exerted forces and impedance only when a task requires it, e.g. to reduce positional deviations. Furthermore, external forces-torques sensing capabilities may allow to incorporate a valuable learning stimuli in form of measured interaction forces into the reward/cost function to provide feedback about exerted forces. In this context, a cost function can be proposed as

$$r_t = w_{acc} \left\| \ddot{\boldsymbol{x}}_t \right\| + w_{gain} \sum_l \lambda_{t,l}^{\mathcal{P}} + w_{ext} \left\| \boldsymbol{T}_{t,ext} \right\| + w_{task} g_t \tag{21}$$

where g_t constitutes an arbitrary cost function term that describes the primary task goal. This goal term might for example penalize positional deviations for phases of a task where precision is required or it might be a more abstract description of a complex task. Other terms in (21) are lower level motor control variables. The term $\|\ddot{\boldsymbol{x}}_t\|$ penalizes high end-effector accelerations to avoid high-jerk motions. High stiffness gains are penalized through the sum $\sum_l \lambda_{t,l}^{\mathcal{P}}$ over the eigenvalues[3] of the stiffness/coordination matrix $\boldsymbol{K}_t^{\mathcal{P}}$ to facilitate a compliant behavior. Eventually, interaction forces are reduced through the term $\|\boldsymbol{T}_{t,ext}\|$ that brings the external force-torque sensing into consideration. The cost weights $w_{acc}, w_{gain}, w_{ext}, w_{task}$ allow to prioritize different aspects of the cost function (21) according to the task, whereby the task weight w_{task} should be chosen sufficiently high compared to the other weights in order to fulfill the task goal. Penalizing high accelerations, high stiffness gains

[3] Eigenvalues are used to quantify the magnitude of stiffness in full matrices.

Algorithm 1 C-PI^2 pseudocode

```
Require:
    $r_{t_i} = q_{t_i} + \ddot{\boldsymbol{x}}^T_{t_i,d}\boldsymbol{R}\,\ddot{\boldsymbol{x}}_{t_i,d}$        ▷ cost function
    $\phi_{t_N}$        ▷ terminal cost
    $h_{t_i,j}$        ▷ temporal weighting basis functions
    $\boldsymbol{\theta}^{\mathcal{X},init}_j,\ \boldsymbol{\theta}^{\mathcal{P},init}_j$        ▷ initial policy parameters
    $\boldsymbol{\Sigma}^{\mathcal{X}}_{\epsilon},\ \boldsymbol{\Sigma}^{\mathcal{P}}_{\epsilon}$        ▷ exploration noise variances
    $K$        ▷ number of rollouts per epoch
    $h_\lambda$        ▷ sensitivity regulation constant

while parameters $\boldsymbol{\theta}^{\mathcal{X}}_j,\ \boldsymbol{\theta}^{\mathcal{P}}_j$ not converged do
    Perform $K$ rollouts:
    for $k = 1, 2, \dots, K$ do
        Perform $k$-th rollout $\boldsymbol{\tau}^k$:
        for $i = 1, 2, \dots, N$ do
            Draw exploration samples:
            $\boldsymbol{\epsilon}^{\mathcal{X},k}_{t_i,j} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\mathcal{X},k}_{\epsilon,j})$;   $\boldsymbol{\epsilon}^{\mathcal{P},k}_{t_i,j} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\mathcal{P},k}_{\epsilon,j})$;
            Generate path and execute policy:
                $\boldsymbol{x}_{t_i,d} = \sum_{j=1}^{p} h_{t_i,j}(\boldsymbol{\theta}^{\mathcal{X},n}_j + \boldsymbol{\epsilon}^{\mathcal{X},k}_{t_i,j})$;
                $\boldsymbol{K}^{\mathcal{P}}_{t_i} = \sum_{j=1}^{p} h_{t_i,j}(\boldsymbol{\theta}^{\mathcal{P},n}_j + \boldsymbol{\epsilon}^{\mathcal{P},k}_{t_i,j})$;
                $\ddot{\boldsymbol{x}}^k_{t_i,d} = \boldsymbol{K}^{\mathcal{P}}_{t_i}(\boldsymbol{x}_{t_i,d} - \boldsymbol{x}_{t_i}) - \kappa^{\mathcal{V}}\dot{\boldsymbol{x}}_{t_i}$;
        end for
    end for
    Estimate parameter update for each time step:
    for $i = 1, 2, \dots, N$ do
        Evaluation for each time step and rollout $\boldsymbol{\tau}^k_i$:
        for $k = 1, 2, \dots, K$ do
                $S(\boldsymbol{\tau}^k_i) = \phi^k_{t_N} + \sum_{j=i}^{N-1} q^k_{t_j} + \frac{1}{2}\sum_{j=i+1}^{N-1} (\ddot{\boldsymbol{x}}^k_{t_j,d})^T \boldsymbol{R}\,\ddot{\boldsymbol{x}}^k_{t_j,d}$;
                $E_S(\boldsymbol{\tau}^k_i) = \exp\left(-h_\lambda \frac{S(\boldsymbol{\tau}^k_i) - \min S(\boldsymbol{\tau}^k_i)}{\max S(\boldsymbol{\tau}^k_i) - \min S(\boldsymbol{\tau}^k_i)}\right)$;
                $P(\boldsymbol{\tau}^k_i) = \frac{E_S(\boldsymbol{\tau}^k_i)}{\sum_{k=1}^{K} E_S(\boldsymbol{\tau}^k_i)}$;
        end for
        Probability-weighted averaging over $K$ rollouts:
            $\delta\boldsymbol{\theta}^{\mathcal{X}}_{t_i,j} = \sum_{k=1}^{K} P(\boldsymbol{\tau}^k_i)\boldsymbol{\epsilon}^{\mathcal{X},k}_{t_i,j}$;
            $\delta\boldsymbol{\theta}^{\mathcal{P}}_{t_i,j} = \sum_{k=1}^{K} P(\boldsymbol{\tau}^k_i)\boldsymbol{\epsilon}^{\mathcal{P},k}_{t_i,j}$;
    end for
    Update through temporal averaging over time steps:
        $\boldsymbol{\theta}^{\mathcal{X},n+1}_j = \boldsymbol{\theta}^{\mathcal{X},n}_j + \frac{\sum_{i=0}^{N-1}(N-i)\psi_{t_i,j}\delta\boldsymbol{\theta}^{\mathcal{X}}_{t_i,j}}{\sum_{i=0}^{N-1}\psi_{t_i,j}(N-i)}$;
        $\boldsymbol{\theta}^{\mathcal{P},n+1}_j = \boldsymbol{\theta}^{\mathcal{P},n}_j + \frac{\sum_{i=0}^{N-1}(N-i)\psi_{t_i,j}\delta\boldsymbol{\theta}^{\mathcal{P}}_{t_i,j}}{\sum_{i=0}^{N-1}\psi_{t_i,j}(N-i)}$;
end while
```

and high interaction forces facilitates safe human-robot interaction and allows to compensate for interaction forces in unknown environments by increasing these variables trial after trial when phases of the task require it to succeed in the primary task goal. This allows a human-like adaption of a force profile for interactions with environment and humans.

2.4 Experimental Results

The effectiveness of the C-PI^2 in combination with an external force-torque sensor signal as learning stimuli is demonstrated on a 7 DoFs KUKA LWR IV+ manipulator.

In order to refine and adapt the policy, the parameter space must be explored by perturbing the parameters through exploration noise $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon)$ with variance $\boldsymbol{\Sigma}_\epsilon$ drawn form a zero-mean Gaussian distribution. These exploration vectors may be sampled at each time step, which results in highly varying noise signals. Another option is to draw one exploration sample at the beginning of a rollout and to keep it constant during the entire rollout. We decided to use a constant exploration noise in all the experiments to quickly reach convergence and thus to require a smaller number of rollouts for a good solution.

The only open tuning parameter in C-PI^2 (as for PI^2) is the magnitude of the exploration noise ξ. The exploration noise magnitude is decreased over the number of updates ϑ by multiplying it with a decay parameter γ^ϑ set to $\gamma^\vartheta = 0.99^\vartheta$ to increase the exploitation of the learned information.

The number of re-used rollouts for learning in C-PI^2 is set to $\sigma = 5$ and the number of rollouts per epoch used for parameter updating is set to $K = 10$. Thus, C-PI^2 updates after the first 10 rollouts and then after 5 more rollouts using these 5 new rollouts and the best 5 of the latter epoch and this is repeated for all subsequent updates.

Object Pushing

The effectiveness of the novel C-PI^2 algorithm in combination with an external force-torque stimuli incorporated in the RL reward function is evaluated on a real 7 DoFs KUKA/DLR Lightweight Robot (LWR). The task thereby consists of pushing a box with the end-effector at constant velocity. Thereby, C-PI^2 has to utilize its ability of simultaneously learning trajectory and stiffness parameters in order to succeed in this task. Pushing operations include a large number of possibly unstable contacts. Hence, it is interesting to evaluate the performance of the C-PI^2 with an external force-torque reward stimuli.

The robot has to push the box on a table until it reaches the goal position $\boldsymbol{x}_{t_N} = [-0.5087, 0.4232, 0.0501]^T$. The end-effector of the robot is initially placed at $\boldsymbol{x}_{t_0} = [-0.5459, -0.0144, 0.0501]^T$. Hence, the major part of the movement occurs in the second Cartesian dimension (40.88 cm in x_2-direction). The box is placed approximately 5 cm away from the end-effector position in the x_2-direction, as illustrated in figure 2. The physical contact will be established soon after the movement is initiated.

The estimated external torques are considered in the RL cost function

$$r = w_{acc} \|\ddot{\boldsymbol{x}}\| + w_{gain} \sum_l \lambda_l^{\mathcal{P}} + w_{ext} \|\boldsymbol{T}_{ext}\| + w_{vel} g \tag{22}$$

while the terminal cost is chosen as

$$\phi_{t_N} = w_{move} \|\dot{\boldsymbol{x}}\| + w_{goal} \|\boldsymbol{x}_{t_N} - \boldsymbol{x}\| \ . \tag{23}$$

As long as it is compatible with the primary task goal defined in g, the cost function (22) avoids high-jerk motions, encourages compliant behaviors ($\lambda_l^{\mathcal{P}}$ are the eigenvalues of full stiffness matrix $\boldsymbol{K}^{\mathcal{P}}$) and reduces interaction forces using torque sensing. The terminal cost penalizes distance from the end-effector goal position and motions after a planned movement duration of 10 s. The primary goal of the object pushing task in hand is

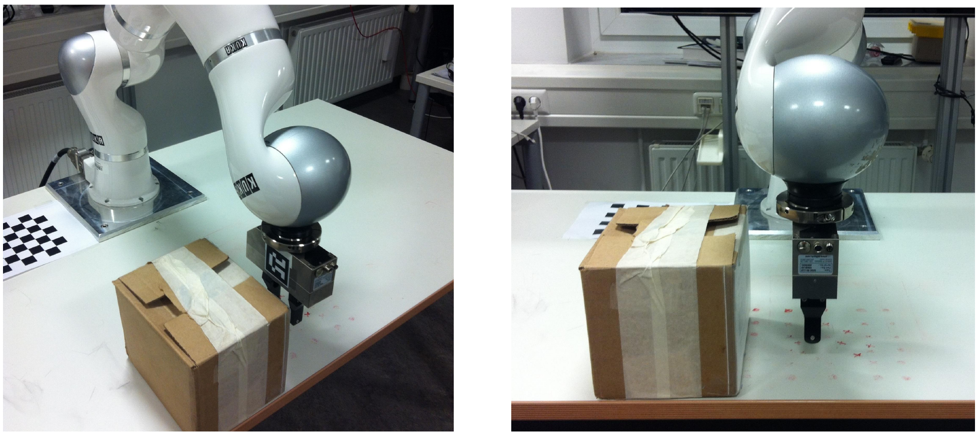


Figure 2: 7 DoFs KUKA/DLR Lightweight Robot (LWR) in its initial pose for an object pushing task: The aim of this RL task is to push a box to a given goal position along the edge of a table. Physical contact between robot end-effector and object will be established soon after the movement is initiated.

defined as holding a constant end-effector velocity of 0.1 m/s in x_2-direction and zero in x_1- and x_3-direction while moving from start to end position, which yields in

$$g = \left\| \begin{pmatrix} 0 \\ 0.1 \\ 0 \end{pmatrix} - \dot{\boldsymbol{x}} \right\| . \tag{24}$$

Hence, the cost weight w_{vel} in equation (22) must be chosen sufficiently high to ensure the primary relevance of this cost function term. The cost function weights are chosen as $w_{acc} = 1\text{e}3, w_{gain} = 5, w_{ext} = 8\text{e}3, w_{vel} = 7\text{e}5, w_{move} = 5\text{e}4, w_{goal} = 5\text{e}5$. The policy is initialized through a user demonstration by manually guiding the robot end-effector without the object. The Correlated DMPs are endowed with $p = 10$ basis functions equally distributed over the movement duration of 10 s with variances $\Sigma_j^{\mathcal{T}} = 0.3$ ($j = 1, 2, \dots, p$) and the varying stiffness is initialized through PbD within the range $\left[\kappa_{min}^{\mathcal{P}} = 300, \kappa_{max}^{\mathcal{P}} = 500\right]$ and with initial gain $\kappa^{\mathcal{P}} = 400$. Thereby, the stiffness matrices are deliberately initialized with low[4] values to facilitate a compliant behavior as long as the task goal allows it, which means the robot has to learn to increase its stiffness when the task requires it.

The standard deviation of trajectory exploration is set to $\xi^{\mathcal{X}} = 0.01$ and the stiffness exploration noise is generated as covariance matrices drawn from a zero-mean Gaussian distribution with a standard deviation that is scaled according to the initial stiffness values multiplied with factor 0.02. This results in stiffness exploration matrices with a standard deviation of 2% of the initial values which are then added to the stiffness parameter matrices that remain symmetric and positive semi-definite. Furthermore, no noise will be added to the first and last basis functions and neither will their parameters be updated during learning to maintain smooth motion transitions from starting position towards desired trajectory or from desired trajectory towards the goal position.

Figure 3 compares the velocities of the initial trajectory learned through PbD and the trajectory refined with C-PI^2 (after 150 rollouts). The initial velocity learned by PbD presents high peaks that reveal the instability of this robot-object interaction. At the beginning the robot is extremely compliant and it is hardly able to exert enough force to overcome the inertia of the object and the friction between object and table surfaces. After learning with C-PI^2 the robot velocity is significantly closer to the desired constant value. The usage of an external force-torque stimuli in the reward function helps to keep interaction forces as low as the primary task

[4] The realizable Cartesian stiffness range of the KUKA/DLR Lightweight Robot lays between 100 and 2000.

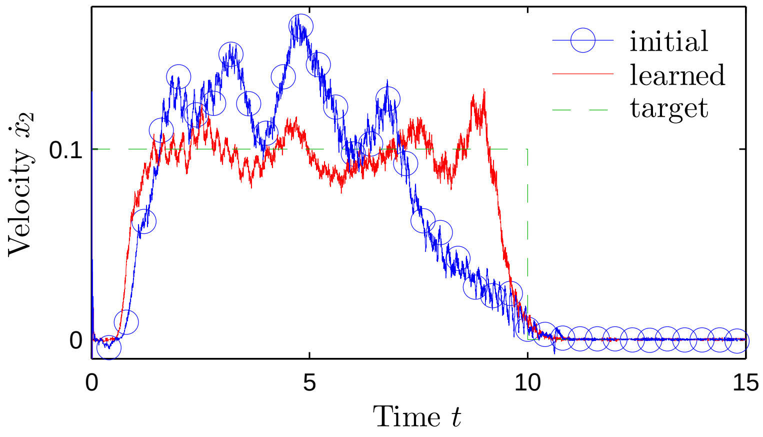


Figure 3: Robot velocity (x_2-direction) in the object pushing task: The initial velocity (blue line) is refined using C-PI^2 (red line) to stay as close as possible to a constant target velocity (green dashed line).

allows it. The measured external torques compared in figure 4 show that the forces exerted on the object are reduced applying C-PI^2.

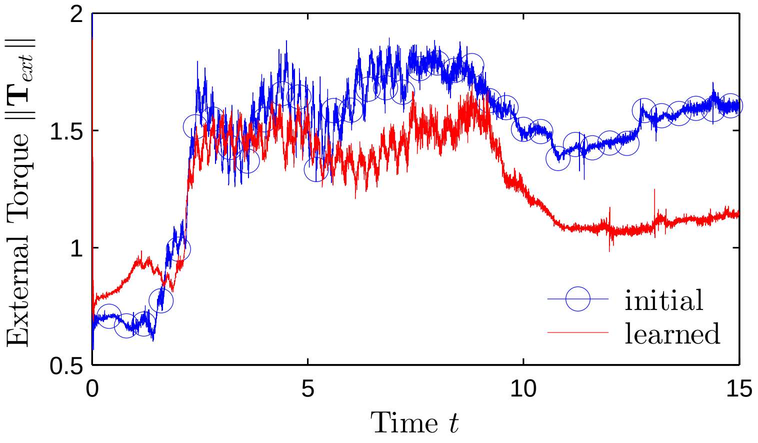


Figure 4: The initial external torque (blue line) corresponds to a reproduction of the user demonstration. The norm of the interaction forces is reduced applying C-PI^2 (red line).

References

[1] M. Saveriano and D. Lee, “Point Cloud based Dynamical System Modulation for Reactive Avoidance of Convex and Concave Obstacles,” IEEE Intl Conf. on Intelligent Robots and Systems, pp. 5380–5387, 2013.

[2] M. Saveriano and D. Lee, “Distance based Dynamical System Modulation for Reactive Avoidance of Moving Obstacles,” IEEE Intl Conf. on Robotics and Automation, pp. 5618–5623, 2014.

[3] S.M. Khansari-Zadeh and A. Billard, “A Dynamical System Approach to Realtime Obstacle Avoidance,” Autonomous Robots, vol. 32, no. 4, pp. 433–454, 2012.

[4] E. Theodorou, J. Buchli and S. Schaal, “A generalized path integral control approach to reinforcement learning,” Journal of Machine Learning Research, vol. 11, pp. 3137–3181, 2010.

[5] J. Kober and J. Peters, “Learning motor primitives for robotics,” IEEE Intl Conf. on Robotics and Automation, pp. 2112–2118, 2009.

[6] J. Buchli, F. Stulp, E. Theodorou and S. Schaal, “Learning variable impedance control,” International Journal of Robotics Research, vol. 30, pp. 820–833, 2011.

[7] E. Todorov and M. Jordan, “Optimal feedback control as a theory of motor coordination,” Nature Neuroscience, vol. 5, pp. 1226-1235, 2002.

[8] T. Flash and N. Hogan, “The Coordination of the Arm Movements: An Experimentally Confirmed Mathematical Model,” Neurology, vol. 5, no. 7, pp. 1688–1703, 1985.

[9] M. Rosenstein, A. Barto and R. Van Emmerik, “Learning at the level of synergies for a robot weightlifter,” Robotics and Autonomous Systems, vol. 54, n. 8, pp. 706–717, 2006.

[10] D. Franklin, E. Burdet, K. Peng Tee, R. Osu, C.-M. Chew, T. Milner and M. Kawato, “CNS Learns Stable, Accurate, and Efficient Movements Using a Simple Algorithm,” Journal of Neuroscience, vol. 28, n. 44, pp. 11165–11173, 2008.

[11] S. Calinon, I. Sardellitti and D. Caldwell, “Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies,” IEEE Intl Conf. on Intelligent Robots and Systems, pp. 249–254, 2010.

[12] J. Dattorro, “Convex Optimization & Euclidean Distance Geometry,” Meboo Publishing, 2011.

[13] R. Shadmehr and F. Mussa-Ivaldi, “Adaptive representation of dynamics during learning of a motor task,” Journal of Neuroscience, vol. 14, no. 5, pp. 3208–3224, 1994.

[14] J. Won and N. Hogan, “Stability properties of human reaching movements,” Experimental Brain Research, vol. 107, no. 1, pp. 125–136, 1995.

[15] T. Milner and C. Cloutier, “Compensation for mechanically unstable loading in voluntary wrist movement,” Experimental Brain Research, vol. 94, no. 3, pp. 522–532, 1993.