Cooperative Suspended Object Manipulation Using Reinforcement Learning and Energy-based Control

Ivana Palunko[1], Philine Donner[2], Martin Buss[2] and Sandra Hirche[3]

***Abstract*— Cooperative dynamic object manipulation can extend the manipulation capabilities of robot-robot and human-robot teams. In order to be able to inject energy into various suspended objects of unknown parameters, in this paper we propose an adaptive controller which combines reinforcement learning with energy based swing-up control. The proposed controller is successfully verified in a single robot and human-robot experimental setup for different types of suspended objects.**

I. INTRODUCTION

With research pushing robots towards becoming part of our every day life, robots will have to be able to physically interact with humans. Physical interaction can happen through direct contact, e.g. in elderly care, or through an object that is jointly manipulated. Whereas quasi-static object transport scenarios have been investigated in a number of papers [1]–[3], dynamic object manipulation has been hardly considered in human-robot cooperation. Fig. 1 depicts an example scenario for cooperative dynamic object manipulation as a suspended object swing. Swing motion is frequently used in robotics, with cable-suspended mechanisms [4] and brachiating robots [5] being just two examples. In the context of human-robot interaction, however, to the best of the authors' knowledge the only works that make use of swing-motion are on rope turning [6], [7]. In order to approach the complex task of flexible object swinging Fig. 1(b), we split it up into two extremes: Fig. 1(a) the swinging of a rigid object, in which case the arms of the two partners together with the object form an oscillating entity, and Fig. 1(c) the swinging of a pendulum-like object, which can oscillate itself. Our previous work on cooperative swinging concentrates on pendulum-like objects as the trapezoidal pendulum displayed in Fig. 1(c) [8] and a simpler v-shaped pendulum with a point-mass instead of the cylindrical mass [9]. The presented energy based approach enables cooperative object swing-up to a desired energy level, at which the object could be released for a goal directed throw. The controller can render a robotic leader and follower, based on whether the desired object energy is known to the robot in advance. One important short-coming of the model-based approach is that the object parameters need to be known and that the approach

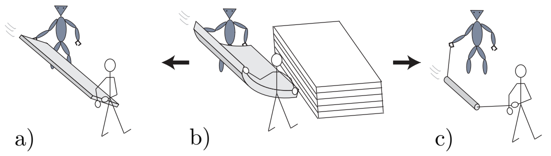


Fig. 1: Flexible object swinging (b) interpreted as a mixture of rigid object swinging (a) and pendulum swinging (c).

cannot directly be applied to different types of suspended objects. In order to move from the extremes to the final goal of flexible object swinging (Fig. 1), the robot needs to be able to adapt to different and unknown types of objects.

To achieve this, in this paper we propose an adaptive algorithm based on least square policy iteration (LSPI) for online adaptation of the control signal used for a single and cooperative manipulation of various suspended objects. Least square policy iteration (LSPI) is a type of reinforcement algorithm. It is based on the temporal difference algorithm (TD-algorithm) developed by [10] and was first introduced by [11]. Policy iteration iteratively evaluates and improves control policies. Least square policy iteration has been successfully implemented to balance an inverted pendulum and riding a bicycle [12], [11]. In [13] least square policy iteration was used for suspended load trajectory tracking and in [14] value iteration was successfully used for load swing damping in suspended load transport using a small quadrotor. In this paper two online adaptations, based on least square policy iteration, estimate the natural frequency of the pendulum and an amplitude factor that specifies the amount of energy injected into the system. The proposed controller is verified in a single robot and human-robot experimental setup for different types of suspended objects. The experimental results show that because of the online adaptation the proposed controller can compensate for changes in the system dynamics as well as for disturbances induced by the human operator. One of the advantages of this method is that no explicit knowledge about the system model is necessary in the design process. However, partial model knowledge, is introduced through an energy based swing-up control framework. Another advantage is that the online learning algorithm enables the robot to interact with the changing environment.

The rest of the paper is organized as follows: in Section II, we formally state the problem, Section III discusses the algorithm and methodology in detail, Section IV shows experimental results with different suspended objects, Section V presents the closing remarks.

[1]I. Palunko is with the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, ivana.palunko@fer.hr

[2]P. Donner and M. Buss are with the Institute of Automatic Control Engineering and TUM-IAS, Technische Universität München, 80290 München, Germany, {philine.donner,mb}@tum.de

[3]S. Hirche is with the Institute for Information-oriented Control, Technische Universität München, 80290 München, Germany, hirche@tum.de

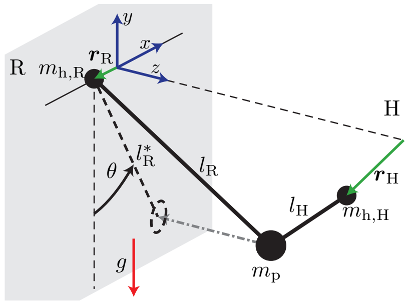


Fig. 2: The v-pendulum: mass m_p suspended from two handles of mass $m_{\mathrm{h,R/H}}$ through ropes of length $l_{\mathrm{R/H}}$.

II. PROBLEM STATEMENT

This paper investigates the problem of suspended object manipulation by two agents with unknown model and system parameters. The control goal is to reach a desired energy level V_θ^d, which is equivalent to a desired maximum deflection angle θ_V^d. Without loss of generality, we assume one agent to be a robot R and the other agent to be a human H. Agents interact with the suspended object through acceleration of handle masses $m_{\mathrm{h,R/H}}$. The only controllable input is the acceleration of the robot $u_\mathrm{R} = \ddot{\nu}_{\mathrm{R},x}$, which we restrict to be acting in only one dimension for simplicity and as a minimum requirement. Forces acting at the robot's end-effector $\boldsymbol{F}_\mathrm{R}$ are the only feedback the robot receives about the state of the pendulum. Consequently, we are looking for a control law

$$u_\mathrm{R} = \ddot{\nu}_{\mathrm{R},x} = f(\boldsymbol{F}_\mathrm{R}), \tag{1}$$

$$\text{with } \left|V_\theta^\mathrm{d} - V_\theta(t > T_\mathrm{s})\right| \leq \epsilon_\theta \quad \text{for } 0 < T_\mathrm{s} < \infty. \tag{2}$$

Thus, we require the energy error $\Delta V_\theta = V_\theta^\mathrm{d} - V_\theta$ to stay below ϵ_θ the latest after the settling time $t = T_\mathrm{s}$. In contrast to our previous work [8], [9], in this paper we concentrate on the robot being a leader, meaning the robot knows the desired energy level V_θ^d.

Remark 1: In this setting, the term cooperative indicates predefined fixed roles and a shared goal. In the experiments the proposed controller can interact with a human as a follower. Furthermore, we test the controller performance to disturbances introduced by the human. Thus, having a robot leader is a first step towards fully cooperative suspended object manipulation.

Remark 2: The v-pendulum from [9] displayed in Fig. 2 serves as an example for a two-agent suspended object to explain our control method. However, our approach can also be applied to more complex pendulum-like objects, as we show for the t-pendulum (Fig. 1(c)) in our experiments. The t-pendulum can swing in an undesired mode of oscillation of angle ψ around the vertical axis in addition to the desired θ-oscillation and is therefore more difficult to be controlled [8].

III. PENDULUM SWING UP USING ONLINE LEAST SQUARE POLICY ITERATION

In this section we describe the adaptive controller which combines energy based swing-up control and online least square policy iteration. The block diagram showing the overall control structure is given in Fig.4.

First we start with the model based control presented in [9] and [8]. This approach uses the energy based swing-up control for a simple linearly accelerated pendulum from [15]. Following this principle we define the control input for the robot as an acceleration of the form

$$\ddot{\nu}_{\mathrm{R},x} \approx a\ {\omega_\theta}^2 \sin(\varphi). \tag{3}$$

The amplitude factor a specifies the direction and amount of energy flow to the pendulum. The natural frequency ω_θ approximately relates to the phase angle φ [15]

$$\varphi(t) \approx \omega_\theta t + \varphi_0. \tag{4}$$

From this we can see that the control law (3) excites the pendulum close to its natural frequency and thus times the energy flow. Fig. 3 shows a phase portrait with the inscribed phase angle $\varphi = \arctan(-\frac{\dot{\theta}}{\theta \hat{\omega}_\theta})$. The normalization of the angular velocity $\dot{\theta}$ with an approximation of the natural frequency $\hat{\omega}_\theta$ results in a phase plot that is close to a circle for no energy injection (see dashed circle in Fig. 3). As

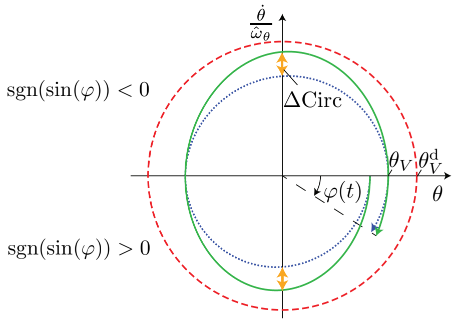


Fig. 3: Phase portrait of a simple pendulum during swing-up: solid green line (blue dotted line) indicates a too small (good) estimation and of the natural frequency $\hat{\omega}_\theta$. The red dashed line shows the desired phase portrait with maximum deflection angle θ_V^d. Note that the yellow errors do not exactly represent but relate to $\Delta\mathrm{Circ}$.

stated in Section II, the robot has to achieve the task goal (2) solely using the force measurement at its end-effector $\boldsymbol{F}_\mathrm{R}$. The deflection angle θ is obtained from

$$\theta = \arctan\left(\frac{-F_{\mathrm{R,p},x}}{F_{\mathrm{R,p},y}}\right), \tag{5}$$

with the force $\boldsymbol{F}_{\mathrm{R,p}}$ obtained from $\boldsymbol{F}_\mathrm{R}$ after dynamic compensation of the handle mass $m_{\mathrm{h,R}}$. In [8], [9] the small angle approximation $\omega_{0,\theta}$ was used to normalize the angular velocity $\dot{\theta}$, which requires knowledge of the projected length l_R^* (Fig. 2). As in [8], [9], we adapt the

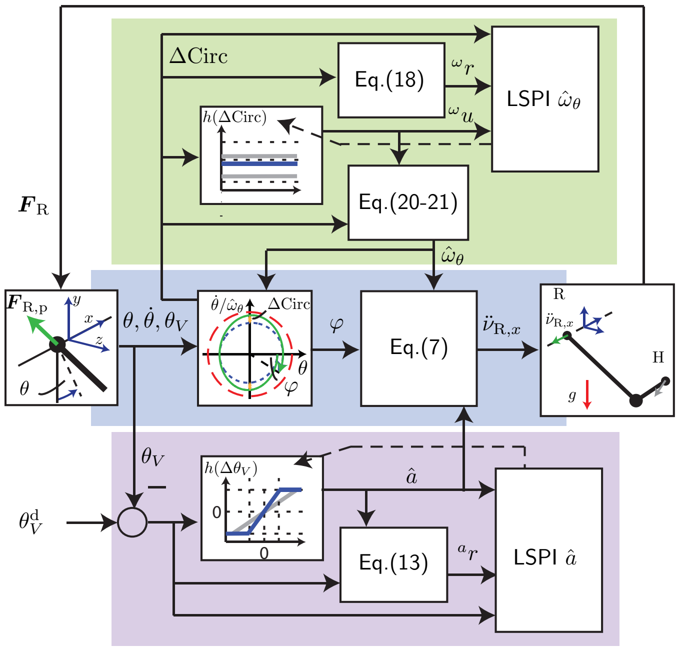


Fig. 4: Block diagram of the adaptive controller. Energy based control law (on blue), LSPI based learning of amplitude a (on purple) and natural frequency $\hat{\omega}_\theta$ (on green).

approach from [15] and feed a reference trajectory

$$\nu_{\mathrm{R,ref},x} = a\frac{1}{|G_\theta(j\hat{\omega}_\theta)|}\sin(\varphi - \pi - \angle G_\theta(j\hat{\omega}_\theta)) \tag{6}$$

through the transfer function

$$G(j\omega) = \frac{(j\omega)^2(\frac{\hat{\omega}_\theta}{c_0})^2}{(j\omega)^2 + 2\zeta\frac{\hat{\omega}_\theta}{c_0}(j\omega) + (\frac{\hat{\omega}_\theta}{c_0})^2}, \tag{7}$$

where c_0 and ζ are design variables. This results in an acceleration of approximately

$$\begin{aligned}\ddot{\nu}_{\mathrm{R},x} \simeq a\,{\omega_\theta}^2\frac{|G_\theta(j\omega_\theta)|}{|G_\theta(j\hat{\omega}_\theta)|}\sin(\varphi - \angle G_\theta(j\hat{\omega}_\theta) + \angle G_\theta(j\omega_\theta))\\ \approx a\,{\omega_\theta}^2\sin(\varphi).\end{aligned} \tag{8}$$

Consequently, higher frequencies than the approximated natural frequency $\hat{\omega}_\theta$ are damped, while we are able to approximately compensate for the amplitude and phase shift $|G_\theta(j\omega_\theta)|$ and $\angle G_\theta(j\omega_\theta)$ caused by the filter $G(j\omega)$ at the natural frequency ω_θ.

We apply this simple pendulum swing-up control law to complex suspended objects through projection onto a simple abstract pendulum with deflection angle θ (Fig. 2). Our goal is to enable the robot to manipulate objects of variable shape and size cooperatively or unassisted without retuning the controller. We achieve this by introducing the adaptation of the parameter a and ω_θ as depicted in Fig. 4. Instead of only adapting to the unknown object model, we also want to inject energy efficiently into the system using optimal adaptation. There are different structures and types of adaptive optimal algorithms described in literature, but the common feature is that they all focus on finding a solution to the nonlinear Bellman equation [16]. In this paper we use the online version of least-square policy iteration (LSPI) which interacts with the system at every iteration k through three variables $\boldsymbol{x_k}, u_k, \boldsymbol{r_k}$. First is the state $\boldsymbol{x_k}$ which represents the measurable system features important for the problem we want to solve, not necessarily the state in the control systems sense. Second, is the action u_k, based on the optimal policy $h_p(\boldsymbol{x_k})^*$, usually chosen as a controllable input of the state $\boldsymbol{x_k}$. Third, is the reward function $\boldsymbol{r_k}$, a negative definite function of the state and/or the action, which serves as a measure how far the algorithm is from the goal after the action u_k is taken. Since the LSPI solves the optimal control problem, to avoid local minima, the algorithm needs to have an exploration phase, which is achieved by taking a random action in the action space every ε_k steps. As the time passes, the algorithm relays more on the current policy $h_p(\boldsymbol{x})$ than on the exploration, so ε_k is taken less often with the μ rate as $k \to \infty$. Using policy iteration, policies $h_p(\boldsymbol{x})$ are evaluated by iteratively constructing an estimate of the state $\boldsymbol{x}$ and action u value function, a $\boldsymbol{Q}$-function. In this paper, as well as in [13], the state-action approximate value function is defined by

$$\hat{\boldsymbol{Q}}(\boldsymbol{x_k}, u_k) = \boldsymbol{\Phi}^T(\boldsymbol{x_k}, u_k)\boldsymbol{\alpha_p}, \tag{9}$$

where

$$\boldsymbol{\Phi}(\boldsymbol{x_k}, u_k) = \boldsymbol{\psi}(u_k) \otimes \boldsymbol{\phi}(\boldsymbol{x_k})$$

is the Kronecker product of the basis function vectors $\boldsymbol{\psi}(u_k)$ and $\boldsymbol{\phi}(\boldsymbol{x_k})$ formed with Chebyshev polynomials; $\boldsymbol{\alpha_p}$ is the approximation parameters vector which needs to be learned. The parameter vector $\boldsymbol{\alpha_p}$ is obtained from the projected Bellman equation for model-free policy iteration which according to [11], and can be written as

$$\boldsymbol{\Gamma}_k\boldsymbol{\alpha_p} = \gamma\boldsymbol{\Lambda}_k\boldsymbol{\alpha_p} + \boldsymbol{z}_k, \tag{10}$$

where $\gamma \leq 1$ is a positive scalar and

$$\begin{aligned}\boldsymbol{\Gamma}_0 = \beta_\Gamma\boldsymbol{I}, \quad \boldsymbol{\Lambda}_0 &= \boldsymbol{0}, \quad \boldsymbol{z}_0 = \boldsymbol{0},\\ \boldsymbol{\Gamma}_k &= \boldsymbol{\Gamma}_{k-1} + \boldsymbol{\phi}(\boldsymbol{x}_k, u_k)\boldsymbol{\phi}(\boldsymbol{x}_{k-1}, u_{k-1})^T,\\ \boldsymbol{\Lambda}_k &= \boldsymbol{\Lambda}_{k-1} + \boldsymbol{\phi}(\boldsymbol{x}_k, u_k)\boldsymbol{\phi}(\boldsymbol{x}_k, h(\boldsymbol{x}_{k+1}))^T,\\ \boldsymbol{z}_k &= \boldsymbol{z}_{k-1} + \boldsymbol{\phi}(\boldsymbol{x}_k, u_k)r_k,\end{aligned} \tag{11}$$

where $\boldsymbol{\Gamma}_k$, $\boldsymbol{\Lambda}_k$ and $\boldsymbol{z}_k$ represent the policy evaluation mapping and projection operator of the Bellman equation [12] and are updated every iteration step k. The parameter vector $\boldsymbol{\alpha_p}$ is calculated every p steps from (10) and is used to improve the Q-function estimate (9). Considering this, new, improved policies are obtained from

$$h_p(x_k) \in arg\max_u \hat{\boldsymbol{Q}}(\boldsymbol{x}_k, u), \tag{12}$$

where $\hat{\boldsymbol{Q}}(\boldsymbol{x}_k, u)$ is constructed from (9) using $\boldsymbol{\alpha_p}$. Due to space limitation we omit the more detailed discussion. For more details the reader is referred to [13] and [11]. To achieve simultaneous adaptation of two parameters a and ω from (3) we define two separate learning loops as described in the remainder of this section.

Online adaptation of the amplitude factor a: In [8], [9] the amplitude factor a was chosen based on the energy error ΔV_θ and a saturated linear mapping, which required model knowledge as well as parameter tuning. In this paper we monitor the maximum deflection angle θ_V as an equivalent to the model dependent energy V_θ. We define the state-action-reward set as ${}^aS = \{{}^ax_i, {}^a u_i, {}^a r_i\}$ where i denotes the algorithm iteration step. The measured state ${}^ax_i = \theta^d_{Vi} - \theta_{Vi}$ is defined as the error between the desired deflection angle θ^d_{Vi} and the current, measured deflection angle of the system θ_{Vi}. The control action ${}^au_i = a$ represents the amplitude factor a from (3) and is constructed as

$$ {}^au_i = h_\mu({}^ax_i){}^ax_i, $$

where

$$ h_\mu({}^ax_i) \rightarrow \begin{cases} \text{u.r.a. in } {}^aU & \text{every } \varepsilon_a \text{ iterations;} \\ h_\mu({}^ax_i) & \text{otherwise,} \end{cases} \tag{13} $$

where u.r.a. is a uniform random action providing the algorithm exploration phase every ε_a steps and $h_\mu({}^ax_i)$ is the policy obtained from (12) using $\boldsymbol{\alpha_\mu}$ vector parameter updated every μ steps by solving (10). The reward function ${}^ar_i$ is defined as

$$ {}^a\rho = -({}^aK_x{}^ax_i^2 - {}^aK_{xu}{}^ax_i{}^au_i + {}^aK_u{}^au_i^2), $$

$$ {}^ar_i \rightarrow \begin{cases} 0 & \text{if } {}^a\rho > 0; \\ {}^a\rho & \text{if } {}^a\rho \leq 0, \end{cases} \tag{14} $$

where ${}^aK_x$, ${}^aK_u$ and ${}^aK_{xu}$ are positive constants. When amplitude $a > 0$, energy is injected into the system and when $a < 0$ energy is decreased. Therefore we introduce the term $-{}^aK_{xu}{}^ax_i{}^au_i$ into the reward function to penalize the decrease in energy when our current energy is below the desired one. The goal of the LSPI is to find the optimal policy $h^*_\mu({}^ax_i)$ that maximizes the return ${}^ar_i$ from an initial state ${}^ax_0$.

Assumptions:

1) the state ${}^ax$ is controllable and bounded $[{}^ax_m \quad {}^ax_M]$,
2) the control action ${}^au$ is bounded $[{}^au_m \quad {}^au_M]$.

The bounds are defined because of the Q-function approximators [13]. Since Chebyshev polynomials are defined on the interval from -1 to 1, the bounds on the state and control action are needed for domain mapping. The measured state, suspended object deflection angle, is inherently bounded $\theta_V \in [0° \quad 360°]$, so the state bounds are physically imposed. Since we want to exploit the energy properties of the robot control acceleration input (3) we restrict the robot range of motion with ${}^au \in [-0.1 \quad 0.1]$. Without the loss of generalization of the LSPI algorithm, the control action bounds can be expanded to the robot workspace. The further discussion about the constants as well as their values are given in Section IV.

Online adaptation of the frequency $\hat{\omega}_\theta$:

Similarly as for the amplitude a adaptation we define the state-action-reward set as ${}^\omega S = \{{}^\omega x_j, {}^\omega u_j, {}^\omega r_j\}$ where j denotes the algorithm iteration step. The state we measure is denoted as ${}^\omega x_j = \Delta\mathrm{Circ}$ where $\Delta\mathrm{Circ}$ is as the difference of the phase-space to a circle

$$ \Delta\mathrm{Circ} = \hat{\omega}^2_\theta\theta^2_V - \dot{\theta}^2_V, \tag{15} $$

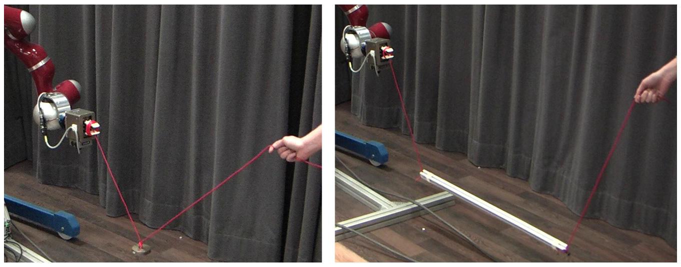


Fig. 5: Experimental setup with the v-pendulum (left) and the t-pendulum (right).

with $\dot{\theta}_V$ being the maximum angular velocity (Fig.3). The control action ${}^\omega u_j \in {}^\omega U$, with ${}^\omega U = [0 \quad {}^\omega u_M]$ is defined

$$ {}^\omega u_j \rightarrow \begin{cases} \text{u.r.a. in } {}^\omega U & \text{every } \varepsilon_\omega \text{ iterations;} \\ h_\kappa({}^\omega x_j) & \text{otherwise,} \end{cases} \tag{16} $$

where κ denotes the iteration in which the new parameter vector ${}^\omega\alpha_\kappa$ for the Q-function is calculated from (10) and ε_ω is the exploration phase step. We obtain the $\hat{\omega}_\theta$ from an iterative update rule

$$ \hat{\omega}_{\theta,j} = \hat{\omega}_{\theta,j-1} - {}^\omega u_j{}^\omega x_j, \tag{17} $$

where the control output from the LSPI algorithm ${}^\omega u_j$ is an optimal adaptive gain multiplying the ${}^\omega x_j$ and $\hat{\omega}_{\theta,0} = \omega_{\theta m}$. Since we do not formally prove the convergence of the proposed iterative update rule, we introduce the saturation

$$ \hat{\omega}_{\theta,j} \rightarrow \begin{cases} \omega_{\theta m} & \text{if } \hat{\omega}_{\theta,j} < \omega_{\theta m} \;, \\ \omega_{\theta M} & \text{if } \hat{\omega}_{\theta,j} > \omega_{\theta M}, \\ \hat{\omega}_{\theta,j} & \text{otherwise,} \end{cases} $$

where the bounds $\omega_{\theta m}$ and $\omega_{\theta M}$ are chosen in such a way that the estimated $\hat{\omega}_\theta$ can compensate for pendulum cable lengths from approximately $l \in [0.1\,\mathrm{m} \quad 10\,\mathrm{m}]$. From the experimental results we can see that the the control action representing the estimated $\hat{\omega}_\theta$ is not saturated during the experiment. The reward function is defined as

$$ {}^\omega r_j = -{}^\omega K_x{}^\omega x_j{}^2, \tag{18} $$

where ${}^\omega K_x$ is a positive constant. This reward function tells us that we want to minimize the error $\Delta\mathrm{Circ}$. The goal is to find the optimal policy $h^*_\kappa({}^\omega x_j)$ using (12) that maximizes the return from any initial state ${}^\omega x_0$.

Assumptions:

1) the state ${}^\omega x$ is controllable and bounded $[{}^\omega x_m \quad {}^\omega x_M]$,
2) the control action ${}^\omega u_j$ is bounded $[{}^\omega u_m \quad {}^\omega u_M]$.

The same argument for the bounds stands here as before.

IV. HUMAN-ROBOT EXPERIMENTS

A. Experimental setup

The proposed control algorithm presented in Section III is verified on a human-robot cooperative load manipulation task. The experiment is performed with a simple pendulum, the v- and the t-pendulum (see Fig. 5) in the ITR lab at TU München [17]. As the robotic actuator we use a commercially available *KUKA LWR 4+* manipulator. The forces $\boldsymbol{F}_\mathrm{R}$ are obtained from a *JR3* 6DoF force/torque sensor mounted between the manipulator and a $m_{\mathrm{h,R}}$ =1.75 kg gripper, to which we tie the rope. The deflection angle θ is calculated using (5), with $\boldsymbol{F}_{\mathrm{R,p}}$ obtained by dynamically compensating for the gripper mass $\boldsymbol{F}_{\mathrm{R,p}} = \boldsymbol{F}_{\mathrm{R}} - m_{\mathrm{h,R}}\left[\ddot{\nu}_{\mathrm{R},x}, 0, 0\right]^T$. The deflection angle θ is filtered using a lowpass Butterworth filter (order 8, passband edge frequency 10 rad/s) and subsequently numerically differentiated to obtain the deflection angle rate $\dot{\theta}$. The desired trajectory is computed from θ and $\dot{\theta}$ as described in Section III and fed as a desired velocity $\dot{\nu}_{\mathrm{R},x}$ to the impedance controlled manipulator. The low-level control of the manipulator as well as the proposed adaptive controller are implemented using *MATLAB/Simulink Real-Time Target*. Using this setup, the low-level control, communication and sensing loop is set to run at T =1 ms sampling time. The a and $\hat{\omega}$ adaptation loops, described in Section III, are set to run at sampling times T_a and T_ω, respectively. They are listed together with other parameters in Table I and Table II.

TABLE I: Pendulum and filter parameters

$m_{\mathrm{p,v}}$	$l_{\mathrm{R,v}}$	$m_{\mathrm{p,t}}$	$l_{\mathrm{R,t}}$	$l_{\mathrm{cyl,t}}$	ζ	c_0
0.5 kg	0.5 m	1.25 kg	0.55 m	0.85 m	1.2	0.9

TABLE II: LSPI for a and $\hat{\omega}_\theta$ adaptation

μ_a	N_a, M_a	ε_a	μ	T_a	${}^a\beta_\Gamma$	${}^a h_0$	${}^a\gamma$
200	12	131	4	0.142 s	0.01	0.1	1
${}^a\Delta u_Q$	${}^a K_x$	${}^a K_u$	${}^a K_{xu}$	μ_ω	ε_κ	κ	T_ω
600	0.01	0.1	20	50	23	25	0.1 s
${}^\omega\Delta u_Q$	N_ω, M_ω	${}^\omega\beta_\Gamma$	${}^\omega h_0$	${}^\omega\gamma$	${}^\omega K_x$		
400	11	0.3	0	0.9	4		

B. Experimental results

Initial tuning in simulation: The initial design phase of the proposed controller was performed in simulation using *MATLAB/Simulink*. A simple damped pendulum model was used to emulate the unknown suspended object. Tuning parameters of the proposed adaptive algorithm are given in Table II. The parameters tuned in simulation are the length of the polynomial state and action approximator $M_{a/\omega}$ and $N_{a/\omega}$, exploration phase $\varepsilon_{p/\kappa}$, the exploration decay $\mu_{a/\omega}$, the exploitation phase p and κ, the choice of the reward function ${}^{a/\omega}r$, states ${}^{a/\omega}x$ and the control action ${}^{a/\omega}u$. After initial tuning, performed in simulation, the proposed controller was implemented in the experimental setup. No additional tuning was performed during the experimental phase, even though we manipulated different kinds of objects. Since the proposed algorithm is discrete, the choice of sample times for the adaptive loops T_a and T_ω can be considered as additional parameters, which need to be tuned. Knowing the system model, this choice can be achieved using the Nyquist-Shannon sampling theorem, but in the model free setting, this can represent an additional challenge. In this paper with sampling times presented in Table II we achieve good experimental results. What we noticed during the initial tuning in simulation, is that if the sampling time is much smaller than the process reaction time, the proposed algorithm has problems with convergence.

Control of the v-pendulum using real-time adaptation: In the experimental video one can see how the experiment went. First, we started with a simple pendulum for which we set different desired deflection angles. The object mass $m_{\mathrm{p,v}}$ and the cable length $l_{\mathrm{R,v}}$ are given in the Table I. Learning parameters α_a and α_ω are initialized to zero and as the learning progresses the parameters get updated (Fig. 9). Also the initial policies were initialized to zero. From the deflection angle θ subplot we can see that the simple pendulum reached the desired deflection angle after 6 pendulum periods. Parameters learn online from the interaction with the real system. After the initial learning phase as a simple pendulum with different desired deflection angles θ_V, the human interaction partner pulled the rope resulting in the v-pendulum displayed in Fig. 5. As a consequence, the mass carried by the robotic manipulator is reduced and the projected length l^*_{R} decreased. Figure 6 shows the experimental results for the phase of changing dynamics. The simple pendulum is controlled to reach a deflection angle $\theta_V = 45^circ$. Due to the increase in velocity and deflection angle during swing-up, phase space errors $\Delta\mathrm{Circ}$ occur, which result in a peak in the adapted natural frequency estimate $\hat{\omega}_\theta$. After the desired deflection angle is reached, the pendulum changes from simple pendulum to v-pendulum, as visible in the increase in $F_{\mathrm{R},z}$. Due to the change in dynamics the maximum deflection drops slightly, but the reference is reached due to adaptation of the amplitude a. As expected, the natural frequency estimate $\hat{\omega}_\theta$ adapts to the frequency increase caused by the decreased projected length l^*_{R}. Note: Causes for the drop in the natural frequency estimate $\hat{\omega}_\theta$ at around 91 to 92 s could be the adaptation to the changed dynamics or exploration. The drop is visible in the phase space in form of a negative error $\Delta\mathrm{Circ}$.

Control of the t-pendulum using real-time adaptation: In the second experiment we use a t-pendulum. The experiment was performed in a similar fashion as the first one, i.e. we have the learning phase and the phase were the human operator injects disturbances into the system. The experimental results for the disturbance phase are given in Fig. 7. The disturbance which included additional energy injection, damping and the excitation of the unwanted oscillation mode of the t-pendulum are visible in the deflection angle θ. High velocities induced by aggressive maneuvers of the human operator are visible in phase space error $\Delta\mathrm{Circ}$ increase. The amplitude a is saturated while controlling the pendulum to reach the reference θ^{d}_V. The natural frequency estimate $\hat{\omega}_\theta$ captures the high frequency disturbance oscillations. These undesired ψ-oscillations are the result of the cylindrical mass rotating around its vertical axis. In this paper we do not try to explicitly damp the undesired oscillation. An in depth investigation for the need of disturbance suppression and the adaption of the learning towards this goal will be part of our

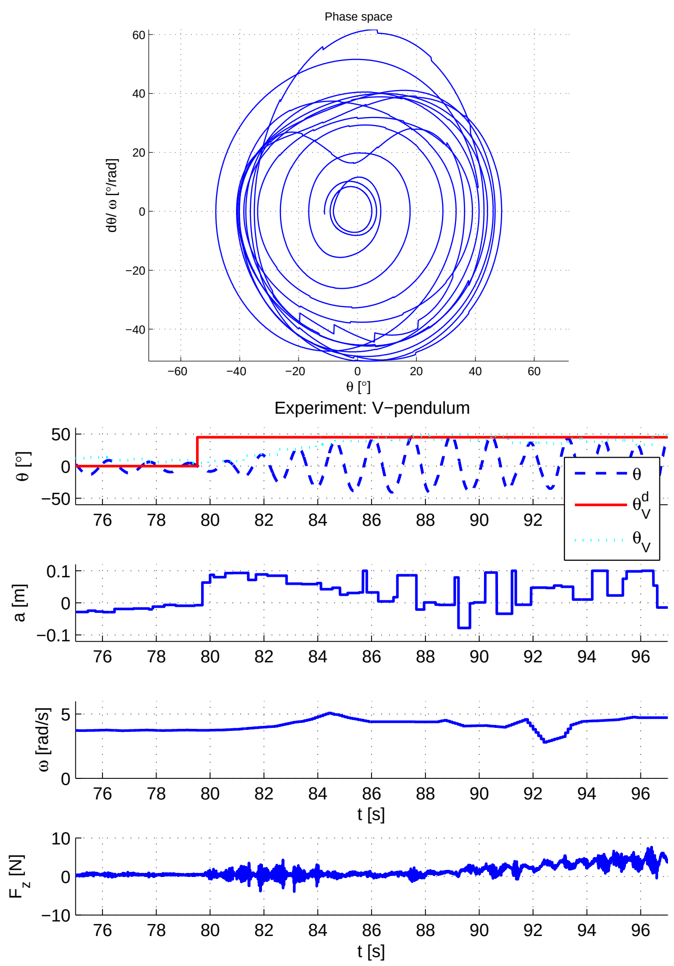


Fig. 6: V-pendulum: changing dynamics from a simple pendulum to a v-pendulum.

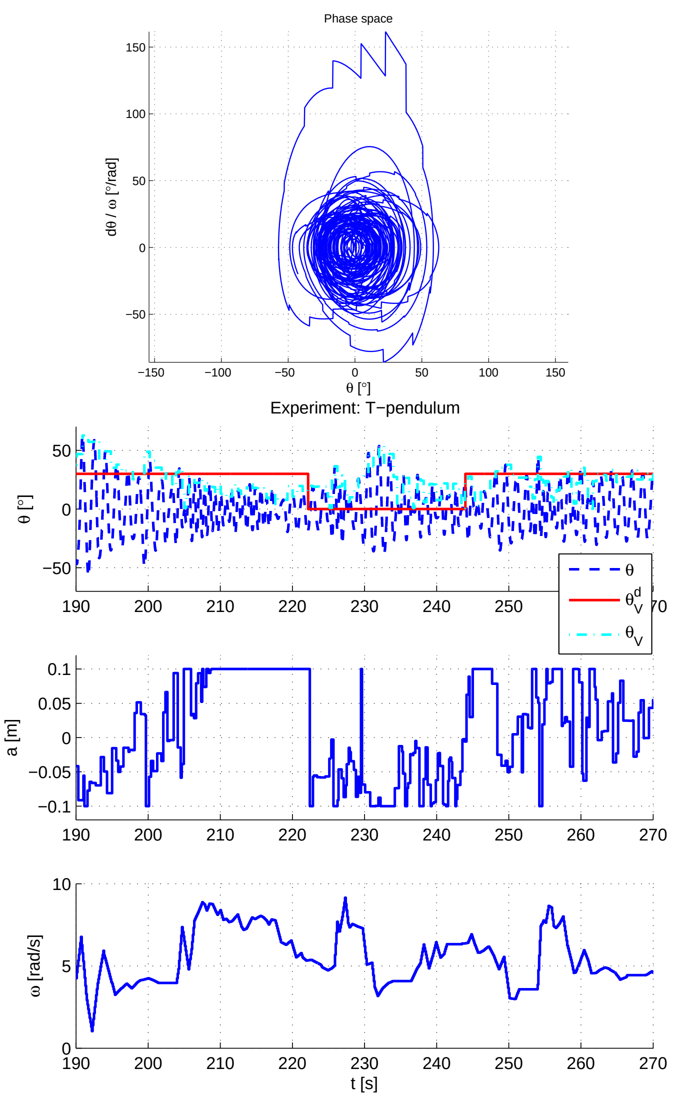


Fig. 7: T-pendulum: an external disturbance is introduced through a human operator.

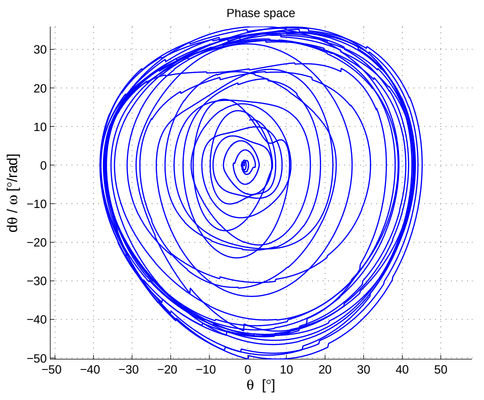
future work.

In Fig. 8 we show the adaptive pendulum swing-up control after a couple of swing-ups. The deflection angle reaches the desired deflection angle precisely. The adaptive parameter a behaves exactly as we would expect. While trying to reach the desired deflection angle and maintain it, the amplitude is positive which means that the controller is injecting energy into the system. Around 345 s the reference is set to zero, which means we want the controller to release the energy which is reflected in the negative a. The estimated natural frequency is constant and from the phase portrait we can see that it is close to the natural frequency of the t-pendulum.

Policy iteration algorithms converge to a fixed point representing the state and action function, Q-function. Since Q-function is approximated using linearly parametrized approximators (9), this means the algorithm converges to a fixed point in parameter space α_a and α_w. This is demonstrated on Fig. 9 representing the parameters α_a and α_w from previously described experiments with v- and t-pendulum. We can see that the LSPI for a adaptation converged to a fixed point considering the α_a parameters. As for the ω adaptation loop, the convergence trend is visible from the α_w figure, but there is room for improvement of this adaptation loop. One direction would be by improving the reward function ${}^{\omega}r$ and thus avoiding the use of (17) but use directly the control action ${}^{\omega}u_j$ for the $\hat{\omega}_{\theta,j}$. Alternatively, the use of the iterative law (17) could be considered if certain guarantees about the algorithm are provided.

V. CONCLUSIONS

This paper presents an adaptive control for cooperative dynamic object manipulation. The proposed controller combines fundamentals of pendulum excitation with online least square policy iteration to form an adaptive, suboptimal controller. The presented experiments show fast online learning enabling the robot to adapt to changes of the object dynamics as well as robustness to disturbances introduced by the human operator. After the initial design and parameter tuning in simulation, the algorithm did not need additional re-tuning after switching to experimental setup. From this we can conclude that the combination of chosen states, control actions, function approximators, and reward functions, during the



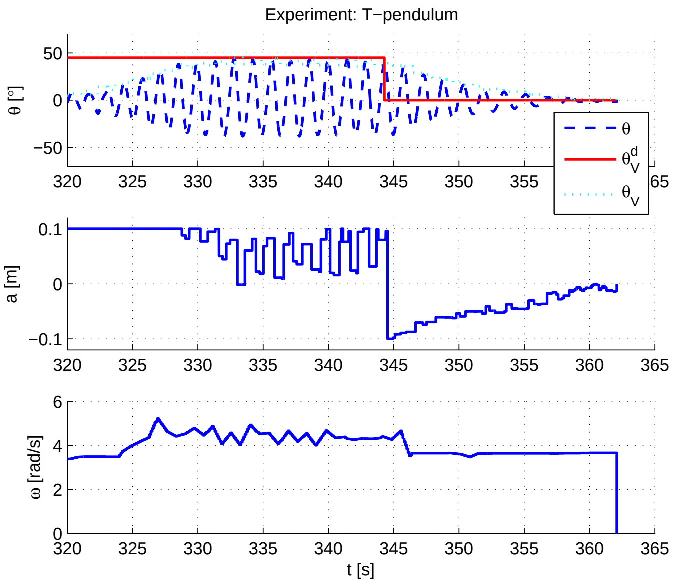


Fig. 8: T-pendulum: end phase of experiment.

initial design phase, extracts the main features for various suspended objects.

In future work, we plan to investigate the convergence properties of the proposed algorithm and how it relates to model based approaches. An in depth investigation of the undesired t-pendulum oscillations as well the the application of the proposed controller to different objects, like a rope, are future work. This approach brings us closer to the goal of cooperative flexible object swinging. However, as we focus in this paper on the robot leader with the human acting as a disturbance, a next step towards cooperative manipulation will be the design of an adaptive robotic follower controller.

ACKNOWLEDGMENT

This work is supported in part by the European Community's Seventh Framework Programme under grant No. 285939 (ACROSS) and in part within the ERC Advanced Grant SHRINE Agreement No. 267877 (`www.shrine-project.eu`).

REFERENCES

[1] K. Kosuge, H. Yoshida, and T. Fukuda, "Dynamic control for robot-human collaboration," in *Proc. 2nd IEEE Int. Workshop on Robot and Human Communication*, 1993, pp. 398–401.
[2] B. Corteville, E. Aertbelien, H. Bruyninckx, J. De Schutter, and H. Van Brussel, "Human-inspired robot assistant for fast point-to-point movements," in *Proc. IEEE ICRA*, 2007, pp. 3639–3644.
[3] A. Mörtl, M. Lawitzky, A. Küçükyılmaz, M. Sezgin, C. Basdogan, and S. Hirche, "The Role of Roles: Physical Cooperation between Humans and Robots," *Int. Journal of Robotics Research*, vol. 31, no. 13, pp. 1656–1674, 2012.
[4] N. Zoso and C. Gosselin, "Point-to-point motion planning of a parallel 3-dof underactuated cable-suspended robot," in *Proc. IEEE ICRA*, may 2012, pp. 2325–2330.
[5] J. Nakanishi, T. Fukuda, and D. Koditschek, "A brachiating robot controller," *IEEE Trans. on Robotics and Automation*, vol. 16, no. 2, pp. 109–123, 2000.
[6] Y. Maeda, A. Takahashi, T. Hara, and T. Arai, "Human-robot cooperation with mechanical interaction based on rhythm entrainment-realization of cooperative rope turning," in *Proc. IEEE ICRA*, vol. 4, 2001, pp. 3477–3482 vol.4.
[7] C. Kim, K. Yonekura, H. Tsujino, and S. Sugano, "Physical control of the rotation center of an unsupported object rope turning by a humanoid robot," in *Proc. 9th IEEE-RAS Int. Conf. on Humanoid Robots*, 2009, pp. 148–153.
[8] P. Donner, F. Christange, and M. Buss, "Human-robot cooperative swinging of complex pendulum-like objects," in *Proc. IEEE/RSJ IROS*. IEEE, 2013, pp. 4602–4608.
[9] P. Donner, A. Mörtl, S. Hirche, and M. Buss, "Human-robot cooperative object swinging," in *Proc. IEEE ICRA*. IEEE, 2013, pp. 4328–4334.
[10] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.
[11] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *Journal of Machine Learning Research*, vol. 4, pp. 1107–1149, December 2003.
[12] L. Buşoniu, R. Babuška, B. De Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, Florida: CRC Press, 2010.
[13] I. Palunko, A. Faust, P. Cruz, L. Tapia, and R. Fierro, "A reinforcement learning approach towards autonomous suspended load manipulation using aerial robots," in *IEEE International Conference on Robotics and Automation*, May 2013, pp. 4896–4901.
[14] A. Faust, I. Palunko, P. Cruz, R. Feirro, and L. Tapia, "Learning swing-free trajectories for uavs with a suspended load," in *IEEE International Conference on Robotics and Automation*, May 2013, pp. 4902–4909.
[15] K. Yoshida, "Swing-up control of an inverted pendulum by energy-based methods," in *Proc. ACC*, vol. 6, 1999, pp. 4045–4047 vol.6.
[16] R. E. Bellman, *Dynamic Programming*. Princeton University Press, New Jersey, 1957.
[17] "Institute for information-oriented control (itr)," http://www.itr.ei.tum.de/, accessed: 2014-02-06.

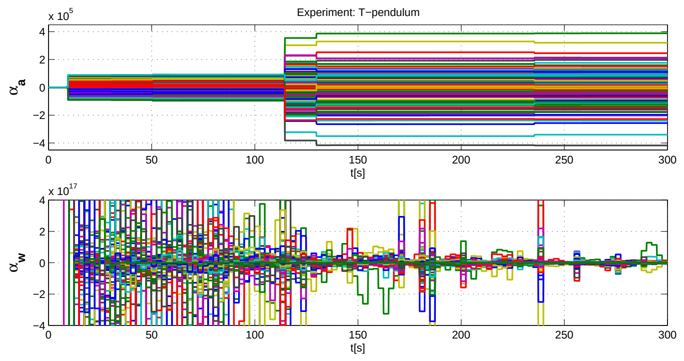


Fig. 9: Learned parameters of the Q-function approximators for the T- pendulum experiment.