Dynamical System-Based Motion Planning for Multi-Arm Systems: Reaching for Moving Objects

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

Abstract

The use of coordinated multi-arm robotic systems allows to preform manipulations of heavy or bulky objects that would otherwise be infeasible for a single-arm robot. This paper concisely introduces our work on coordinated multi-arm control [Salehian *et al.*, 2016a], where we proposed a *virtual object* based dynamical systems (DS) control law to generate autonomous and synchronized motions for a multi-arm robot system. We show theoretically and empirically that the multi-arm + virtual object system converges asymptotically to a moving object. The proposed framework is validated on a dual-arm robotic system. We demonstrate that it can re-synchronize and adapt the motion of each arm in a fraction of a second, even when the object's motion is fast and not accurately predictable.

1 Introduction

Humans display an underlying coordination with external agents while manipulating objects. They are capable of coordinating both arms in synchrony to accomplish tasks such as lifting, carrying and reaching for large or heavy objects which require both *spatial* and *temporal* constraints [Coats and Wann, 2012]. 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 tasks is beyond a single arm's capabilities.

This is also true for robotic systems. A dual or multi-arm system extends the workspace of a single robot arm. Hence, it allows for complex manipulation of heavy or large objects that would otherwise be infeasible for single-arm systems. One can imagine applications in smart-factories or homes, that would benefit from such coordinated motion strategies. Examples include, grabbing, catching, lifting of large-sized parts traveling on a cart, a conveyor belt, carried by humans (see Fig. 1) or even flying towards the multi-arm robot system. Research in the field of multi-arm control has mostly focused on devising control strategies for coordinated and stable manipulation of partially or fully grasped objects by the multi-arm system, [Smith *et al.*, 2012; Wimböck *et al.*, 2012]. Seldom work has focused on developing *reaching strategies* that a multi-arm system can use to reach and grab *moving* objects.

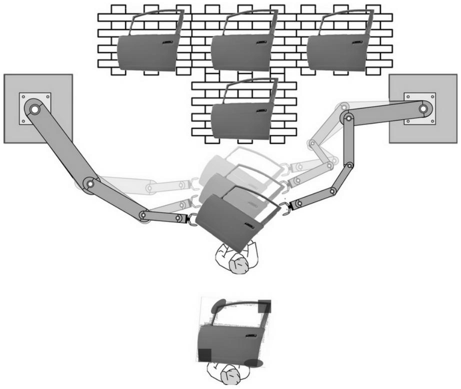


Figure 1: Illustration of a possible application where a multi-arm system has to reach for large moving objects carried to them by humans. The arms move in synchrony towards the coupled feasible reaching points of the object *(e.g. squares or ellipses)*. As the object approaches, the arms' end-effectors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

We consider a scenario in which an object with arbitrary mass and shape is moving towards a multi-arm system. The object dynamics is unknown and the only available information about the object is the position and the orientation of *coupled feasible reaching points*, which are the preferred reaching points on the object, specified by the operator (see Fig. 1). Accomplishment of the multi-arm *reach* is fulfilled if and only if all the arms simultaneously intercept the object on its feasible reaching points. From a robotics point of view, this condition can be translated to *task* and *coordination* constraints. *Task constraints* impose position and velocity constraints at the object's interception. Position constraints ensure that the planned motion of each end-effector is coordinated with the feasible reaching positions of the object. Whereas the velocity constraints allow readjustments of the hands' posture if there are uncertainties in the objects' motion. *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.

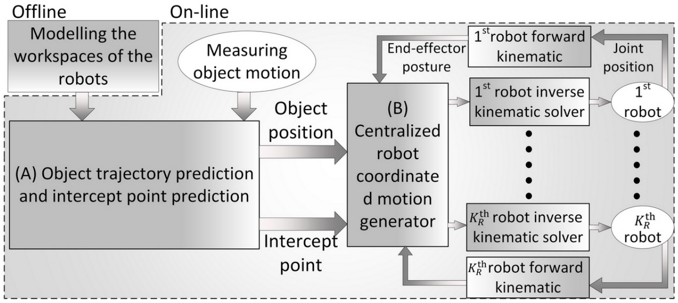

Offline
On-line
Modelling the workspaces of the robots
Measuring object motion
Object position
(A) Object trajectory prediction and intercept point prediction
Intercept point
End-effector posture
(B) Centralized robot coordinate d motion generator
1st robot forward kinematic
1st robot inverse kinematic solver
Joint position
1st robot
K_R^{th} robot inverse kinematic solver
K_R^{th} robot
K_R^{th} robot forward kinematic

Figure 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.

In [Salehian *et al.*, 2016a], we propose a *virtual object* based dynamical system that generates coordinated trajectories for a multi-arm robot system to reach the moving object simultaneously. The virtual object is used to both *coordinate* the motion of the multiple robots with each other and with a moving object (target), such that the robots reach the moving object in *synchrony*, whilst ensuring collision avoidance between the *end-effectors* (Figure 1). In this paper, we provide a high-level introduction of this work.

2 Problem Statement

In order to achieve stable and coordinated reaching motions for multiple arms when the target is in motion, several problems need to be solved simultaneously: (i) prediction of the objects trajectory; (ii) computing intercept points for each arm and (iii) dynamically planning coordinated motion of the robot arms towards their corresponding intercept points. An overview of the proposed framework is illustrated in Figure 2. As seen in the illustration, the intercept point is computed based on the trajectory of the object in sub-component (A) . It uses a probabilistic representation of the reachable workspace of the multi-arm system (learned off-line prior to the experiment) and an on-line step in which it continuously measures the objects pose from a visual tracking system. Sub-component (B) uses the intercept points and the current end-effector poses to generate the desired end-effector poses of the K_R-robot multi arm system. For simplicity and practicality, Fig. 3 summarize the most relevant notation used throughout the paper.[1]

The real object is represented by a set of *feasible reaching points* ($\xi_i^O \in \mathbb{R}^d \; \forall i \in \{1, \dots, K_R\}$), defined by the user. Their resultant vector ($\xi^O(t) \in \mathbb{R}^d$) corresponds to the position of the object. The *virtual object* is a duplicate of these set of points, with the end-effectors of each robot attached to the virtual reaching points ($\xi_i^V \in \mathbb{R}^d \; \forall i \in \{1, \dots, K_R\}$) by zero length springs and dampers (Fig.3). To model the motion of the *virtual object*, we first ensure that the object will travel/travels through the workspace of the robots. In order to find the feasible intercept point, the kinematic feasibility of the predicted reaching points must be evaluated. Once the real object starts moving towards the robots, we predict its motion through simple ballistic motion estimation, which computes the object's progress ahead of time. We then determine a point along this trajectory where the object will become reachable by all robot arms. We name this point the *feasible intercept point* ($\xi^O(T^*)$). As the motion of the object is not accurately predicted, the object might not pass through the predicted point at T^*, however, its likely that the object will move in that direction.

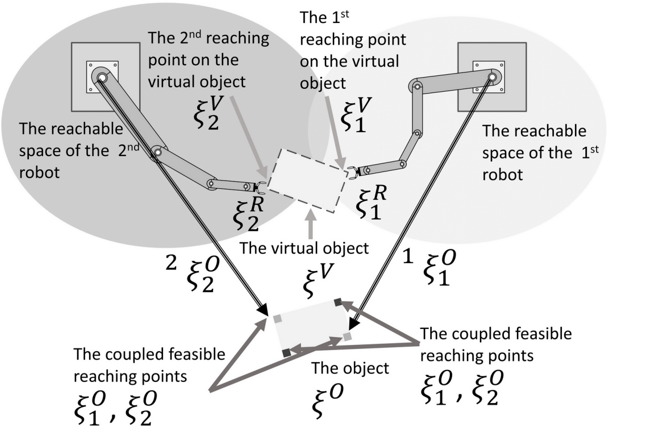

The 2nd reaching point on the virtual object
ξ_2^V
The 1st reaching point on the virtual object
ξ_1^V
The reachable space of the 2nd robot
The reachable space of the 1st robot
ξ_2^R
ξ_1^R
The virtual object
ξ^V
$^2\xi_2^O$
$^1\xi_1^O$
The coupled feasible reaching points
ξ_1^O, ξ_2^O
The object
ξ^O
The coupled feasible reaching points
ξ_1^O, ξ_2^O

Figure 3: An illustration of the variables defined in the paper is presented. The reachable areas are feasible areas for grasping the object. Except for $^2\xi_2^O$ and $^1\xi_1^O$, the variables are expressed in the reference frame located on the desired intercept point; i.e. $\xi^O(T^*) = \begin{bmatrix}0 & \dots & 0\end{bmatrix}^T$.

3 Centralized Robot Coordinated Motion Generator

The motion of the virtual object is generated by the following class of fully observable continuous-time Linear Parameter Varying (LPV) systems.

$$\dot{x}^V(t) = \frac{1}{K_R+1}\left(\dot{x}^O_\gamma + A(\theta)(x^V(t) - x^O_\gamma(t)) + \sum_{j=1}^{K_R}(U_j)\right) \tag{1}$$

Where $x^O_\gamma(t) = \begin{bmatrix}\gamma(t)\xi^O(t) & \gamma(t)\dot{\xi}^O(t) + \dot{\gamma}(t)\xi^O(t)\end{bmatrix}^T$ and $x^V(t) = \begin{bmatrix}\xi^V(t) & \dot{\xi}^V(t)\end{bmatrix}^T$ are the states of the real and the virtual objects, respectively. $0 < \gamma(t) < 1$ is the *coordination* parameter and of class C^1. The origin is located on the desired intercept point; i.e. $\xi^O(T^*) = [0 \quad \dots \quad 0]^T$. $\theta \in \mathbb{R}^{K\times 1}$ is a vector of scheduling parameters, $\theta = \begin{bmatrix}\theta_1 & \dots & \theta_K\end{bmatrix}^T$.[2] $A(.)\mathbb{R}^{K\times 1} \rightarrow \mathbb{R}^{2d\times 2d}$ is the affine dependence of state-space matrices on the scheduling parame-

[1] In this paper, we mainly focus on part (iii), the readers are referred to [Salehian *et al.*, 2016a] for detailed information about part (i) and (ii).

[2] The scheduling parameters can be a function of time t, the states of the system $x^V(t)$ or external signals $d(t)$, i.e. $\theta_A(t, \xi^V(t), d(t))$. In the rest of the paper, we assume that it is

ter and the state vectors; i.e. $A(\theta) = \sum_{k=1}^{K} \theta_k A_k \quad A_k \in \mathbb{R}^{2d\times 2d} \quad \theta_k \in \mathbb{R}$. To determine the parameters of this LPV system, we formulate $A(.)$ as a Gaussian Mixture Regression (GMR), similar to [Salehian *et al.*, 2016b]. This technique inherently results in normalized scheduling parameters; i.e. $0 < \theta_j \leq 1 \, \forall j \in \{1, \dots, K\}$, $\sum_{k=1}^{K} \theta_k = 1$. U_j is the interaction effect of the tracking controller of the j^{th} end-effector on the virtual object:

$$U_j = \dot{x}_j^R(t) + A_j^R(x_j^V(t) - x_j^R(t)) \tag{2}$$

Where $A_j^R \in \mathbb{R}^{2d\times 2d}$ 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 j^{th} point on the virtual object and the end-effector:

$$\dot{x}_j^R(t) = \dot{x}_j^V(t) - A_j^R(x_j^V(t) - x_j^R(t)) \tag{3}$$

Theorem 1 *The dynamical systems given by* (1) *asymptotically converge to* $\begin{bmatrix}\gamma(t)\xi^O(t) & \gamma(t)\dot{\xi}^O(t) + \dot{\gamma}(t)\xi^O(t)\end{bmatrix}^T$ *and the* j^{th} *end-effector asymptotically converges to the* j^{th} *reaching area on the virtual object i.e.*

$$\lim_{t\to\infty} \| x_j^R(t) - x_j^V(t) \| = 0 \tag{4}$$

$$\lim_{t\to\infty} \| \xi^V(t) - \gamma(t)\xi^O(t) \| = 0 \tag{5}$$

$$\lim_{t\to\infty} \| \dot{\xi}^V(t) - (\gamma(t)\dot{\xi}^O(t) + \dot{\gamma}(t)\xi^O(t)) \| = 0 \tag{6}$$

if there are P^V, P_j^R, Q^V, Q_j^R *such that:*

$$\begin{cases} 0 \prec P^V \; 0 \prec P_j^R & 0 \prec Q^V \; 0 \prec Q_j^R \\ P_j^R A_j^R + {A_j^R}^T P_j^R \prec -Q_j^R & \forall j \in \{1, \dots, K_R\} \\ P^V A_k + {A_k}^T P^V \prec -Q^V & \forall k \in \{1, \dots, K\} \\ 0 \leq \theta_k \leq 1, & \end{cases} \tag{7}$$

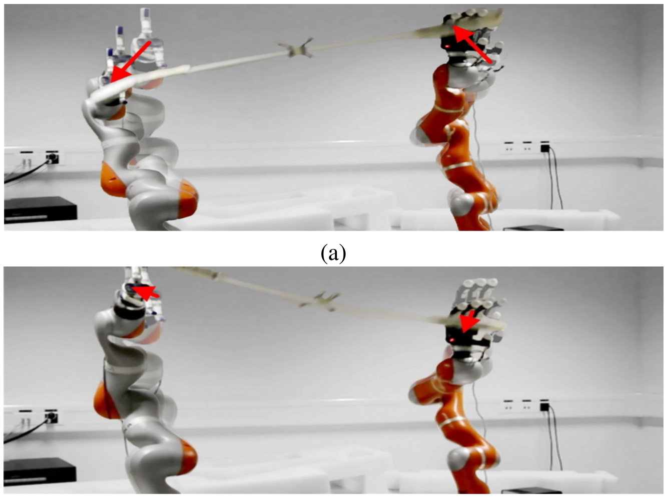
where $0 \prec$ and $\prec 0$ refer to positive and negative definiteness of a matrix, respectively.

Proof: see [Salehian *et al.*, 2016a].

If the *coordination* parameter is set to zero $(\gamma(t) = \dot{\gamma}(t) = 0)$, (1) generates asymptotically stable motions towards the predicted intercept point. Hence, the coordination between the robots and the object is lost and the coordination between the robots is preserved . If $\gamma(t) = 1$ and $\dot{\gamma}(t) = 0$, (1) generates asymptotically stable motions towards the real object: i.e. perfect *coordination* with the object.[3] However, in this case, there is no guarantee that the virtual object intercepts the real object inside the workspace of the robots, i.e. due to the robots constraints, the coordination between the robots is lost. Thus, one can vary the values of the coordination parameters between 0 and 1 such that they are one at the vicinity of the desired intercept time as follows:

$$\dot{\gamma} = \frac{1-\gamma}{\|\xi^O(t) - \xi^O(T^*)\| + \varepsilon} = \frac{1-\gamma}{\|\xi^O(t)\| + \varepsilon} \quad \gamma(0) = 0 \tag{8}$$

. (8) 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. $\gamma(T^*) = 1$. Hence, the robots coordinately track the desired reaching points on the object.



(a)

(b)

Figure 5: Snapshots of the arms reaching for a fast moving object in two different trials. In order to not damage the robot's hands, they do not close on the object when they intercept the object. The arrows show the direction of the robots' motions. Even though the robots do not move in same directions, they move in synchrony such that the resultant motion is coordinated with the object.

4 Empirical Validation

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 fingers of the robotic hands are controlled via 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.

Our empirical validation is composed of three parts which demonstrate the controller's ability: (i) coordination of the multi-arm systems (arm-to-arm); (ii) adaptation of the two arms' motions in coordination so as to reach and grab a large moving object and (iii) rapid adaptation and coordination of a dual-manual 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`

The first scenario is designed to illustrate the coordination capabilities of the arms with each other through the virtual object. 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

only function of the states of the system $x^V(t)$ and the arguments are dropped for simplicity.

[3] We assume that the dynamical system (1) is fast enough to converge to an acceptable neighborhood around the desired trajectory $\gamma \begin{bmatrix}\xi^O(t) & \dot{\xi}^O(t)\end{bmatrix}^T$ before T^*; i.e. $\|\xi^V(T^*) - \gamma\xi^O(T^*)\| \leq \varepsilon$. Where $\varepsilon$is a small positive constant.

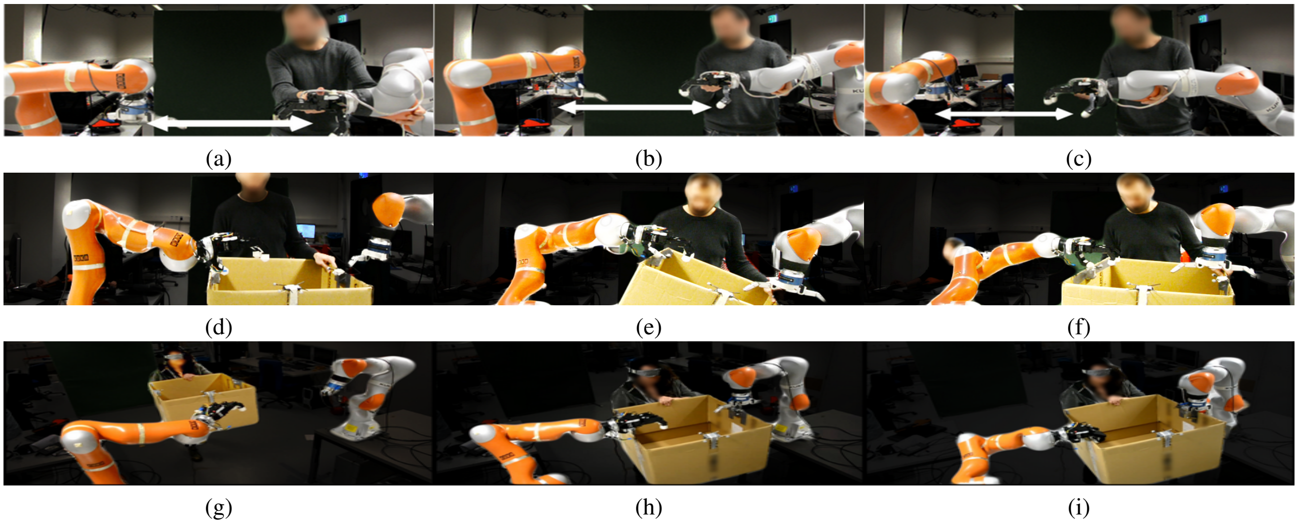


Figure 4: Snapshots of the video illustrating coordination of the arms with each other and with the object. In (a), (b) and (c), the human operator perturbs one of the arms, which leads the other arm to move in synchrony following the motion of the virtual object attached to the two end-effectors. In (g), (h) and (i) snapshots of the robots' motion when reaching for a moving object, carried by a blindfolded operator. (g) onset of object trajectory's prediction. (i) arms have intercepted the object and the fingers have closed on the object.

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. We used a box as an object, whose edges are specified as the *feasible reaching points*. When the box is inside the 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, see Fig. 4.

In this second scenario, a blindfolded 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. 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, see Fig.4. As expected, the end-effectors converge to the box and continue to track its motion. As initially $\gamma = 0$, the virtual object asymptotically converges to the feasible intercept point. While the box is approaching the robots, γ starts increasing and finally reaches to $\gamma = 1$ when the object is in the workspace of the robots.

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. Due to inaccurate prediction of the object trajectory, the feasible intercept points need to be updated and redefined during the motion execution. 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 in (8) are set to 0.5. This decreases the convergence duration of the robots to the real object. Snapshots of the real robot experiments are shown in Fig. 5. 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.

5 Discussion

This paper summarizes our work on coordinating multiple robot arms in the task of reaching for a moving object. Based on a centralized dynamical system framework, our algorithm ensures that the object is intercepted by the robots in synchrony and aligned to the velocity of the object. The robots are guided by a virtual object that converges to the real object through a coordination parameter, which is modulated based on the closeness of the real object to the robots. One of the key challenges in multi-robot control is self-collision avoidance. In our current architecture, we can only guarantee self-collision avoidance between the end-effector, i.e. at the task-level. To ensure self-collision avoidance at the joint-level, one could explore the joint workspace of the multi-arm system to learn the regions that might lead to collision and use it as a constraint in a centralized inverse kinematics solver. By developing such a learning strategy, we will be able to provide coordination both at the task and joint-level.

Acknowledgments

This work was supported by EU projects *Cogimon* $H2020 - ICT - 23 - 2014$.

References

[Coats and Wann, 2012] 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.

[Salehian *et al.*, 2016a] Seyed Sina Mirrazavi Salehian, Nadia Figueroa, and Aude Billard. Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In *Proceedings of Robotics: Science and Systems*, 2016.

[Salehian *et al.*, 2016b] Seyed Sina Mirrazavi 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.

[Smith *et al.*, 2012] Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V. Dimarogonas, and Danica Kragic. Dual arm manipulationa survey. *Robotics and Autonomous Systems*, 60(10):1340 – 1353, 2012.

[Wimböck *et al.*, 2012] 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.