

Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty

Seyed Sina Mirrazavi Salehian, Nadia Figueroa and Aude Billard
LASA Laboratory, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland
Email: {sina.mirrazavi, nadia.figueroafernandez, aude.billard@epfl.ch}

Abstract—Coordinated control strategies for multi-robot systems are necessary for tasks that cannot be executed by a single robot. This encompasses tasks where the workspace of the robot is too small or where the load is too heavy for one robot to handle. Using multiple robots makes the task feasible by extending the workspace and/or increase the payload of the overall robotic system. In this paper, we consider two instances of such tasks: a co-worker scenario in which a human hands over a large object to a robot; intercepting a large flying object. The problem is made difficult as the pick-up/intercept motions must take place while the object is in motion and because the object’s motion is not deterministic. The challenge is then to adapt the motion of the robotic arms in coordination with one another and with the object. Determining the pick-up/intercept point is done by taking into account the workspace of the multi-arm system. The point is continuously recomputed to adapt to change in the object’s trajectory. We propose a virtual object based dynamical systems (DS) control law to generate autonomous and synchronized motions for a multi-arm robot system. We show theoretically that the multi-arm + virtual object system converges asymptotically to the moving object. We validate our approach on a dual-arm robotic system and demonstrate that it can re-synchronize and adapt the motion of each arm in a fraction of a second, even when the motion of the object is fast and not accurately predictable.

I. INTRODUCTION

Many daily activities involve tasks such as lifting, carrying and reaching for large and heavy objects. To accomplish these tasks, humans rely heavily on bi-manual reaching and, more generally, on coordination between the hands [6]. Performing these tasks with one arm is often infeasible, mainly because a single arm has a limited workspace. Moreover, the dexterity and flexibility required for such task is beyond a single arm’s capabilities. This holds for robotic systems as well. A single robot arm is simply incapable of meeting the requirements of such complex tasks. A dual or multi-arm robotic system, on the other hand, extends the workspace of a single robot arm. It allows for highly complex manipulation of heavy or large objects that would otherwise be infeasible for single-arm systems. Most effort in the field of multi-arm control has focused primarily on devising strategies for coordinated and stable manipulation of *static* objects that are already partially or fully grasped by the multi-arm system [1, 23, 5, 21, 33, 7]. Seldom work has focused on developing *reaching strategies* that a multi-arm system can use to reach and grab *moving* objects.

One can envision a plethora of applications in factories, airports, or storage facilities that would benefit from such

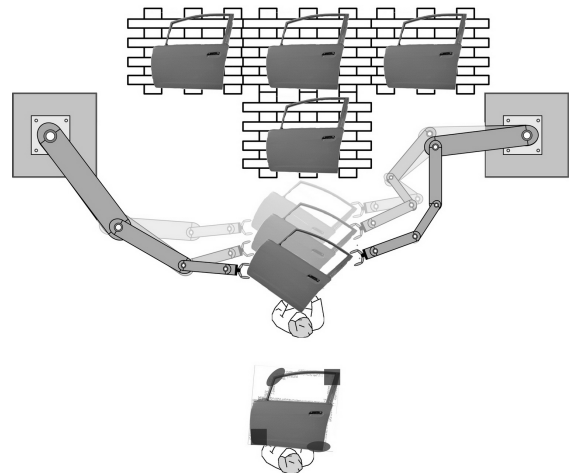


Fig. 1: Illustration of a possible application where a multi-arm system has to reach for large moving objects carried to them by humans. The arms move in synchrony towards the coupled feasible reaching points of the object (e.g. squares or ellipses). As the object approaches, the arms’ end-effectors align their trajectories with that of the object.

strategies. Examples include, grabbing, catching, carrying, lifting or re-orienting packages or large-sized parts traveling on a cart or a running conveyor belt (Fig. 1), carried by humans or even flying towards the multi-arm robot system. We posit that, until now, these applications have not yet been explored, due to the unsuitability of the state-of-the-art to tackle the challenges of coordinating the motion of multiple arms while reaching and adapting to the motion of a dynamic object in a computationally efficient way.

According to studies in motor and cognitive development, reaching for an object in a smooth and efficient manner requires ones to deal simultaneously with different issues [27]. Each hand has to adjust to the orientation, shape and size of the object while reaching for it. Moreover, the action of grasping the object (i.e. closing the fingers of the hands) must be timed prior to rather than as a reaction to intercepting the object. Hence, bi-manual, and by extension, multi-manual reaching require us to solve simultaneously several spatial and temporal coordination constraints to move toward the object in coordination and to intercept the object [27].

We propose an approach that generates coordinated trajectories for a multi-arm robot system that ensures that the arms will reach the moving object simultaneously. Most importantly, the approach updates the arms’ motion continuously and in synchrony to adapt to changes in the target object’s trajectory. To validate the approach, we consider a scenario in which

an object with arbitrary shape and mass is moving towards a multi-arm system. We do not assume a known model of the dynamics of the object. To simulate an arbitrary object's motion, not easily predictable, our experiments have a human carry the object toward the robot while being blindfolded. The sole knowledge about the object is its *coupled feasible reaching points*, which are the preferred reaching positions and orientations on the object, specified by the user (see Fig. 1). A multi-arm *reach* is thus deemed successful if and only if all robot arms simultaneously intercept the object on its feasible reaching points. To achieve this, we must solve for the aforementioned constraints that constitute a multi-arm reach. From a robotics point of view, these constraints can be translated to object level constraints (i.e. *task constraints*) and arm level constraints (i.e. *coordination constraints*).

Task constraints impose position and velocity constraints at the object's interception. Position constraints ensure that the planned motion of each robot's end-effector is coordinated with the feasible reaching positions of the object. Whereas the velocity constraints allow readjustments of the hand, palm and fingers posture while there are uncertainties in the hand orientation and position. On the other hand, *coordination constraints* impose that the robots move in coordination with each other. This is necessary not only to ensure that the arms simultaneously intercept the object, but also to avoid collisions between their end-effectors while they adapt to the moving object's motion. Handling multiple constraints simultaneously is a problem that could be addressed by using standard optimal control approaches. These, however, are time consuming and will not converge within a few milliseconds, which is the expected reaction time necessary for all arms to rapidly and simultaneously reach and adapt to the moving object.

In this paper, we use autonomous dynamical systems (DSs) to instantaneously re-plan the coordinated motion for each K_r -th hand-arm robot in a multi-arm system [20, 11, 18]. The proposed dynamical system couples both *task* and *coordination constraints* by modeling the motion of a *virtual object*, which is expressed as a Linear Parameter Varying (LPV) system subject to stability constraints to ensure that the object is simultaneously intercepted by all the arms at the desired *time*, *position* and *velocity*. A theoretical analysis of the stability and convergence of this proposed DS based control law is presented and empirically validated on a real-world dual-arm experiment with a KUKA IIWA and a KUKA LWR 4+, both 7 degree of freedom robot arms mounted with a 16 DOF (Degree of Freedom) Allegro hand and a 4 DOF Barret hand, respectively.

II. RELATED WORK

Although coordinating multiple robotic arms for object manipulation has been extensively studied in robotics literature, planning the reach-to-grasp motion for a moving object with a multi-arm robotic system, while keeping coordination constraints, is a new field of research. Recently, Vahrenkamp et al. [25, 24] proposed an RRT-based algorithm to generate collision free motions to grasp an object at rest with

a bi-manual platform. Given its search-based strategy, this approach can guarantee feasible grasps by both arms and self collision avoidance. However, due to its computational complexity and the fact that it cannot guarantee simultaneous interception of the object by all arms, it becomes inadequate when trying to reach for a *moving* object. To the best of the author's knowledge, there is no related work in the field of *reaching* for large *moving* object with multi-arm systems.

Extensive surveys on coordination strategies for dual/multi-arm systems are presented in [21, 31]. Early approaches addressed the coordination problem with master-slave strategies. In the work of [16], the motion of the master robot is planned and always known, whereas the slave robot must follow the master's motion while conforming with closed-chain geometrical constraints. On the same track, Gams et al. [9] proposed a control architecture that addresses the task of lifting an unknown object with a dual-arm system. It uses feedback errors constructed from position and velocity components to synchronize the motion of one arm to a predefined motion of the other arm. Even though this strategy is computationally efficient, performance is adversely affected when communication delays arise. Moreover, when dealing with moving objects, a difficulty arises when the master and slave roles have to be assigned to the arms.

Centralized controller strategies can address some of these issues [1, 22, 28]. These strategies consider the robots and the manipulated object as a closed kinematic chain. In this line, Wimböck and Ott [32] proposed an impedance control architecture for dual-arm manipulation, where the two end-effectors and a virtual frame, which is a function of the end-effectors' poses, are coupled via spatial springs. Zhu [34] proposed a motion synchronization controller to coordinate the end-effectors of two robots when they are rigidly or flexibly holding an object without payload. Likar et al. [14] proposed a velocity level motion synchronization algorithm for controlling a cooperative dual-arm system. They introduced an augmented kinematic chain which is a representation of two arms and the object. The corresponding Jacobian is calculated to control the augmented kinematic chain by solving the inverse kinematics problem at the velocity level. On the other hand, some researchers have proposed decentralized control architectures, where the robots are controlled separately by their own local controllers [15]. Caccavale et al. [4] proposed a combination of a centralized and decentralized impedance control strategies to achieve a desired impedance at the object and the end-effector level, respectively.

All previously mentioned works assume that the object is firmly attached to the robots and modeled via a virtual object frame or by closing the kinematic chain. In this work, we adopt the idea of the *virtual object* to coordinate the motion of the robots with each other and further, use it to coordinate the robots to the real object, albeit it is not attached to the arms until they reach it. The term *virtual object* is mostly used in robotics literature to represent the internal forces of a grasping task [29, 30]. In this paper, however, this term is used in the framework of motion coordination. The motion of the *virtual*

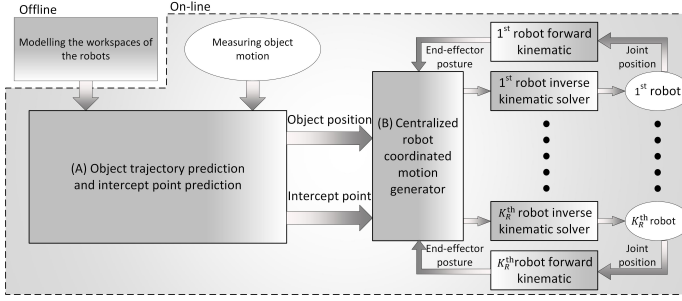


Fig. 2: Block diagram for coordinated multi-arm motion planning for reaching a large moving object. Where K_r represents the total number of robot arms. In this paper, we assume that the low-level controller of the robot is a perfect tracking controller.

object is coordinated and aligned to the motion of the real object to satisfy the *task constraints*. Concurrently, it is used to *coordinate* the motion of the robots through a centralized controller, which is governed by the *virtual object's* motion and robust to perturbations from any of the K_r -th robot arms.

III. MULTI-ARM REACHING FOR MOVING OBJECTS

In order to achieve stable and coordinated reaching motions for multiple arms when the target object is in motion, several problems need to be solved simultaneously: (i) prediction of the object's trajectory; (ii) computing intercept points for each arm and (iii) planning coordinated motion of the robot arms towards their corresponding intercept points. An overview of the proposed framework is illustrated in Fig. 2. As seen on the illustration, sub-component (A) computes the intercept point and plans the trajectory to the intercept point. It uses an estimate of the reachable workspace of the multi-arm system (learned off-line prior to experiment) and an on-line step in which it continuously measures the object's pose from a visual tracking system. Sub-component (B) uses the intercept points predicted from (A) and the current end-effector poses to generate the desired end-effector poses of the K_r -robot multi-arm system. It is based on a centralized controller that exploits a forward model of the virtual object's motion. A detailed description of each of these sub-components is presented in the following subsections. For simplicity and practicality, we summarize the most relevant notation used throughout the paper in Table I and illustrate them in Fig.3.

A. Object Trajectory and Intercept Point Prediction

As stated previously, the proposed control architecture coordinates and aligns the motion of a *virtual object* attached to the end-effectors of the K_r robots to the motion of a real object (Fig.3). The real object is represented by a set of *feasible reaching points* ($\xi_{p1}^O, \xi_{p2}^O \in \mathbb{R}^3$), defined by the user. Their resultant vector ($\xi_p^O(t) \in \mathbb{R}^3$) corresponds to the position of the object. Its orientation ($\xi_o^O(t) \in SO(3)$) is estimated from the positions of the reaching points. The *virtual object* is a duplicate of these set of points, with the end-effectors of each robot attached to the virtual reaching points ($\xi_{p1}^V, \xi_{p2}^V \in \mathbb{R}^3$) by zero length springs and dampers (Fig.3). Once the real object starts moving towards the robots, carried or thrown by an operator, the virtual object follows its motion. A forward

TABLE I: Notation

Variable	Definition
ε	A small positive number.
K	A large constant positive number.
K_r	Number of robot arms.
ξ_j^R	Pose of the j^{th} end-effector for $\forall j \in 1, \dots, K_r$
ξ_{pj}^V	j^{th} reaching point (pose) on the <i>virtual object</i> .
ξ_p^V, ξ_o^V	Position and orientation of the <i>virtual object</i>
ξ_p^O, ξ_o^O	Position/orientation of the real object.
ω	Angular velocity of the virtual object
ξ_{pj}^O	j^{th} feasible reaching point on the real object.
$j\xi$	ξ in the reference frame of j^{th} robot base.
θ^W	Set of K_r workspace models $\theta^W = \{\theta_i^W \forall i \in 1, \dots, K_r\}$
θ_i^W	GMM parameters for each workspace $\theta_i^W = \{\pi_{li}, \mu_{li}, \Sigma_{li}\}$
$\pi_{li}, \mu_{li}, \Sigma_{li}$	Priors, means and covariance matrices for GMM with $l = 1 \dots K_i^W$ Gaussian functions.
$x_p^V(t), x_o^V(t)$	States of the virtual object's dynamical system.
x_j^R	States of j^{th} end-effector .
$\theta_{A_p}, \theta_{A_o}$	Scheduling parameters for pos./orient. dynamics.
$A_i(\theta_{A_i})$	Affine dependent state-space matrices for $\forall i \in \{p, o\}$.
$u_p(t), u_o(t)$	Control input for the pos./orient. virtual object dynamics.
$\gamma_p(t), \gamma_o(t)$	Coordination parameters for the virtual object dynamics.

model of the virtual object computes its progress ahead of time and determines a point along this trajectory where the object will become reachable by all robot arms. We name this point the *feasible intercept point*. Using the planned position and orientation of the virtual object at intercept point ($\xi_p^V(t) \in \mathbb{R}^3$, $\xi_o^V(t) \in SO(3)$), we can then generate the desired reaching motions for each arm.

If the motion of the object is predictable, one can use a model-based prediction approach [13] and find feasible intercept postures by extending a single-robot arm feasible posture extraction algorithm (such as [12]) to K_r -robot arms. In this case, a simple point-to-point motion can be devised to intercept the object. As we assume that the motion of the object is not accurately predictable (e.g when carried by a blind-folded human operator as in our experiments); using a model-based approach would be impractical and limiting. Instead, we use a simple ballistic motion algorithm to predict the motion of the real object and propose a controller (Sec. III-B) that is compliant and robust to prediction inaccuracies.

To model the motion of the *virtual object*, we must first make sure that the object travels through the workspace of the robots; i.e. there is at least one feasible intercept point. In order to find the feasible intercept point, the kinematic feasibility of the predicted reaching points must be evaluated. The reachable workspace of each robot is modeled via a probabilistic classification model $p_j(j\xi_{pj}^O; \theta_j^W) \forall j \in \{1, \dots, K_r\}$, namely a Gaussian Mixture Model as follows:

$$p_j(j\xi_{pj}^O; \theta_j^W) = \sum_{l=1}^{K_j^W} \pi_{lj} \mathcal{N}(j\xi_{pj}^O | \mu_{lj}, \Sigma_{lj}) \quad (1)$$

where $\pi_{lj}, \mu_{lj}, \Sigma_{lj}$ correspond to the prior, mean and covariance matrix of the $l = \{1 \dots K_j^W\}$ Gaussian functions, respectively, estimated by using the Expectation-Maximization algorithm [3]. In order to generate the training dataset, all possible postures of each robot are simulated by testing all possible displacements of their joints. δ_j is the *minimum likelihood* threshold and it is determined such that the likelihood

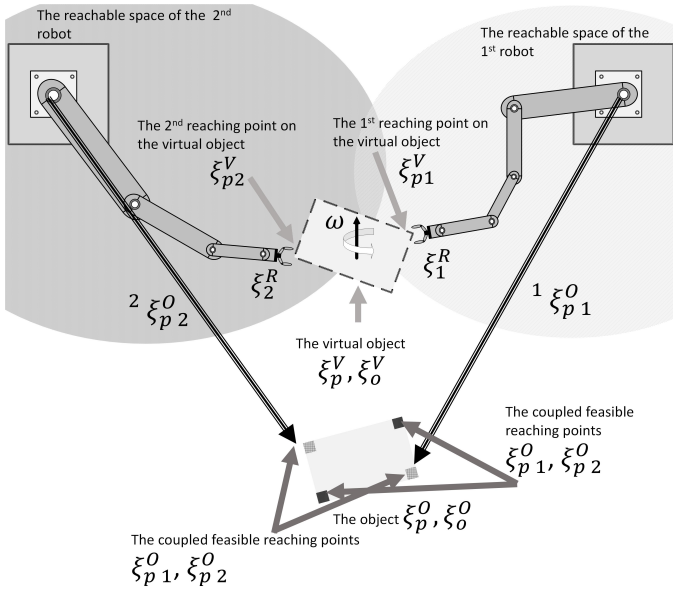


Fig. 3: An illustration of the variables defined in the paper is presented. The reachable areas are feasible areas for grasping the object. Except for ${}^2\xi_{p2}^O$ and ${}^1\xi_{p1}^O$, the variables are expressed in the reference frame located on the desired intercept point; i.e. $\xi_i^O(T^*) = [0 \dots 0]^T \forall i \in \{p, o\}$. Note that ω is the angular velocity of the virtual object, which is different from the numerical differentiation of the virtual object orientation; i.e. $\omega \neq \dot{\xi}_o^V$.

of 99% of the training points is higher than the threshold δ_j . If $p_j(j\xi_{pj}^O; \theta_j^W)$ exceeds it, $j\xi_{pj}^O$ is classified as a feasible configuration. As the reachable workspace of each robot is statistically independent from the others, we can calculate the joint distribution of all the models by computing the product of all distributions:

$$p({}^1\xi_{p1}^O, \dots, {}^{K_r}\xi_{pK_r}^O; \theta^W) = \prod_{j=1}^{K_r} p_i(j\xi_{pj}^O; \theta_j^W) \quad (2)$$

Similarly the *minimum joint likelihood threshold* is $\delta = \prod_{j=1}^{K_r} \delta_j$ and if $p({}^1\xi_{p1}^O(T^*), \dots, {}^{K_r}\xi_{pK_r}^O(T^*); \theta^W)$ exceeds it, $(\xi_p^O(T^*) = \frac{1}{K_r} \sum_{j=1}^{K_r} j\xi_{pj}^O(T^*))$ is classified as the feasible intercept point. As the motion of the object is not predicted accurately, there is no guarantee that the object will pass through the predicted point at T^* , yet we can ensure that the object is moving toward the robots.

B. Centralized Robot Coordinated Motion Generator

To model the position and the orientation of the virtual object, we consider the following class of fully observable continuous-time LPV based DSs [26]. LPV systems can be considered as a non-linear combination of linear systems which are valid at particular operating points. Formulating the motion of the virtual object as a LPV system allows modeling complex and nonlinear behaviors and the use of many tools from the linear systems theory for analysis and control [8].¹

$$\dot{x}_p^V(t) = A_p(\theta_{A_p}(x_p^V(t)))x_p^V(t) + u_p(t) \quad (3a)$$

¹As the states of the virtual object are fully observable, $y_p(t) = x_p^V(t)$ and $y_o(t) = x_o^V(t)$ where $y_i(t) \forall i \in \{p, o\}$ are the outputs.

$$\dot{x}_o^V(t) = A_o(\theta_{A_o}(x_o^V(t)))x_o^V(t) + u_o(t) \quad (3b)$$

Where $x_p^V(t) = [\xi_p^V(t) \ \dot{\xi}_p^V(t)]^T$ and $x_o^V(t) = \xi_o^V(t)$ are the states of the dynamical system.² $\theta_{A_i} \in \mathbb{R}^{K_i \times 1} \forall i \in \{p, o\}$ are vectors of scheduling parameters³;

$$\theta_{A_i} = [\theta_{A_{i1}} \ \dots \ \theta_{A_{iK_i}}]^T \quad \forall i \in \{p, o\} \quad (4)$$

$A_i(\cdot) : \mathbb{R}^{K_i \times 1} \rightarrow \mathbb{R}^{j \times j} \forall (i, j) \in \{(p, 6), (o, D)\}$ are the affine dependence of state-space matrices on the scheduling parameter and the state vectors:

$$A_p(\theta_{A_p}) = \sum_{k=1}^{K_p} \theta_{A_{pk}} A_{pk} \quad A_{pk} \in \mathbb{R}^{6 \times 6} \quad \theta_{A_{pk}} \in \mathbb{R}^{1 \times 1} \quad (5)$$

$$A_o(\theta_{A_o}) = \sum_{k=1}^{K_o} \theta_{A_{ok}} A_{ok} \quad A_{ok} \in \mathbb{R}^{D \times D} \quad \theta_{A_{ok}} \in \mathbb{R}^{1 \times 1}$$

One popular choice to approximate the parameters of LPV systems is regression models; i.e. polynomial, periodic functions [2] or Gaussian Mixture Regression (GMR) [19]. In this paper, an approach similar to that offered in [19] is taken, which inherently results in a normalized scheduling of parameters; i.e. $0 < \theta_{A_{ij}} \leq 1$, $\sum_{k=1}^{K_i} \theta_{A_{ik}} = 1 \forall (i, j) \in (p, 1), \dots, (p, K_p), (o, 1), \dots, (o, K_o)$.

As discussed in Section III-A, the virtual object and consequently the arms must simultaneously intercept the object at the feasible reachable points. To achieve this, we define the following control inputs (u_p, u_o) for the position and orientation of the virtual object:

$$u_p(t) = \frac{1}{K_r + 1} \left(\dot{x}_{\gamma p}^O(t) - A_p(\theta_{A_p})x_{\gamma p}^O(t) + \sum_{j=1}^{K_r} (U_j - \omega \star R_j) \right) - \frac{K_r}{K_r + 1} A_p(\theta_{A_p})x_p^V(t) \quad (6a)$$

$$u_o(t) = \gamma_o(t)\dot{x}_o^O(t) + \dot{\gamma}_o(t)x_o^O(t) - \gamma_o(t)A_o(\theta_{A_o})x_o^O(t) \quad (6b)$$

Where $x_{\gamma p}^O(t) = [\gamma_p(t)\xi_p^O(t) \ \gamma_p(t)\dot{\xi}_p^O(t) + \dot{\gamma}_p(t)\xi_p^O(t)]^T$ ⁴ and $x_o^O(t) = \xi_o^O(t)$. The origin is located on the desired intercept point $\xi_o^O(T^*) = [0 \dots 0]^T \forall i \in \{p, o\}$. $\dot{x}_o^O(t)$ is the numerical differentiation of $x_o^O(t)$. $\omega \star R_j = [\omega \times r_j \ \omega \times (\omega \times r_j)]^T$, where $r_j = \xi_{pj}^V(t) - \xi_p^V(t)$. $0 < \gamma_i(t) < 1 \forall i \in \{p, o\}$ are the *coordination* parameters and are of class C^1 . U_j is the interaction effect of the tracking controller of the j^{th} end-effector on the virtual object:

$$U_j = \dot{x}_j^R(t) + A_j^R(x_{pj}^V(t) - x_j^R(t)) \quad (7)$$

²The dynamics of the position and orientation of the virtual object are presented in the acceleration and velocity level due to implementation constraints. However, the framework is not theoretically constrained to this.

³The scheduling parameters can be a function of time t , the states of the system $x_i^V(t) \forall i \in \{p, o\}$ or external signals $d(t)$, i.e. $\theta_{A_i}(t, \xi(t), d(t))$. In the rest of the paper, we assume that it is only function of the states of the system $x_i^V(t) \forall i \in \{p, o\}$ and the arguments are dropped for simplicity.

⁴ $\dot{x}_{\gamma p}^O(t)$ is the derivative of $x_{\gamma p}^O(t)$ with respect to time; $\dot{x}_{\gamma p}^O(t) = [\gamma_p(t)\dot{\xi}_p^O(t) + \dot{\gamma}_p(t)\xi_p^O(t) \ \gamma_p(t)\ddot{\xi}_p^O(t) + 2\dot{\gamma}_p(t)\dot{\xi}_p^O(t) + \ddot{\gamma}_p(t)\xi_p^O(t)]^T$

Where $A_j^R \in \mathbb{R}^{6 \times 6}$ is a constant matrix. The desired motion of the j^{th} end-effector ($x_j^R(t)$) is calculated based on the tracking error between the position of j^{th} point on the virtual object ($x_{pj}^V(t)$) and the end-effector:

$$\dot{x}_j^R(t) = \dot{x}_{pj}^V(t) - A_j^R(x_{pj}^V(t) - x_j^R(t)) \quad (8)$$

By substituting, (6a) and (7) into (3a) and (6b) into (3b), we have:

$$\begin{aligned} \dot{x}_p^V(t) = & \frac{1}{K_r + 1} \left(\dot{x}_{\gamma p}^O + A_p(\theta_{A_p})(x_p^V(t) - x_{\gamma p}^O(t)) \right. \\ & \left. + \sum_{j=1}^{K_r} (\dot{x}_j^R(t) + A_i^R(x_{pj}^V(t) - x_j^R(t)) - \omega \star R_j) \right) \quad (9a) \end{aligned}$$

$$\dot{x}_o^V(t) = \gamma_o(t)\dot{x}_o^O + \dot{\gamma}_o(t)x_o^O(t) + A_o(\theta_{A_o})(x_o^V(t) - \gamma(t)x_o^O) \quad (9b)$$

Theorem 1: The dynamical systems given by (9a) and (9b) asymptotically converge to $[\gamma_i(t)\xi_i^O(t) \quad \gamma_i(t)\dot{\xi}_i^O(t) + \dot{\gamma}_i(t)\xi_i^O(t)]^T \forall i \in \{p, o\}$ and the j^{th} end-effector asymptotically converges to the j^{th} reaching area on the virtual object i.e.

$$\lim_{t \rightarrow \infty} \|x_j^R(t) - x_{pj}^V(t)\| = 0 \quad (10)$$

$$\lim_{t \rightarrow \infty} \|\xi_i^V(t) - \gamma_i(t)\xi_i^O(t)\| = 0 \quad (11)$$

$$\lim_{t \rightarrow \infty} \|\dot{\xi}_i^V(t) - (\gamma_i(t)\dot{\xi}_i^O(t) + \dot{\gamma}_i(t)\xi_i^O(t))\| = 0 \quad (12)$$

if there are P_i^V , P_j^R , Q_i^V , Q_j^R such that:

$$\begin{cases} 0 \prec P_i^V \quad 0 \prec P_j^R & 0 \prec Q_i^V \quad 0 \prec Q_j^R \\ P_i^V A_{ik} + A_{ik}^T P_i^V \prec -Q_i^V & \forall k \in \{1, \dots, K_i\} \\ P_j^R A_j^R + A_j^{R^T} P_j^R \prec -Q_j^R & \forall j \in \{1, \dots, K_r\} \\ 0 \leq \theta_{A_{ik}} \leq 1, & \forall i \in \{p, o\} \end{cases} \quad (13)$$

where $0 \prec$ and $\prec 0$ refer to positive and negative definiteness of a matrix, respectively.

Proof: see Appendix A.⁵

If we set the *coordination* parameters $\gamma_i(t) = \dot{\gamma}_i(t) = 0 \forall i \in \{p, o\}$, (9) generates asymptotically stable motions towards the predicted intercept point: i.e. the coordination between the robots is preserved, but the coordination between the robots and the object is lost. On the other hand, if $\gamma_i(t) = 1$ and $\dot{\gamma}_i(t) = 0 \forall i \in \{p, o\}$, (9) will generate asymptotically stable motions towards the real object even though its motion is not accurately predicted: i.e. perfect *coordination* with the object.⁶ However, in this case, there is no guarantee that the virtual object converges to the real object by following a path which is kinematically feasible for all arms, i.e. due to the robots constraints, the coordination between the robots is lost. Thus, one can vary the values of the coordination parameters

⁵Due to space limitation, only the proof for the case $i = p$ is presented. Stability and convergence proof for $i = o$ is similar to $i = p$.

⁶We assume that the dynamical system (9) is fast enough to converge to an acceptable neighborhood around the desired trajectory $\gamma[\xi_i^O(t) \quad \dot{\xi}_i^O(t)]^T$ before T^* ; i.e. $\|\xi_i^V(T^*) - \gamma\xi_i^O(T^*)\| \leq \varepsilon \forall i \in \{p, o\}$

between zero and one such that they are one at the vicinity of the predicted intercept time; e.g.:

$$\dot{\gamma}_p = \frac{1 - \gamma_p}{\|\xi_p^O(t) - \xi_p^O(T^*) + \varepsilon\|} = \frac{1 - \gamma_p}{\|\xi_p^O(t) + \varepsilon\|} \quad (14a)$$

$$\gamma_o = \frac{1}{1 + e^{-K(p(\xi_{p1}^O(t), \dots, K_r \xi_{pK_r}^O(t); \theta^W)) - \delta}} \quad (14b)$$

(14a) improves the robustness of the multi-arm reaching motion in face of inaccuracies in the object's motion prediction as it ensures that when the object is close enough to the feasible reaching positions, the virtual object converges to the real object and perfectly tracks it; i.e. $\gamma_i(T^*) = 1 \forall i \in \{p, o\}$. Hence, the robots can simultaneously track the desired reaching points on the object in coordination. (14b) states that the orientation of the virtual object tracks the orientation of the object when the reaching points of the object are in the workspace of the corresponding robot. (14b) is advantageous in two ways. First, when the object is far from the robots, the virtual object does not rotate around itself hence $\omega \approx 0$. This significantly simplifies (9a). Second, if there are more than one coupled reaching points, (14b) is advantageous in that each robot only converges to the object by intercepting it from the corresponding feasible reaching point, avoiding jumps in the end-effector from one reaching point to another.

IV. EMPIRICAL VALIDATION

A. Experimental Setup

The proposed framework is implemented on a real dual-arm platform, consisting of two 7 DOF robotic arms, namely a KUKA LWR 4+ and a KUKA IIWA mounted with a 4 DOF Barrett hand and a 16 DOF Allegro hand. The distance between the base of the two robots is $[0.25 \quad 1.5 \quad -0.1]^T m$. The robot implementation involves converting the output of the dynamical system (8) (e.g. the desired motion of each K_r -th end-effector) into a 7-DOF joint state (for each arm, see Fig. 2) using a velocity based control method without joint velocity integration [17]. In order to avoid high torques, the resulting joint angles are filtered by a critically damped filter. The robot is controlled at a rate of 500 Hz. The fingers are controlled with joint position controllers. All the hardware involved (e.g. arms and hands) are connected to and controlled by one 3.4-GHz i7 PC. The position of the *feasible reaching points* of the objects are captured by an Optitrack motion capture system from Natural point at 240 Hz. Since the control loop is faster than the motion capture system, the predicted position of the object is used as the object position in (6), when the current position of the object is not available.

Our empirical validation is divided into three parts that demonstrate the controller's ability: (i) coordinate the multi-arm systems; (ii) adapt the two arms' motions in coordination so as to reach to grab a large moving object, when introducing unpredictability in the object's motion (by having the object be carried by a blindfolded human) and (iii) rapidly adapt bi-manual coordination to intercept a flying object, without using a pre-defined model of the object's dynamics. A corresponding video is available on-line: <https://youtu.be/UfucwRGa7k8>

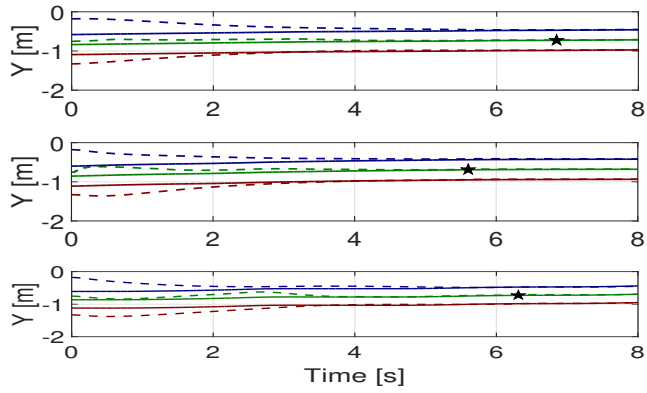


Fig. 6: Different examples of the position of the end-effectors generated by the dynamical system (9a). Due to the space limit, only the trajectories along y axis is presented. The illustrated object trajectory is the predicted trajectory of the uncaught object. The prediction of the box’s trajectory requires some data to be initialized and uses almost all of the first 0.2 meter of the object in x. The initial value of γ_p is 0.0 and (14) is used to change it with respect to the distance of the object to the robots. As expected, the outputs of (9a) first converges to desired intercept position, since γ_p is a small value, then it softly intercepts the object’s trajectory and follow the object’s motion. The robots are stopped if the object is not moving or the fingers are closed.

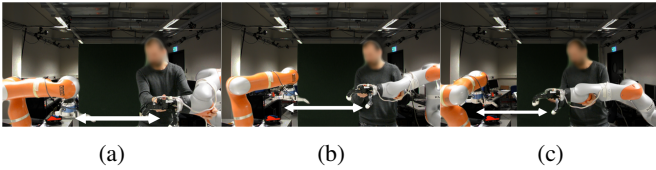


Fig. 4: Snapshots of the video illustrating coordination of the arms in free space. The real object is outside the workspace of the robots; hence, the coordination parameter γ_i is close to 0 and arm-to-arm coordination is favored. The human operator perturbs one of the arms, which leads the other arm to move in synchrony following the motion of the virtual object attached to the two end-effectors.

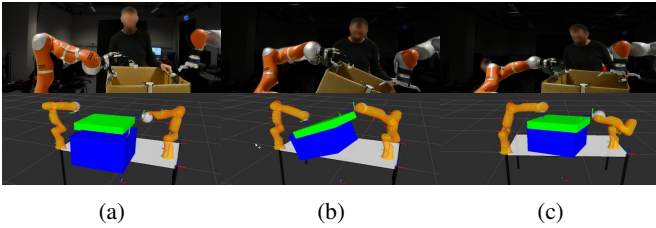


Fig. 5: Snapshots of the coordination capabilities between the arms and a moving/rotating object. The real object is inside the workspace of the robots; hence, the coordination parameter γ_i is close to 1 and the arms-to-object coordination is favored. The bottom figures show the real-time visualization of the robots, and the virtual (green) and real object (blue).

B. Coordination Capabilities

The first scenario is designed to illustrate the coordination capabilities of the arms with each other and with the object. We initially show arm-to-arm coordination capabilities by keeping the real object outside the workspace of the robots, this will force the coordination parameter γ_i to be 0, favoring arm-to-arm coordination. As the human operator perturbs one of the robot arms, the virtual object is perturbed as well, resulting in a stable synchronous motion of the other unperturbed arm (Fig. 4). Since we offer a centralized controller based on the virtual object’s motion, there is no master/slave arm; thus, when any of the robots are perturbed, the others will synchronize their motions accordingly. We then present the coordination of the arms with the object by moving it inside the workspace of the robots. The object used is a large box ($60 \times 60 \times 40\text{cm}$) held by a human operator. The edges of the

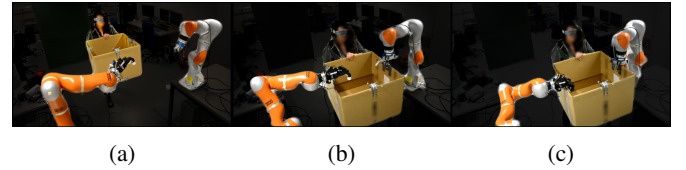
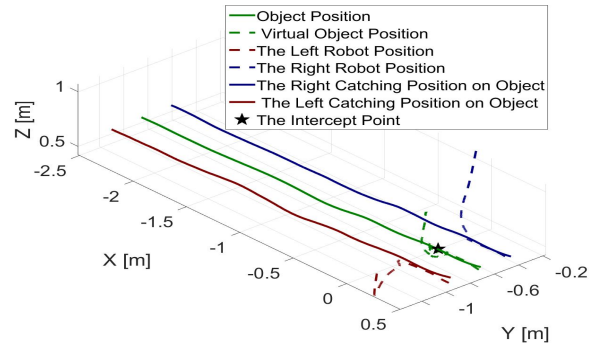


Fig. 7: Snapshots of the robots’ motion when reaching for a moving object, carried by a blindfolded operator. (a) onset of object trajectory’s prediction. (c) arms have intercepted the object and the fingers have closed on the object.

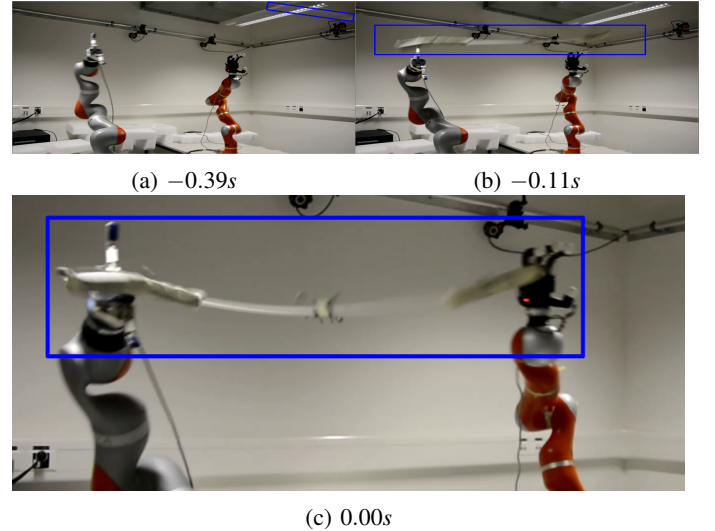


Fig. 8: Snapshots of the arms reaching for a fast moving object. In order to not damage the robot’s hands, the robot hands do not close on the object when the hands intercept the object. The robots follow the unperturbed trajectory of the object predicted by the object trajectory prediction; see Section III-A

box are specified as the *feasible reaching points*. When the box is inside the joint workspace of the robots, the operator changes the orientation and the position of the box to show the coordination capabilities between the robots and the object (Fig. 5). Due to (14), successful tracking and coordination with the object and with each other is achieved.

C. Reaching to Grab a Large Moving Object

In this second scenario, we use the same object as before. Yet, now the operator holds the box while walking towards the robots. Once the end-effectors are less than 2 cm away from

the *feasible reaching points*, finger closures of the hands are triggered and the box is successfully grabbed from the human. As can be seen in Fig.7, we blindfolded the operator to achieve *unpredictable* trajectories and avoid the natural reactions of the humans to help the robots. When the human operator carrying the box is approaching the robots, the virtual object converges to the box and follows it until the desired interception points are reached. The fingers close and the box is grabbed from the human. The initial values of γ_p and γ_o in (14) are set to 0. An example of the desired robot trajectory and the box trajectory are shown in Fig.6. As expected, the end-effectors converge to the box and continue to track its motion. As initially $\gamma_i = 0 \forall i \in \{p, o\}$, the virtual object asymptotically converges to the desired reaching position. While the box is approaching the robots, $\gamma_i \forall i \in \{p, o\}$ starts increasing and finally reaches to $\gamma_i = 1$ when the object is in the workspace of the robots. Hence, (6) generates asymptotically stable motions towards the real object instead of the intercept point. Consequently, the prediction of the intercept point does not play a vital role in grabbing the box.

D. Reaching for Fast Flying Objects

The third scenario is designed to show the coordination between the robots and a fast moving object, where a rod ($150 \times 1cm$) is thrown to the robots from $2.5m$ away, resulting in approx. $0.56s$ flying time. The distance between the base of the two robots is reduced to $[0.25 \ 1.26 \ -0.1]^T m$. The first $0.4m$ of the object trajectory initializes the trajectory prediction introduced in Section III-A. Due to inaccurate prediction of the object trajectory, the feasible intercept points need to be updated and redefined during the reach. The new feasible intercept point is chosen in the vicinity of the previous one to minimize the convergence time. As the motion of the object is fast and the predicted reaching points are not accurate, the initial values of γ_p and γ_o in (14) are set to 0.5 and 0.0, respectively. This decreases the convergence duration of the robots to the real object. Snapshots of the real robot experiments are shown in Fig. 8. Visual inspection of the data and video confirmed that the robots coordinately follow the motion of the object and intercept it at the vicinity of the predicted feasible intercept point.

E. Systematic Assessment

The success rates of our experiments are measured by defining a Boolean metric; i.e. success or failure. A trial is classified as a success if the robot intercepts the object at the desired point within less than $2cm$ error. The success rate is 85% in the box scenario and 37.5% in the rod scenario. Failures are due to either inaccuracies in the measurement of the object's state or the IK solver. To track the object, all the markers must be visible to the cameras. In the box scenario, the object's tracking was obscured partly when the object was covered by the robotic arms or the operator. In the rod scenario, the vision system loses track of the markers approx. 55% of the time. This happens, for example, when the rod rotates rapidly. To systematically assess the robustness of the algorithm to unmeasured object positions, a set of simulations

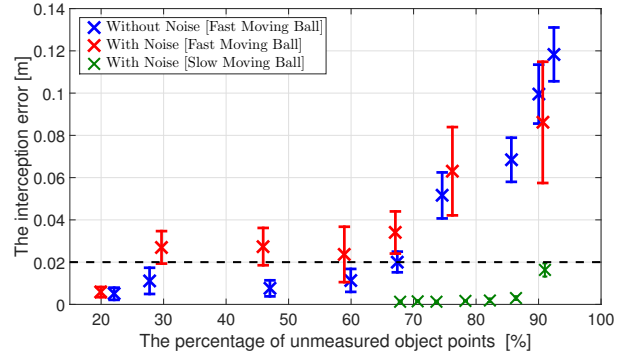


Fig. 9: The interception error is the average of the minimum distance between the ball and the end-effectors. The throwing positions are randomly chosen within the range of $[-3.5 \pm 0.1 \ -1.0 \pm 0.1 \ -0.0 \pm 0.1]^T m$. The initial ball speeds for fast and slow motions are randomly chosen within range of $8.94 \pm 0.173 \frac{m}{s}$ and $1.63 \pm 0.173 \frac{m}{s}$, respectively. We only consider trials when the ball passes through the robots workspaces. The measurement noise is simulated with pseudo-random values within the range of $\pm 0.02m$. The cut-off success/failure assessment is illustrated by the back dashed line.

was designed to reach for a moving ball. The ball diameter is $1.2m$. The simulation is repeated 130 times in total for two different object velocities; i.e. fast and slow motions. To assess the effects of the unmeasured object positions on the interception error, the desired and the real end-effectors' states are assumed equal. Results from this evaluation indicate that the interception error is directly correlated to the percentage of unmeasured object points and consequently, the velocity of the moving object (see Fig. 9). Thus, the faster the object, the more sensitive the system is to tracking inaccuracies.

Failures caused by the IK solver are mostly observed in the flying rod scenario, where the inability to accurately calculate joint-level motions corresponding to the desired end-effectors' trajectories results in errors in the robots' motions. In over 40 trials, the tracking error between the desired and the actual end-effector position at the intercept point is approximately $1.42 \pm 1.92cm$. The large variance in the error indicates the implemented IK Solvers sensitivity to the robot's kinematic configuration.

V. SUMMARY AND DISCUSSION

In this paper, we proposed a dynamical system based framework to coordinate multiple robotic arms to reach a moving object. If provided with robotic arms that can travel fast enough, our algorithm ensures that the object is intercepted by the robots in coordination and with the desired velocity aligned to the object. To improve the robustness of the framework, we define a parameter (i.e. the *coordination* parameter) and modulate it, such that the robots converge and track the real object when it is close enough.

The stability and convergence of the proposed dynamical system depends on ensuring conditions (13). As there are no constraint on the magnitude of the eigenvalues ($|\lambda_{A_j^i}|$) of $A_j^i \forall (i, j) \in \{(r, 1), \dots, (r, K_r), (p, 1), \dots, (p, K_p), (o, 1), \dots, (o, K_o)\}$, there is no analytical proof that (6) is fast enough to converge to $\gamma \xi^O$ in time. To address this challenge, a potential direction would be to estimate the parameters of (6) with respect to

the stability and the convergence rate constraints. One way to calculate the convergence rate of a dynamical system is to prove that it is exponentially stable [10].

Throughout the proofs, we assume that the intercept point is a fixed attractor. However, due to the imperfect prediction of the object trajectory, the feasible intercept postures need to be iteratively updated. Nevertheless, this does not affect the convergence of the system for two main reasons. First, when $\gamma_p < 1$ the new feasible intercept point is chosen in the vicinity of the previous ones; i.e. the convergence rate is much faster than the rate of update. Second, when the object is reachable, $\gamma_p = 1$, the virtual object converges to the real object and the position of the intercept point does not affect the convergence.

In this paper, we control the motion of the arm from initial condition (palm open, robots far from the object) to the point when the arms reach the object and the fingers are about to close on the object. Hence, there are no interaction forces (which would arise once in contact with the object). Once the fingers close on the object, the robots-object system become a closed kinematic chain. In this case, devising an appropriate force controller is necessary to coordinate the robots. Future work in multi-arm manipulation will be directed to address the challenges of devising an appropriate force controller and the transition between the position and force controller.

Since the control architecture is designed in terms of the virtual object, the coordinated motion of the robots is ensured during motion execution; furthermore, it can be guaranteed that the end-effectors will not collide. However, the proposed algorithm does not guarantee that the rest of the arms will not collide with each other. One possible direction for avoiding collision across the robots' links is to define an IK (Inverse Kinematics) null-space, which maximizes the arms distance, and solve the IK problems with respect to this null-space. In the implementation with the box and the flying bar, since the min. distance between the base of the arms is 1.29m, there is a max. of 30 cm intersection between the workspaces. Yet, the distance between the coupled reaching area on the bar is 1.2m. Hence, the arms are ensured to not collide.

One of the main advantages of the proposed framework is the computational cost. The implementation shows that the overall computation is rapid, thus enabling us to not only compute the feasible intercept point, the reaching motion and solving IK problems, but also to control a 34 (7 + 7 + 14 + 4) DOF system with one 3.4-GHz i7 PC. Even though the robot control rate is 500Hz, the algorithm is capable of running at a rate of up to 3300Hz.

Finally, we are currently working on improving the performance of (6), by learning its parameters via a convex optimization problem with respect to the workspace constraint of the robots. With this approach, we could ensure that the performance of the dynamical system is optimal and the generated motion is not infeasible for the robots to follow.

APPENDIX A PROOF OF THEOREM 1

The necessary condition for stability of (9) and asymptotic stability of the tracking error (8); i.e. $\lim_{t \rightarrow \infty} \|x_{pj}^V(t) - x_j^R(t)\| = 0 \ \forall j \in \{1, \dots, K_r\}$. (8) is

asymptotically stable if and only if there are $0 \prec P_j^R$ and $0 \prec Q_j^R$ such that ⁷

$$P_j^R A_j^R + A_j^{RT} P_j^R \prec -Q_j^R \quad \forall j \in \{1, \dots, K_r\} \quad (15)$$

If (8) is asymptotically stable, by substantiating (8) in (9a), we have:

$$\dot{x}_p^V(t) = \frac{1}{K_r + 1} \left(\dot{x}_{\gamma p}^O(t) + A_p(\theta_{A_p})(x_p^V(t) - x_{\gamma p}^O(t)) + \sum_{j=1}^{K_r} (x_{pj}^V(t) - \omega \star R_j) \right) \quad (16)$$

The first and the second derivative of $\xi_{pj}^V = r_j + \xi_p^V$ with respect to time are $\dot{\xi}_{pj}^V = \omega \times r_j + \dot{\xi}_p^V$ and $\ddot{\xi}_{pj}^V = \omega \times (\omega \times r_j) + \ddot{\xi}_p^V$, respectively. Hence, $\dot{x}_{pj}^V = \omega \star R_j + \dot{x}_p^V$ and (16) can be written as:

$$\begin{aligned} \dot{x}_p^V(t) &= \frac{1}{K_r + 1} \left(\dot{x}_{\gamma p}^O(t) + A_p(\theta_{A_p})(x_p^V(t) - x_{\gamma p}^O(t)) + \sum_{j=1}^{K_r} \dot{x}_p^V(t) \right) \\ &= \frac{1}{K_r + 1} \left(\dot{x}_{\gamma p}^O(t) + A_p(\theta_{A_p})(x_p^V(t) - x_{\gamma p}^O(t)) + K_r \dot{x}_p^V(t) \right) \\ \dot{x}_p^V(t) &= \dot{x}_{\gamma p}^O(t) + A_p(\theta_{A_p})(x_p^V(t) - x_{\gamma p}^O(t)) \end{aligned} \quad (17)$$

We propose a Lyapunov function as follows:

$$V = \frac{1}{2} (x_p^V(t) - x_{\gamma p}^O(t))^T P_p^V (x_p^V(t) - x_{\gamma p}^O(t)) \quad (18)$$

V is positive definite, radially unbounded, continuous and continuously differentiable. The derivative of V with respect to time is as follows

$$\begin{aligned} \dot{V} &= \frac{dV}{dt} = \frac{1}{2} \left((x_p^V(t) - x_{\gamma p}^O(t))^T P_p^V (\dot{x}_p^V(t) - \dot{x}_{\gamma p}^O(t)) \right. \\ &\quad \left. + (x_p^V(t) - x_{\gamma p}^O(t))^T P_p^V (x_p^V(t) - x_{\gamma p}^O(t)) \right) \end{aligned} \quad (19)$$

Substituting (5) and (17) into (19), we have:

$$\begin{aligned} \dot{V} &= \frac{dV}{dt} = \frac{1}{2} \left((x_p^V(t) - x_{\gamma p}^O(t))^T P_p^V A_p(\theta_{A_p})(x_p^V(t) - x_{\gamma p}^O(t)) \right. \\ &\quad \left. + (x_p^V(t) - x_{\gamma p}^O(t))^T A_p(\theta_{A_p})^T P_p^V (x_p^V(t) - x_{\gamma p}^O(t)) \right) \\ &= \frac{1}{2} \left((x_p^V(t) - x_{\gamma p}^O(t))^T P_p^V \left(\sum_{k=1}^{K_p} \theta_{A_{pk}} A_{pk} \right) (x_p^V(t) - x_{\gamma p}^O(t)) \right. \\ &\quad \left. + (x_p^V(t) - x_{\gamma p}^O(t))^T \left(\sum_{k=1}^{K_p} \theta_{A_{pk}} A_{pk}^T \right) P_p^V (x_p^V(t) - x_{\gamma p}^O(t)) \right) \\ &= \underbrace{\sum_{k=1}^{K_p} \theta_{A_{pk}} (x_p^V(t) - x_{\gamma p}^O(t))^T (P_p^V A_k^2 + A_k^{2T} P_p^V) (x_p^V(t) - x_{\gamma p}^O(t))}_{> 0} \underbrace{\phantom{\sum_{k=1}^{K_p} \theta_{A_{pk}} (x_p^V(t) - x_{\gamma p}^O(t))^T (P_p^V A_k^2 + A_k^{2T} P_p^V) (x_p^V(t) - x_{\gamma p}^O(t))}}_{< -Q_k^R} \\ &< 0 \end{aligned} \quad (20)$$

Therefore, dynamical system (17) and (9a) are globally stable; i.e. x_p^V and \dot{x}_p^V are bounded as $x_{\gamma p}^O(t)$ and $\dot{x}_{\gamma p}^O(t)$ are bounded. Since \dot{V} is finite, Barbalat's lemma [10] indicates that the attractor is globally asymptotically stable; i.e.:

$$\begin{aligned} \lim_{t \rightarrow \infty} \|x_p^V - x_{\gamma p}^O(t)\| &= 0 \quad \Rightarrow \\ \lim_{t \rightarrow \infty} \|\dot{\xi}_p^V - (\gamma_p \dot{\xi}_p^O + \dot{\gamma}_p \xi_p^O)\| &= 0, \quad \lim_{t \rightarrow \infty} \|\xi_p^V - \gamma_p \xi_p^O\| = 0 \end{aligned} \quad (21)$$

■, c.q.f.d.

ACKNOWLEDGMENTS

This work was supported by EU projects *Cogimon H2020-ICT-23-2014* and EU FP7 Project *RoboHow* (Grant Agreement Number 288533). The authors kindly thank Nicolas Sommer and Felix Duvallet for preparing the experimental setup with Allegro hand and Seyed Mohammad Khansari-Zadeh and Klas Kronander for providing IIWA connectivity package.

⁷The proof is omitted due to lack of space. Asymptotic stability of (8) can be proven using the non-negative Lyapunov function: $V = \frac{1}{2} (x_{pj}^V(t) - x_j^R(t))^T P_j^R (x_{pj}^V(t) - x_j^R(t))$.

REFERENCES

- [1] F. Aghili. Adaptive control of manipulators forming closed kinematic chain with inaccurate kinematic model. *IEEE/ASME Transactions on Mechatronics*, 18(5):1544–1554, Oct 2013.
- [2] Bassam Bamieh and Laura Giarre. Identification of linear parameter varying models. *International journal of robust and nonlinear control*, 12(9):841–853, 2002.
- [3] M. C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. ISBN 978-0387310732.
- [4] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani. Six-dof impedance control of dual-arm cooperative manipulators. *IEEE/ASME Transactions on Mechatronics*, 13(5):576–586, Oct 2008.
- [5] Fabrizio Caccavale and Masaru Uchiyama. Cooperative manipulators. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 701–718. Springer Berlin Heidelberg, 2008.
- [6] Rachel O. Coats and John P. Wann. Reaching a better understanding of the control of bimanual movements in older adults. *PLoS ONE*, 7(10):e47222, 10 2012.
- [7] Benjamin Cohen, Mike Phillips, and Maxim Likhachev. Planning single-arm manipulations with n-arm robots. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.
- [8] Zlatko Emedi and Alireza Karimi. Fixed-structure LPV Discrete-time Controller Design with Induced l_2 -Norm and H_2 Performance. *International Journal of Control*, 2015.
- [9] A. Gams, A. Ude, and J. Morimoto. Accelerating synchronization of movement primitives: Dual-arm discrete-periodic motion of a humanoid robot. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2754–2760, Sept 2015. doi: 10.1109/IROS.2015.7353755.
- [10] Hassan Khalil. *Nonlinear Systems*. Prentice Hall, 2002. ISBN 0-13-067389-7.
- [11] S.M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.
- [12] S. Kim, A. Shukla, and A. Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, 2014.
- [13] Seungsu Kim and Aude Billard. Estimating the nonlinear dynamics of free-flying objects. *Robotics and Autonomous Systems*, 60(9):1108 – 1122, 2012. ISSN 0921-8890.
- [14] Nejc Likar, Bojan Nemeč, and Leon Lajpah. Virtual mechanism approach for dual-arm manipulation. *Robotica*, FirstView:1–16, 8 2013. ISSN 1469-8668.
- [15] Y.-H. Liu and S. Arimoto. Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations. *International Journal of Robotics Research*, 17(3):232–247, 1998.
- [16] J. Y. S. Luh and Y. F. Zheng. Constrained relations between two coordinated industrial robots for motion control. *International Journal of Robotics Research*, 6(3):60–70, September 1987.
- [17] J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal. Comparative experiments on task space control with redundancy resolution. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3901–3908, 2005.
- [18] Klaus Neumann and Jochen J. Steil. Learning robot motions with stable dynamical systems under diffeomorphic transformations. *Robotics and Autonomous Systems*, 70: 1 – 15, 2015. ISSN 0921-8890.
- [19] S. S. M. Salehian, M. Khoramshahi, and A. Billard. A dynamical system approach for softly catching a flying object: Theory and experiment. *IEEE Transactions on Robotics*, 32(2):462–471, April 2016. ISSN 1552-3098. doi: 10.1109/TRO.2016.2536749.
- [20] S. Schaal, S. Kotosaka, and D. Sternad. Nonlinear dynamical systems as movement primitives. In *Humanoids2000, First IEEE-RAS International Conference on Humanoid Robots*, 2000.
- [21] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic. Dual arm manipulation survey. *Robotics and Autonomous Systems*, 60(10):1340 – 1353, 2012.
- [22] R. Suda, K. Kosuge, and H. Kakuya. Object-impedance-based cooperative handling of object by mobile robot helper and human using visual and force information. In *Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on*, volume 1, pages 592–597 vol.1, July 2003. doi: 10.1109/AIM.2003.1225161.
- [23] Dong Sun and James K. Mills. Manipulating rigid payloads with multiple robots using compliant grippers. *IEEE/ASME Transactions on Mechatronics*, 7(1):23–34, Mar 2002.
- [24] N. Vahrenkamp, M. Do, T. Asfour, and R. Dillmann. Integrated grasp and motion planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2883–2888, 2010.
- [25] N. Vahrenkamp, T. Asfour, and R. Dillmann. Simultaneous grasp and motion planning: Humanoid robot armarii. *IEEE Robotics Automation Magazine*, 19(2):43–57, 2012.
- [26] Jan-Willem van Wingerden and Michel Verhaegen. Subspace identification of bilinear and LPV systems for open- and closed-loop data. *Automatica*, 45(2):372 – 381, 2009.
- [27] David Vernon, Claes von Hofsten, and Luciano Fadiga. *A Roadmap for Cognitive Development in Humanoid Robots*, volume 11 of *Cognitive Systems Monographs*. Springer Berlin Heidelberg, 2011.
- [28] Meiling Wang, Minzhou Luo, Tao Li, and Marco Ceccarelli. A unified dynamic control method for a redundant dual arm robot. *Journal of Bionic Engineering*, 12(3):

361 – 371, 2015.

- [29] D. Williams and O. Khatib. The virtual linkage: a model for internal forces in multi-grasp manipulation. In *IEEE International Conference on Robotics and Automation*, pages 1025–1030 vol.1, 1993.
- [30] T. Wimbock, C. Ott, and G. Hirzinger. Analysis and experimental evaluation of the intrinsically passive controller (ipc) for multifingered hands. In *IEEE International Conference on Robotics and Automation*, pages 278–284, 2008.
- [31] T. Wimböck, C. Ott, A. Albu-Schäffer, and G. Hirzinger. Comparison of object-level grasp controllers for dynamic dexterous manipulation. *International Journal of Robotics Research*, 31(1):3–23, 2012.
- [32] Thomas Wimböck and Christian Ott. Dual-arm manipulation. In *Towards Service Robots for Everyday Environments*, volume 76 of *Springer Tracks in Advanced Robotics*, pages 353–366. Springer Berlin Heidelberg, 2012.
- [33] Thomas Wimböck, Christian Ott, Alin Albu-Schäffer, and Gerd Hirzinger. Comparison of object-level grasp controllers for dynamic dexterous manipulation. *The International Journal of Robotics Research*, 31(1):3–23, 2012.
- [34] Wen-Hong Zhu. On adaptive synchronization control of coordinated multirobots with flexible/rigid constraints. *IEEE Transactions on Robotics*, 21(3):520–525, June 2005. ISSN 1552-3098.