Chapter 7

Learning and Coordination of Movement Primitives for Bimanual Manipulation Tasks using Concurrent Synchronization

Ashwin Dani, Iman Salehi and Kyle Hunte

7.1 Introduction

Coordination of multiple robots is very useful in advanced manufacturing applications, where the robots can be used to pick up, transport and manipulate heavy and/or deformable objects, performing a wire harness assembly or a screw assembly. This chapter focuses on methods that involve coordination between the two arms. Dual arm manipulation can be broadly classified into uncoordinated and coordinated manipulation [Smith *et al.* (2012)]. Coordinated dual arm manipulation can further be categorized into goal-coordinated and bimanual manipulations. In goal-coordinated manipulation, the two arms are not interacting with each other but they are coordinating to achieve a same end goal, such as filling up a box with wooden pallets, sanding different parts of a workpiece to achieve a time-efficient sanding operation, bimanual cleaning operations [Langsfeld *et al.* (2016)]. In contrast, bimanual manipulation requires the arms to interact with the same object. Naturally, bimanual manipulation requires two arms to synchronize and move with a defined transformation to carry out a common goal, such as tying a knot by pulling different parts of the string, uncurling the curled wires in wire harness assembly, carrying heavy wooden/metal pallets, transporting large deformable objects like fabric, composite materials. In this chapter, we present coordinated control laws to synchronize motions of two arms for bimanual manipulation, where the learning and coordination for bimanual manipulation tasks are considered. The synchronization of arms is achieved by considering transformations such as a fixed rigid transformation, reflection/mirror transformation or time-varying transformations. The dynamics of the task for one of the arms are first learned from demonstration data using dynamic movement primitives (DMPs). In the same spirit as [Chung and Slotine (2009); Chung *et al.* (2013)], a concurrent tracking and synchronization laws based on contraction analysis are derived for the two arms to track the trajectory generated by the DMP and achieve coordination of two arms.

In this chapter, DMP representation is used for encoding both discrete and pe-

riodic motion. Kinesthetic demonstration obtained from robotic arm manipulator is used to learn the DMP. The demonstration may include the motion (position, velocity, acceleration) in robot joint space or robot task space represented using 3D positions and orientation angles. If the dynamics of end effector pose (position+orientation) are learned using DMPs, inverse kinematics solution is required to convert the end effector pose to joint angles during implementation using robot arm manipulators. Learning the dynamics of two arms using independent sets of DMPs does not take into consideration the task-specific spatio-temporal constraints of the two arms. Hence, the two arms controlled using independently learned sets of DMPs might fail to complete the bimanual task, especially under perturbations (e.g., one arm is pushed while dual-arm wooden pallet transportation operation is in action). This issue can be circumvented by synchronizing the motions of two arms according to task-specific constraints. The end-effector motion dynamics of one of the arms is modeled using a set of DMPs. The learned set of DMPs is used as the desired trajectory for one arm, while a pose transformation is applied to obtain the desired end-effector trajectory for the second arm. The transformation can be different based on the task objectives, and it can be any combination of translation, rotation and reflection. The transformation depends on the geometry of the object, which can be estimated using vision based estimation algorithms (see e.g., [Dani *et al.* (2012); Dani and Dixon (2010); Gans *et al.* (2008); Yang *et al.* (2015); Chwa *et al.* (2016)]).

For ensuring tracking of the desired trajectories of both the arms, a tracking controller is designed for each arm. The desired trajectory for one of the arms is obtained using a DMP and the other arm is obtained using a task space transformation between two arms. In addition, the synchronization between both the arms is essential in bimanual manipulation tasks for ensuring the temporal and spatial constraints between two arms are satisfied at all times. The synchronization between the motions of two arms is achieved by adding tracking error from one arm as feedback to the controller of another arm, called as a coupling term. The gains of the feedback term are selected such that the error dynamics for the two arms are contracting ([Lohmiller and Slotine (1998)]). Solutions of contracting dynamics are robust to external perturbations and exponentially forgets the initial condition. Hence, the solutions of the error dynamics converge towards each other. The contraction ensures robust tracking of the desired trajectory of two arms in the task space even under perturbations.

The synchronization laws presented in this chapter are generic and can be used for synchronization of more than 2 arms coordinating in different configurations, e.g., a) a cyclic configuration, b) all-to-all configuration. In cyclic structure, every agent is coupled only with the next and prior agents and all the couplings are bidirectional couplings [Chung and Slotine (2009); Chung *et al.* (2013); Bandyopadhyay *et al.* (2017)]. In all-to-all structure, every agent is coupled with all the other agents in the system and similar to the cyclic structure, all the couplings are bidirectional.

For dual-arm manipulation, both of these configurations coincide with each other.

Three different experiments are presented. These experiments are inspired by some of the commonly occurring tasks in manufacturing and assembly operations. Some examples include bimanual motion coordination for holding a flexible wire with two hands for an insertion into a pin operation, coordinating motions of two hands to pull a box in which an object is kept which is being supported by the other arm, motion coordination for bimanual surgeon's knot tying operation or bimanual paper folding operation for origami folding [Balkcom and Mason (2008); Namiki and Yokosawa (2015)].

Experiment 1: In the first experiment, a wooden pallet transportation task is conducted using a Baxter robot. End effector data is collected in task space and a DMP is learned based on the position and orientation data. For this task, the end-effector task space pose transformation between two arms is a nearly fixed transformation. Using the tracking and synchronization control laws the task of moving the wooden pallet from one location to other is performed. This experiment shows utility of the bimanual coordination approach for manufacturing tasks involved in wire harness assembly mentioned above. Perturbations are added to one of the arms while the pallet transportation task is in motion. The results show that the other arm synchronizes its motion to the perturbed arm's motion exponentially fast. The synchronization feature is important so that the robot does not drop the object while the transportation task is in process.

Experiment 2: In the second experiment, a capability of adaptation to the change in goal location while the task is progress is demonstrated. It is shown in experiments that with the change in reaching goal location of one of the two robot arms in task space, the other arm is able to synchronize its motion to the new location without disturbing the bimanual manipulation task. Such a capability is very useful in the manufacturing tasks such as sliding a box on the table with one of the robot arms and an object kept in the box is supported by the other robot arm.

Experiment 3: In the third experiment, a knot tying task is carried out using coordination of two arms. DMP is learned for encoding task space motion of one of the two arms. The knot tying task requires the other arm to move with a reflection transformation. Knot tying operations are ubiquitous in robotics manufacturing. For example, tying knots during surgery using robot arms, tying knots in parachute packing operations, etc. An implementation of knot tying task on Baxter robot is presented.

Related Work

Several works have introduced control designs that enable coordination between the two arms. Classical approaches consider force or motion control of closed chains formed by the arms and an object. Methods involving input-output linearization [Yun and Kumar (1991); Sarkar *et al.* (1997)], hybrid/force control [Doulgeri and Golfakis (2006); Watanabe *et al.* (2005); Tinós *et al.* (2006); Erhart and

Hirche (2013)], impedance control [Schneider and Cannon (1992); Bonitz and Hsia (1996a,b); Erhart *et al.* (2013)], neuro-adaptive control [Gueaieb *et al.* (2007); Zhao and Cheah (2009); Lian *et al.* (2002)] have been used to either maintain a desired object pose or track a desired pose trajectory. In contrast, this chapter treats the two arms as separate agents and focuses on their coordinated motion. Earlier work has demonstrated the use of a leader-follower architecture in the coordinated motion of the two arms [Kume *et al.* (2007); Sun and Mills (2002); Zhu (2005)]. In such configurations, the leader is assumed to be the desired or optimal trajectory and the follower arm is controlled to follow the trajectory (after suitable transformations) of the leader arm. In [Dauchez *et al.* (2005); Huebner *et al.* (2009); Maitin-Shepard *et al.* (2010)], methods that rely on visual feedback to coordinate the arms via visual servoing are developed. Such methods ensure that the motion of the arms minimizes the error between the observed features and the desired features of both the arms. Motion planning of multiple arms is studied for coordinated motion of dual arm manipulators and grasp planning using multiple robots [Knepper *et al.* (2013); Barraquand and Ferbach (1994); Dogar *et al.* (2015); Bien and Lee (1992); Sezgin *et al.* (1997); Harada *et al.* (2014); Basile *et al.* (2012)]. Motion planning-based methods are capable to joint space control that take into account obstacles. Control architectures for formation control of multiple robot arms has been studied in [Li *et al.* (2008); Sieber *et al.* (2013); Hirata *et al.* (2003)].

Another approach for creating robot trajectories is based on learning from demonstration (LfD), which requires demonstration of the task to be performed by the robot. Learning-based approaches [Zöllner *et al.* (2004); Calinon *et al.* (2010)] leverage statistical learning to teach robots new tasks from demonstrated data and provide the ability to perform those tasks by adapting to the changes in the workspace. One of the LfD algorithms, called DMP, represents both learnable point attractor systems and limit cycle attractor systems which can be used to encode discrete and periodic trajectories, respectively [Ijspeert *et al.* (2013)]. Besides the attractor system, DMP consists of a learnable autonomous forcing term which is guided by a canonical system. DMPs can be used to learn motion trajectories such as point-to-point reaching motions, grinding, polishing operations, etc. which involve either discrete or periodic motions of the end-effector in workspace. DMP can also encode the periodic motion and its transient behavior [Ernesti *et al.* (2012)] for single arm motion learning. In this representation, an oscillator, which is a stable limit cycle is used as a canonical system that guides the forcing term. In [Park *et al.* (2008)], the online obstacle avoidance feature is incorporated into a DMP by simple addition of repelling force around obstacles in acceleration term. In [Tamosiunaite *et al.* (2011)], simultaneous goal and parameter learning is carried out for a pouring task. In [Bitzer and Vijayakumar (2009)], learning of DMPs in joint space is discussed for the adaptation of DMP learned in joint space. In [Matsubara *et al.* (2010, 2011)], a method of learning DMPs from multiple demonstrations by finding the styles in the shapes of desired attractor landscapes from multiple

demonstrations without losing the useful properties of DMPs is discussed. DMPs are also used in joining movement sequencing where smooth and natural transitions in position and velocity are generated with modifications to original DMPs [Nemec and Ude (2012); Kulvicius *et al.* (2012)]. DMP has been used for formation control of multi-agent systems in [Umlauft *et al.* (2014); Thota *et al.* (2016)].

Table 7.1: An Overview of Literature on Bimanual Manipulation.

Approaches	Methods	Literature
Classical	Input/Output Linearization	[Yun and Kumar (1991)],[Sarkar *et al.* (1997)]
	Hybrid/Force Control	[Doulgeri and Golfakis (2006)], [Watanabe *et al.* (2005)] [Tinós *et al.* (2006)], [Erhart and Hirche (2013)]
	Impedance Control	[Schneider and Cannon (1992)] [Bonitz and Hsia (1996a,b); Erhart *et al.* (2013)]
	Neuro-adaptive Control	[Gueaieb *et al.* (2007)] [Zhao and Cheah (2009); Lian *et al.* (2002)]
	Leader-Follower Architecture	[Kume *et al.* (2007)] [Sun and Mills (2002); Zhu (2005)]
	Visual Servoing Coordination	[Dauchez *et al.* (2005)] [Huebner *et al.* (2009); Maitin-Shepard *et al.* (2010)]
	Motion Planning	[Knepper *et al.* (2013); Barraquand and Ferbach (1994)] [Dogar *et al.* (2015); Bien and Lee (1992)] [Sezgin *et al.* (1997)] [Harada *et al.* (2014); Basile *et al.* (2012)]
	Formation Control	[Li *et al.* (2008); Sieber *et al.* (2013); Hirata *et al.* (2003)]
LfD	GMM - Bimanual Coordination	[Zöllner *et al.* (2004)]
	GMM with HMM	[Calinon *et al.* (2010)]
	DMP	[Ijspeert *et al.* (2003); Ernesti *et al.* (2012)] [Nemec and Ude (2012); Park *et al.* (2008)] [Tamosiunaite *et al.* (2011); Bitzer and Vijayakumar (2009)] [Harada *et al.* (2014); Matsubara *et al.* (2010)] [Matsubara *et al.* (2011); Kulvicius *et al.* (2012)] [Umlauft *et al.* (2014); Thota *et al.* (2016)]
	GMM - Multivariate Robot Motions	[Gribovskaya *et al.* (2010)]
	Iterative Learning Control	[Van Den Berg *et al.* (2010)]
	GMM with Lyapunov	[Khansari-Zadeh and Billard (2014)]
	NN with Contraction	[Ravichandar and Dani (2015); Ravichandar *et al.* (2016)]
	GMM with Contraction	[Ravichandar *et al.* (2017); Ravichandar and Dani (2018)]

Apart from DMP, there are other LfD approaches that learns the end-effector motion trajectories by considering a joint state in the workspace ([Gribovskaya *et al.* (2010); Khansari-Zadeh and Billard (2014)]). In our prior work [Ravichandar and Dani (2015); Ravichandar *et al.* (2017); Ravichandar and Dani (2018)], the dynamics of reaching motions are learned using neural networks (NN) and Gaussian Mixture Models (GMMs) under contraction analysis constraints. In [Ravichandar *et al.* (2016)], this work is extended to periodic motions. For bimanual coordination, an LfD approach is developed in [Zöllner *et al.* (2004)] to classify bimanual motions based on the spatial relationship between the motion trajectories of both the arms. In [Calinon *et al.* (2010)], Hidden Markov Models (HMMs) and Gaussian Mixture Regression (GMR) are used to learn bi-manual tasks from several demonstrations. While learning-based methods generally perform well in circumstances similar to those encountered during training, they might fail in unfamiliar circumstances. In

[Van Den Berg *et al.* (2010)], an iterative learning controller for two-handed tying a knot is designed. LfD methods using non-rigid registration for bimanual manipulation tasks are presented in [Schulman *et al.* (2016)]. In Table 7.1, a classification of contributions related to coordinated manipulation that appeared in the literature based on approaches and methods is provided.

Contributions

The contributions of this chapter are as follows.

- Provide an overview of literature for bimanual manipulation.
- Design the control laws for desired trajectory tracking and synchronization of multi-agent system modeled with DMPs by using contraction analysis. The contraction analysis provides an exponential convergence to the desired trajectory, which is important for maintaining accuracy and productivity in the manufacturing tasks using robots.
- Demonstrate robustness of synchronization laws under perturbations which is very important for dual arm manipulations. When perturbed all the agents first synchronize and then resume tracking the desired trajectory, which is important in tasks such as manipulation with objects in workspace with two arms. The robustness property is important to prevent carried object from being dropped while performing bimanual heavy load transportation task.
- Specific control laws and constraints for tracking and synchronization are derived for performing the bimanual task. This property is useful for performing bimanual coordination tasks such as carrying a rigid object using two arms, or tying a knot or stripping a wire using two arms of a robot.
- Simulation and experimental results are presented which shows the tracking and synchronization, robustness of synchronization under perturbations. An example of moving wooden flanks is presented, where two arms must synchronize their motion with a rigid relative transformation. This experiment is motivated from many manufacturing operations that are found in industrial setting such as carrying large and heavy payload. Another experiment for tying a knot is presented, where two arms synchronize their motion with a reflective transformation between them. This experiment is motivated from manufacturing operations such as wire stripping operation or straightening of curled wires in wire harness assembly, tying knots in medical robotics applications.

Rest of the chapter is organized as follows. In Section 7.2, relevant preliminary mathematical concepts are revisited. In Section 7.3, modeling of multi-agent system represented using a system of DMPs and corresponding control design is presented. In Section 7.4, analyses of control design for tracking and synchronization control objective are provided. In Section 7.4.3, coordinate transformation between desired

behavior of multiple agents is provided that can be used to generate desired behavior of agents given behavior of one of the agents. In Section 7.5, results from two experiments are provided that implements the multi-agent control framework for bimanual manipulation task using Baxter robot.

7.2 Preliminaries

In this section, brief review of dynamic movement primitives and contraction analysis is presented.

7.2.1 *Review of Dynamic Movement Primitive*

The DMP consists of a transformation system which drives the system to the goal location, and a canonical system which drives the forcing term that generates the desired shape of the trajectory.

The DMP that encodes both rhythmic and transient motions is given by

$$\ddot{y}(t) = \Omega^2 \left(\alpha_y \left(\beta_y \left(g - y\left(t\right) \right) - \frac{\dot{y}\left(t\right)}{\Omega} \right) + f\left(\phi\left(t\right), r\left(t\right), \tilde{w}, w \right) \right) \tag{7.1}$$

where y is the state of the system in (7.1), $g \in \mathbb{R}$ is the goal location, $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^M \times \mathbb{R}^N \rightarrow \mathbb{R}$ is the forcing term to generate the desired behavior, where M and N are the number of basis functions encoding the transient and periodic motion respectively, $\tilde{w} \in \mathbb{R}^M$, $w \in \mathbb{R}^N$ are the weights of the basis functions, such that system in (7.1) represents the desired trajectory closely, $\alpha_y \in \mathbb{R}^+, \beta_y \in \mathbb{R}^+$ are the constant gains driving the system towards the goal location. In the following sub-sections, the forcing term $f(\cdot)$, the canonical system that drives the forcing term, the set of basis functions used to encode transient and periodic behavior are presented. Further, the construction of $f(\cdot)$ term for encoding discrete motions, i.e., point-to-point reaching is also described.

7.2.1.1 *Forcing Term $f(\cdot)$ for Periodic Motions*

The forcing term $f(\cdot)$ in (7.1), driven by the canonical system is obtained by using the following formula

$$f(\phi(t), r(t), \tilde{w}, w) = \frac{\sum_{j=1}^{M} \psi_j(\phi(t), r(t))\tilde{w}_j + \sum_{i=1}^{N} \zeta_i(\phi(t), r(t))w_j}{\sum_{j=1}^{M} \psi_j(\phi(t), r(t)) + \sum_{i=1}^{N} \zeta_i(\phi(t), r(t))}, \tag{7.2}$$

where $\psi : \mathbb{R}\times\mathbb{R} \rightarrow \mathbb{R}$, and $\zeta : \mathbb{R}\times\mathbb{R} \rightarrow \mathbb{R}$ are the basis functions. While $\psi_j(\phi(t), r(t))$ are the basis functions encoding the transient part of the motion, the basis functions $\zeta_i(\phi(t), r(t))$ are 2π-periodic in the first argument, and encode the periodic pattern as described in [Ernesti *et al.* (2012)].

Canonical System The canonical system for a DMP that encodes both rhythmic and transient motions is an oscillator in the phase plane given by

$$\begin{cases} \dot{\phi}\left(t\right) & = \Omega, & \phi\left(0\right) = \phi_0 \\ \dot{r}\left(t\right) & = \eta\left(\mu^{\alpha} - \left(r\left(t\right)\right)^{\alpha}\right)\left(r\left(t\right)\right)^{\beta}, & r\left(0\right) = r_0 \end{cases} \tag{7.3}$$

where $r(t) \in \mathbb{R}$ is the distance from center of limit cycle, $\phi(t) \in \mathbb{R}$ is the phase, $\alpha \in \mathbb{R}^+$, $\beta \in \mathbb{R}^+$ are constants, $\eta \in \mathbb{R}^+$ is gain, $\Omega \in \mathbb{R}^+$ defined as $\Omega = 2\pi/p$ is the frequency of execution, $p \in \mathbb{R}^+$ is the period of rhythmic movement in seconds and $\mu \in \mathbb{R}^+$ is the radius of the limit cycle.

Basis Functions The set of basis functions $\psi_j(\phi(t), r(t)),\ j = 1, 2, ..., M$ are used to encode the non-periodic transient behavior and another set of basis function $\zeta_i(\phi(t), r(t)),\ i = 1, 2, ...N$ are used to encode the periodic behavior. This means that in the beginning of the movement, the system should be only affected by $\psi_j(\phi(t), r(t))$ while in the long run their impact vanishes and $\zeta_i(\phi(t), r(t))$ smoothly begin to take over the control of the system. Therefore, $\psi_j(\phi(t), r(t))$ vanish close to the limit cycle, i.e., there exists a $\mu_1 \in \mathbb{R}^+$ such that $\mu_1 > \mu$ and $\psi_j(\phi(t), r(t))|_{\mathcal{R}\times(0,\mu_1)} = 0$. After passing the limit μ_1, the $\zeta_i(\phi(t), r(t))$ dominates the control of the system. Hence, the condition $\zeta_i(\phi(t), r(t))|_{\mathcal{R}\times(0,\mu_1)} = 1$ holds. The value of μ_1 is a constant that can be chosen arbitrarily.

Since the movement should be smooth, there has to be a region where the supports of the $\zeta_i(\phi(t), r(t))$ and the $\psi_j(\phi(t), r(t))$ overlap. The time needed for the canonical oscillator to pass that region is called transient fading time t_f. To create the fading region, μ_2 is set to be greater than μ_1, forcing the supports of $\zeta_i(\phi(t), r(t))$ and $\psi_j(\phi(t), r(t))$ to at most overlap for $r(t) \in (\mu_1, \mu_2)$. Here, $\mu_2 \in \mathbb{R}^+$ is chosen such that the transient fading time, which the oscillator needs to converge from μ_2 to μ_1, is equal to t_f.

Encoding Periodic Movement The basis functions $\zeta_i(\phi(t), r(t))$ encode the periodic pattern and thus should vanish away from the limit cycle. Therefore, $\zeta_i(\phi(t), r(t)) = k(r(t))h_i(\phi(t))$ is composed of two functions, where, $h_i : \mathbb{R} \to \mathbb{R}$ is 2π-periodic and encodes the periodic pattern and the function $k : \mathbb{R} \to \mathbb{R}$ makes $\zeta_i(\phi(t), r(t))$ vanish away from the limit cycle. The two functions are given by

$$h_i(\phi(t)) = \exp(v_i(\cos(\phi(t) - c_i) - 1)) \tag{7.4}$$

$$k(r(t)) = \begin{cases} 1 & r(t) < \mu_1 \\ \left(1 - \left(\frac{r(t)-\mu_1}{\mu_2-\mu_1}\right)^3\right)^3 & \mu_1 < r(t) < \mu_2 \\ 0 & r(t) > \mu_2 \end{cases} \tag{7.5}$$

where, $v_i \in \mathbb{R}^+$ is the variance and $c_i \in \mathbb{R}$ is the mean of the basis function

Encoding Transient Movement The basis functions $\psi_j(\phi(t), r(t))$ are arranged on the phase plane away from the limit cycle. Similarly to the $\zeta_i(\phi(t), r(t))$, the basis functions $\psi_j(\phi(t), r(t))$ are composed of two functions: one for the actual encoding and another one for keeping them away from the limit cycle. Hence, in each function $\psi_j(\phi(t), r(t)) = a(r(t))b_j\left(\left\|\begin{bmatrix} r(t)\cos(\phi(t)) \\ r(t)\sin(\phi(t)) \end{bmatrix} - p_j\right\|_2\right)$ the function $a : \mathbb{R} \to \mathbb{R}$ ensures that the $\psi_j(\phi(t), r(t))$ are nonzero only away from the limit cycle.

The function $b : \mathbb{R} \to \mathbb{R}$ is a standard basis function which can be represented in the form of a Gaussian. Placing the norm difference into b_j leads to a radially symmetric function centered on $p_j \in \mathbb{R}^2$ on the phase plane with the variance $\tilde{v} \in \mathbb{R}$ given by

$$a\left(r\left(t\right)\right) = \begin{cases} 0 & r(t) < \mu_1 \\ (1 - (\frac{\mu_2 - r(t)}{\mu_2 - \mu_1})^3)^3 & \mu_1 < r(t) < \mu_2 \\ 1 & r(t) > \mu_2 \end{cases} \tag{7.6}$$

$$b_j\left(r\left(t\right)\right) = \exp\left(-\tilde{v}r(t)^2\right) \tag{7.7}$$

7.2.1.2 *Forcing Term $f(\cdot)$ for Discrete Movements*

In order to obtain a merely point attractor dynamics, the following nonlinear function $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^M \to \mathbb{R}$. In order to obtain a merely point attractor dynamics, the following nonlinear function $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R}^M \to \mathbb{R}$

$$f(x, v, \tilde{w}) = \frac{\sum_{i=1}^{M} \Psi_i \tilde{w}_i v}{\sum_{i=1}^{M} \Psi_i} \qquad \Psi_i = \exp(-h_i(x/g - c_i)^2) \tag{7.8}$$

can be used as a forcing term in (7.1) to generate the desired behavior, where M is the number of basis function encoding the discrete motion, $\tilde{w} \in \mathbb{R}^M$ are the weights of the basis functions, and x, v are derived from the following canonical dynamical system that is selected to be a second order dynamical system

$$\dot{v} = \Omega\alpha_v(\beta_v(g - x) - v) \qquad \dot{x} = \Omega v \tag{7.9}$$

similar to (7.1) without the forcing term and thus its monotonic global convergence to g can be guaranteed with the proper choice of constant gains α_v, β_v [Ijspeert *et al.* (2003)]. The high-level design parameters of the discrete system are Ω, the temporal scaling factor, and g the goal position. Depending on degrees of freedom associated with each manipulator, multiple DMPs are needed to represent the dynamics.

7.2.1.3 *Learning the Forcing Term*

The weights $\tilde{w}, w$ are learned such that the system in (7.1) reproduces the demonstrated trajectory $y_{\text{demo}}(t)$. The required forcing term is computed by rearranging (7.1) as follows

$$f_{\text{target}}(t) = \frac{\ddot{y}_{\text{demo}}(t)}{\Omega^2} - \alpha_y \left(\beta_y (g - y_{\text{demo}}(t)) - \frac{\dot{y}_{\text{demo}}(t)}{\Omega} \right) \tag{7.10}$$

The weights in the forcing term are learned using linear regression tools. Also, the attractor point for all the periodic parts is calculated as $g = \frac{1}{L - L_{\text{trans}}} \left(\sum_{L_{\text{trans}+1}}^{L} y_k(t) \right)$, where $L_{\text{trans}} \in \mathbb{R}^+$ is the time duration of transient motion and $L \in \mathbb{R}^+$ is time duration of the task.

7.2.2 *Brief Review of Contraction Analysis*

In this section, contraction analysis tool that is used to analyze the stability of the derived controller is reviewed. Consider a system of the form $\dot{x}(t) = f_{\text{dyn}}(x(t))$, where $f_{\text{dyn}} : \mathbb{R}^p \to \mathbb{R}^p$ is a nonlinear vector function and $x(t) \in \mathbb{R}^p$ is a state vector. Any trajectory which starts in a ball of constant radius centered about a given trajectory and contained at all times in a contraction region, remains in that ball and converges exponentially to this trajectory [Lohmiller and Slotine (1998)]. Also, a region of the state space is called a contraction region with respect to a symmetric and uniformly positive definite metric $M(x,t) = \Theta(x,t)^T \Theta(x,t)$, if

$$\forall x, \left[\frac{\partial f_{\text{dyn}}^T}{\partial x} M(x,t) + M(x,t) \frac{\partial f_{\text{dyn}}}{\partial x} + \dot{M}(x,t) \right] \leq -\beta M(x,t) \tag{7.11}$$

in that region, where $\beta \in \mathbb{R}^+$ is the contraction rate and $\Theta(x,t)$ is a square matrix.

Contraction analysis will be used in determining the gains of the closed-loop system of DMPs with feedback (discussed in later sections) such that the overall system is contracting.

Theorem 7.1. *(Theorem 3 of [Wang and Slotine (2005)]) Synchronization in two way coupling configuration. Consider two coupled systems. If the dynamics equations verify $\dot{x}_1 - h(x_1,t) = \dot{x}_2 - h(x_2,t)$, where the function $h(\cdot)$ is contracting, then x_1 and x_2 will converge to each other exponentially, regardless of the initial conditions.*

Proof. See [Wang and Slotine (2005)]. □

7.3 Multi-agent System of Systems Modeling and Control Design

Let $q(t) \in \mathbb{R}^n$ be a state vector given by $q(t) = [\, y_1(t), y_2(t), \ldots y_n(t)]^T$, where the dynamics of each $y_i(t), i = 1, 2, .., n$ is represented using a single DMP. A set of six DMPs can be used to represent the end effector positions and the end effector orientations in the form of Euler angles or a set of 7 DMPs can be used to represent joint motion of a 7 degree of freedom (DoF) Baxter or ABB's YuMi robots.

For a given agent (e.g. Robot arm), there can be multiple DMPs each associated with a DOF. An agent with n - DMPs can be represented as follows

$$\begin{aligned}\ddot{y}_i(t) = &-\Omega\alpha_{yi}\dot{y}_i(t) - \Omega^2\alpha_{yi}\beta_{yi}y_i(t) + \Omega^2\alpha_{yi}\beta_{yi}g_i \\ &+ \Omega^2 f_i(\phi, r) + u_i(t), \quad i = 1, 2, .., n. \end{aligned} \tag{7.12}$$

where $u_i\,(t) \in \mathbb{R}$ is the external control input. From (7.12), it can be inferred that in a system with m agents, an i^{th} agent can be represented for $i = 1, 2 \cdots m$ as follows

$$\ddot{q}_i(t) = -K_a\dot{q}_i(t) - K_b q_i(t) + K_c + U_i(t), \tag{7.13}$$

where $K_a \in R^{n\times n}$, $K_b \in R^{n\times n}$, and $K_c \in \mathbb{R}^{n\times n}$ are given by $K_a = \begin{bmatrix} \Omega\alpha_{y1} & 0 & \cdots & 0 \\ 0 & \Omega\alpha_{y2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \Omega\alpha_{yn} \end{bmatrix}$, $K_c = \begin{bmatrix} \Omega^2\alpha_{y1}\beta_{y1}g_1 + \Omega^2 f_1(\phi, r) \\ \Omega^2\alpha_{y2}\beta_{y2}g_2 + \Omega^2 f_2(\phi, r) \\ \vdots \\ \Omega^2\alpha_{yn}\beta_{yn}g_n + \Omega^2 f_n(\phi, r) \end{bmatrix}$, $K_b = \begin{bmatrix} \Omega^2\alpha_{y1}\beta_{yn} & 0 & \cdots & 0 \\ 0 & \Omega^2\alpha_{y2}\beta_{yn} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \Omega^2\alpha_{yn}\beta_{yn} \end{bmatrix}$, and $U_i(t) = \begin{bmatrix} u_{i1} & u_{i2} & \cdots & u_{in} \end{bmatrix}^T$.

7.3.1 *Control Design*

The control input $U_i(t) \in \mathbb{R}^{n\times 1}$ is applied to an agent to alter its dynamics to achieve the requirements of tracking and synchronization. The control input in (7.13) is designed for $i = 1, 2 \cdots m$ as follows

$$\begin{aligned} U_i(t) =&\ddot{q}_{i,r}(t) + K_a\dot{q}_{i,r}(t) + K_b q_i(t) + K_1\left[\dot{q}_i(t) - \dot{q}_{i,r}(t)\right] \\ &+ \sum_{j\in N(i)} K_2\left[\dot{q}_j(t) - \dot{q}_{j,r}(t)\right] - K_c, \end{aligned} \tag{7.14}$$

where the gains $K_1 \in \mathbb{R}^{n\times n}$, $K_2 \in \mathbb{R}^{n\times n}$, and $D \in \mathbb{R}^{n\times n}$ are positive definite diagonal matrices, $N(i)$ represents the agents coupled with i^{th} agent in a system of m agents, and $\dot{q}_{i,r}(t) = \dot{q}_d(t) + D\left(q_d(t) - q_i(t)\right)$ with $q_d(t)$ and $\dot{q}_d(t)$ representing the desired trajectory and its velocity term respectively. $q_d\,(t)$ is generated using the DMPs learned from the demonstration data.

After substituting (7.14) in (7.13), the resulting closed-loop dynamics of the system are given by

$$\ddot{q}_i(t) - \ddot{q}_{i,r}(t) = -K_a\left[\dot{q}_i(t) - \dot{q}_{i,r}(t)\right] + K_1\left[\dot{q}_i(t) - \dot{q}_{i,r}(t)\right] + \sum_{j\in N(i)} K_2\left[\dot{q}_j(t) - \dot{q}_{j,r}(t)\right]. \tag{7.15}$$

In a system of m agents, let the tracking error $e_i \in \mathbb{R}^{n\times 1}$ of i^{th} agent be defined as $e_i(t) = \dot{q}_i(t) - \dot{q}_{i,r}(t) = \dot{q}_i(t) - \dot{q}_d(t) + D\left[q_i(t) - q_d(t)\right]$. Based on (7.15), the error dynamics of the i^{th} agent can be computed as follows

$$\dot{e}_i(t) = -K_a e_i(t) + K_1 e_i(t) + \sum_{j\in N(i)} K_2 e_j(t). \tag{7.16}$$

7.4 Analysis of Multi-agent Tracking Control and Synchronization

In this section, gain designs for tracking control and synchronization of multi-agent systems with cyclic and all-to-all configurations are presented.

7.4.1 *Tracking Control Analysis*

Cyclic Configuration For the system in (7.16) with cyclic configuration of coupling, the error dynamics are represented by

$$\dot{E}(t) = -L_c E(t), \tag{7.17}$$

where $E(t) \in R^{mn\times 1}$ and $L_c \in \mathbb{R}^{mn\times mn}$ are given by $E(t) = \left[e_1(t), e_2(t), \ldots, e_m(t)\right]^T$ and

$$L_c = \begin{bmatrix} K_a - K_1 & -K_2 & 0 & . & 0 & -K_2 \\ -K_2 & K_a - K_1 & -K_2 & & & 0 \\ 0 & & . & & & . \\ . & & & . & & 0 \\ 0 & & & -K_2 & K_a - K_1 & -K_2 \\ -K_2 & 0 & . & 0 & -K_2 & K_a - K_1 \end{bmatrix},$$

respectively.

For the system in (7.17), $J = \frac{\partial(-L_c E(t))}{\partial E(t)} = -L_c$ and L_c is a symmetric matrix. With $J = -L_c$ and $J^T = -L_c$ and identity matrix as the positive definite metric $M(x)$, the condition in (7.11) can be written as $-L_c \leq -\frac{\beta I_{mn\times mn}}{2}$ which in turn yields

$$L_c \geq \gamma I_{mn\times mn}, \tag{7.18}$$

where $\gamma = \frac{\beta}{2}$. By satisfying the condition in (7.18), the system in (7.17) is contracting with the rate of β. Based on (7.18), the conditions for designing gains for $m = 2$ are derived first. Then, a generalized case with $m \geq 3$ is shown

Systems with Two Agents

For a system in (7.17), with $m = 2$ agents, the error dynamics representation is as follows

$$\begin{bmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \end{bmatrix} = - \begin{bmatrix} K_a - K_1 & -K_2 \\ -K_2 & K_a - K_1 \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix}.$$

Following the result of m agent system, $K_a - K_1 > 0$, and $K_2 > 0$, the conditions for selecting gains in this case are given by $K_a - K_1 + K_2 \geq \gamma_1 I_{n\times n}$ and $K_a - K_1 - K_2 \geq \gamma_1 I_{n\times n}$, where $\gamma_1 \in \mathbb{R}^+$ for the system in (7.17) with two agents to be contracting.

Systems with More than Two Agents

Given $K_a - K_1 > 0$ and $K_2 > 0$, it can be observed that for a system in (7.17) with $m \geq 3$ in order to satisfy the condition in (7.18), the smallest eigenvalue of matrix L_c should be greater than γ

$$\lambda_{\min}(L_c) \geq \gamma \tag{7.19}$$

Therefore, by designing the gains $K_a, K_1,$ and K_2 such that $K_a - K_1 - 2K_2 \geq \gamma I_{n\times n}$ the condition in (7.19) is satisfied which in turn ensures that the system in (7.17) is contracting.

Example 7.1. $(m = 5)$

Similarly, for the system in (7.17) with 5 agents, the error dynamics representation is as follows

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \\ \dot{e}_5 \end{bmatrix} = - \begin{bmatrix} K_a - K_1 & -K_2 & 0 & 0 & -K_2 \\ -K_2 & K_a - K_1 & -K_2 & 0 & 0 \\ 0 & -K_2 & K_a - K_1 & -K_2 & 0 \\ 0 & 0 & -K_2 & K_a - K_1 & -K_2 \\ -K_2 & 0 & 0 & -K_2 & K_a - K_1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix}$$

In order to satisfy (7.19), the necessary conditions are given by $K_a - K_1 + 2K_2 \geq \gamma_2 I_{n\times n}$, $K_a - K_1 - \frac{K_2}{2} - \frac{\sqrt{5}K_2}{2} \geq \gamma_2 I_{n\times n}$, $K_a - K_1 - \frac{K_2}{2} - \frac{\sqrt{5}K_2}{2} \geq \gamma_2 I_{n\times n}$, $K_a - K_1 - \frac{K_2}{2} + \frac{\sqrt{5}K_2}{2} \geq \gamma_2 I_{n\times n}$, and $K_a - K_1 - \frac{K_2}{2} + \frac{\sqrt{5}K_2}{2} \geq \gamma_2 I_{n\times n}$, where $\gamma_2 \in \mathbb{R}^+$. It can be observed that designing gains such that $K_a - K_1 + K_2 \geq \gamma_2 I_{n\times n}$ is sufficient to satisfy the condition in (7.19).

All-to-all Configuration Considering a system with m agents with each agent's error dynamics represented as in (7.16) and all-to-all configuration of coupling, its error dynamics are given by

$$\dot{E}(t) = -L_a E(t), \tag{7.20}$$

where $E(t) = [\, e_1(t), e_2(t), \ldots, e_m(t)\,]^T$ and

$$L_a = \begin{bmatrix} K_a - K_1 & -K_2 & . & . & . & -K_2 \\ -K_2 & K_a - K_1 & -K_2 & & & . \\ . & & . & & & . \\ . & & & . & & . \\ . & & & -K_2 & K_a - K_1 & -K_2 \\ -K_2 & . & . & . & -K_2 & K_a - K_1 \end{bmatrix}.$$

With $J = \frac{\partial(-L_a E(t))}{\partial E(t)} = -L_a$ and $J^T = -L_a$ and identity matrix as the positive definite metric $M(x)$, the condition in (7.11) can be written as $-L_a \leq -\frac{\beta I_{mn\times mn}}{2}$ which in turn yields

$$L_a \geq \gamma I_{mn\times mn}. \tag{7.21}$$

Given $K_a - K_1 > 0$ and $K_2 > 0$, it can be observed that for a system in (7.20) in order to satisfy the condition in (7.21), the smallest eigenvalue of matrix L_a should be greater than γ.

$$\lambda_{\min}(L_a) \geq \gamma \tag{7.22}$$

Therefore, by designing the gains $K_a, K_1,$ and K_2 such that $K_a - K_1 - ((m-1)\times K_2) \geq \gamma I_{n\times n}$ the condition in (7.22) is satisfied which in turn ensures that the system in (7.20) is contracting.

Example 7.2. $(m = 4)$

For the system in (7.20) with 4 agents, the error dynamics representation is as follows

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{bmatrix} = - \begin{bmatrix} K_a - K_1 & -K_2 & -K_2 & -K_2 \\ -K_2 & K_a - K_1 & -K_2 & -K_2 \\ -K_2 & -K_2 & K_a - K_1 & -K_2 \\ -K_2 & -K_2 & -K_2 & K_a - K_1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}.$$

In order to satisfy (7.22), the necessary conditions are given by $K_a - K_1 + K_2 \geq \gamma_3 I_{n\times n}$, $K_a - K_1 + K_2 \geq \gamma_3 I_{n\times n}$, $K_a - K_1 + K_2 \geq \gamma_3 I_{n\times n}$, and $K_a - K_1 - 3K_2 \geq \gamma_3 I_{n\times n}$, where $\gamma_3 \in \mathbb{R}^+$. It can be observed that designing gains such that $K_a - K_1 - 3K_2 \geq \gamma_3 I_{n\times n}$ is sufficient to satisfy the condition in (7.22).

7.4.2 *Synchronization Analysis*

Cyclic Configuration Considering the system in (7.17) with $m = 2$ agents, the error dynamics of the first agent can be written as $\dot{e}_1(t) = -K_a e_1(t) + K_1 e_1(t) + K_2 e_2(t)$. By adding $K_2 e_1(t)$ on both sides, the error dynamics of the first agent can be reformulated as

$$\dot{e}_1(t) - h(e_1(t)) = g(e_1(t),\ e_2(t)), \tag{7.23}$$

where $h(e_1(t)) = -(K_a - K_1 + K_2)\, e_1(t)$ and $g(e_1(t),\ e_2(t)) = K_2(e_1(t) + e_2(t))$. Similarly, the error dynamics for the second agent are given by

$$\dot{e}_2(t) - h(e_2(t)) = g(e_1(t),\ e_2(t)). \tag{7.24}$$

Based on Theorem 3 of [Wang and Slotine (2005)], (7.23), and (7.24), if $(K_a - K_1 + K_2) < 0$ the trajectories of the error dynamics of both the agents will always converge to each other exponentially, even under spatial perturbations. Similarly, for m=3 agents, equations similar to (7.23) and (7.24) can be formulated to prove the synchronization. Note that for $m > 3$ agents, all the agents are no longer coupled with each other. The synchronization can be proved by using the matrix decomposition approach in [Chung and Slotine (2009)].

All-to-all Configuration Considering the system in (7.20), the error dynamics of an i^{th} agent can be written as $\dot{e}_i(t) = -K_a e_i(t) + K_1 e_i(t) + \sum_{j=1, j\neq i}^{m} K_2 e_j(t)$. By adding $K_2 e_i(t)$ on both sides, the error dynamics of the first agent can be reformulated as

$$\dot{e}_i(t) - h(e_i(t)) = g(e_1(t),\ e_2(t), ...,\ e_m(t)), \tag{7.25}$$

where $h(e_i(t)) = -(K_a - K_1 + K_2)\, e_i(t)$ and $g(e_1(t),\ e_2(t), ...,\ e_m(t)) = K_2(\sum_{i=1}^{m} e_i(t))$. Similarly, the error dynamics for $i+1^{th}$ agent are given by

$$\dot{e}_{i+1}(t) - h(e_{i+1}(t)) = g(e_1(t),\ e_2(t), ...,\ e_m(t)). \tag{7.26}$$

Based on Theorem 3 of [Wang and Slotine (2005)], (7.25), and (7.26), if $(K_a - K_1 + K_2) < 0$, then the trajectories of the error dynamics of any two agents will always converge to each other exponentially, even under spatial perturbations.

7.4.3 *Coordinate Transformation between Agents for Bimanual Manipulation*

The bimanual manipulation can be considered as the system with two agents (right and left arms). While implementing the learned bimanual task, such as transporting a box and moving it to a new goal location, learning the dynamics of one arm motion is sufficient. The synchronization and tracking control laws developed in previous section can be used to obtain the motion of two arms in task space. The desired trajectory for the second arm can be obtained by transforming the states of synchronized second arm's task space motion by a desired fixed or variable transformation.

Suppose that ${}^{\mathcal{R}}\xi_{\mathcal{B}}(t)$ and ${}^{\mathcal{L}}\xi_{\mathcal{B}}(t)$ are the task space poses of right arm and left arm end effectors with respect to the robot body reference frame at any given time t. Also, let ${}^{\mathcal{L}}\xi_{\mathcal{R}}(t)$ be the transformation of the right arm end effector with respect to left arm end effector, then the relationship between ${}^{\mathcal{R}}\xi_{\mathcal{B}}(t)$ and ${}^{\mathcal{L}}\xi_{\mathcal{B}}(t)$ is given by ${}^{\mathcal{L}}\xi_{\mathcal{B}}(t) = {}^{\mathcal{L}}\xi_{\mathcal{R}}(t)\ {}^{\mathcal{R}}\xi_{\mathcal{B}}(t)$. Additionally, if the object being manipulated is rigid, then the transformation between the two arms' end effectors is fixed, i.e., ${}^{\mathcal{L}}\xi_{\mathcal{B}}(t) = {}^{\mathcal{L}}\xi_{\mathcal{R}}\ {}^{\mathcal{R}}\xi_{\mathcal{B}}(t)$. In the case of rigid object manipulation, the constant transformation ${}^{\mathcal{L}}\xi_{\mathcal{R}}$ is chosen according to the experimental setup. If the object being manipulated is deformable, then the transformation between the two arms' end effectors is time varying, e.g., a mirror transformation for a two string pulling portion of knot tying operation.

Let $\Psi \subset \mathbb{R}^6$ be the space which satisfies the robot's end effector position and orientation restrictions, $q_{\mathcal{R}}(t) \in \Psi$ and $q_{\mathcal{L}}(t) \in \Psi$ represent the end effector positions and the end effector orientations in the form of Euler angles for right and left arms respectively. Let $f_{\text{trans}} : SE(3) \to \Psi$ be a function that computes the end effector position and orientation in the form of Euler angles given pose and $f_{\text{trans}}^{-1} : \Psi \to SE(3)$ computes pose given end effector position and orientation. The right arm end effector dynamics are learned using DMPs with $q_R(t) = f_{\text{trans}}\left({}^{\mathcal{R}}\xi_{\mathcal{B}}(t)\right)$ as the states. Consider a virtual agent with pose ${}^{\tilde{\mathcal{L}}}\xi_{\mathcal{B}}(t)$ such that

$$ {}^{\mathcal{L}}\xi_{\mathcal{B}}(t) = {}^{\mathcal{L}}\xi_{\mathcal{R}}(t)\ {}^{\tilde{\mathcal{L}}}\xi_{\mathcal{B}}(t). \tag{7.27} $$

The virtual agent's dynamics are represented using the same learned DMPs with $q_{\tilde{L}}(t) = f_{\text{trans}}\left({}^{\tilde{\mathcal{L}}}\xi_{\mathcal{B}}(t)\right)$ as states. By designing the control law as described in Section 7.4, $q_R(t)$ and $q_{\tilde{L}}(t)$ will converge to each other and track the desired trajectory. The left arm end effector pose ${}^{\mathcal{L}}\xi_{\mathcal{B}}(t)$, is obtained by using ${}^{\tilde{\mathcal{L}}}\xi_{\mathcal{B}}(t) = f_{\text{trans}}^{-1}(q_{\tilde{L}}(t))$ and (7.27).

7.5 Experimental Results

Three experiments involving the bimanual manipulation tasks are performed using the learned DMPs and the control laws designed in Sections 7.4. The experiments are implemented using Robotics toolbox of [Corke (2011)] with MATLAB 2014b on a computer running a Intel i5 processor with 8 Gigabytes of memory. Kinesthetic demonstrations are obtained using the Baxter robot platform in order to learn the DMPs.

7.5.1 *Experiment 1*

A single kinesthetic demonstration of a bimanual manipulation task with a solid object as shown in Fig. 7.1 is obtained using the 7 degrees of freedom Baxter robot. The DH-parameters of the robot are used to obtain the end effector pose (position and orientation) of the robot arms during the task. To test the robustness of the

Algorithm 1: Learning and Synchronization of Movement Primitives for Bimanual Manipulation Tasks

1 Obtain a joint angle measurements of the robot through kinesthetic demonstrations while user performing the task by guiding the robot arms;
2 Obtain the end effector pose from join angle measurements using DH-parameters of the robot;
3 Obtain the position, velocity, and acceleration estimates of the pose of the end effector using Kalman filter with constant acceleration model;
4 Define the gains of the DMP attractor system and the canonical system;
5 Learn the forcing terms for the single arm end effector pose and obtain the weights of the basis functions in the forcing term;
6 Determine the gains of the closed-loop system such that the overall system in (7.17) or (7.20) is contracting;
7 Obtain the second arm's end effector pose by transforming the second agent in the system as shown in (7.27) for implementation on the robot;

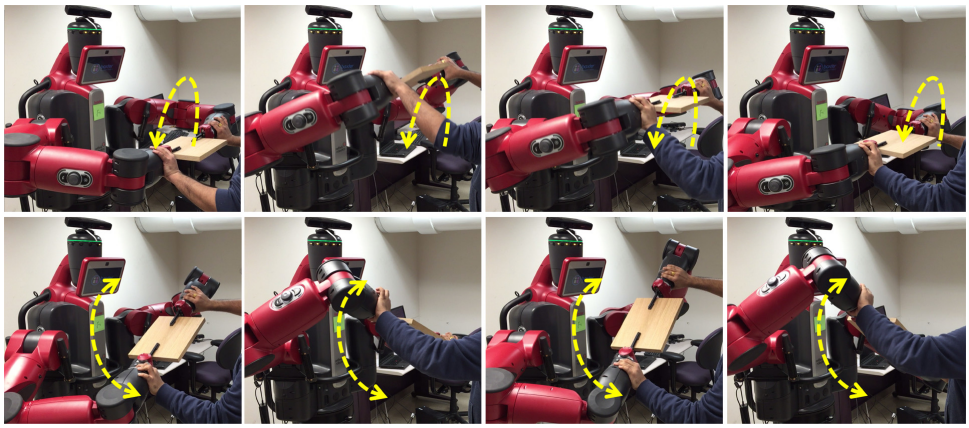

Fig. 7.1: Demonstrations obtained using the Baxter robot platform are shown in the figure. The pictures in the top row correspond to the transient motion at the beginning of the task and the pictures in the bottom row correspond to the subsequent periodic motion.

learned DMPs and the designed control laws, spatial perturbations are applied to one of the arms and the response of the system containing both the arms is observed. A DMP is learned in a single dimension, and due to the forward mapping between the joint-space ($\mathbb{R}^7$) and the end-effector space ($\mathbb{R}^6$), a set of 6 DMPs is used to learn the end-effector dynamics of one of the arms of the robot, as explained in Section 7.2.1. During the learning phase, the gains of the attractor system of each DMP are chosen to be $\alpha_y = 32$ and $\beta_y = 8$. The gains and constants used in

canonical system are chosen to be $\alpha = \frac{1}{6}$, $\beta = \frac{1}{1000}$, $\eta = 35$, $\mu = 1$, $\mu_1 = 1.2$, and $\mu_2 = 1.4$. These gains are chosen such that the DMP can accurately reproduce the demonstrated data. Also, as the task requires the robot to use two arms (i.e., two agents in the system (7.17)), the gains K_1, K_2, and D are selected to be $K_1 = 5I_{6\times6}$, $K_2 = 45I_{6\times6}$ and $D = 10I_{6\times6}$ in order to satisfy the contraction condition given in (7.18). The DMPs are used to represent the task space motion of one of the arms of the robot, and then the other arm's dynamics are obtained by transforming the learned DMPs according to (7.27).

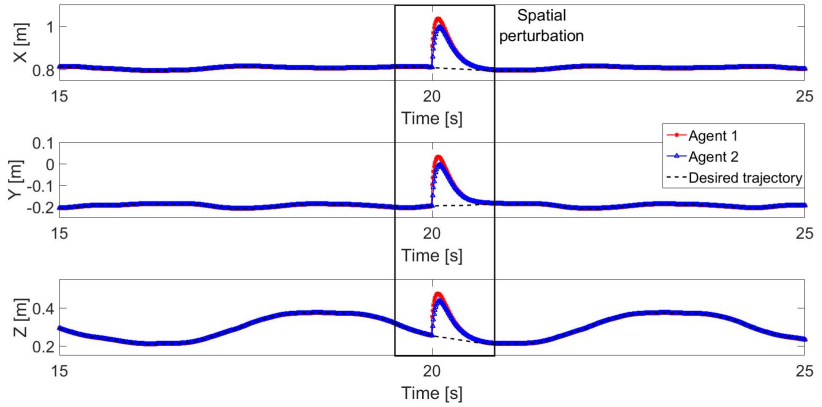

Fig. 7.2: Behaviors of both the arms when one of the arms is perturbed.

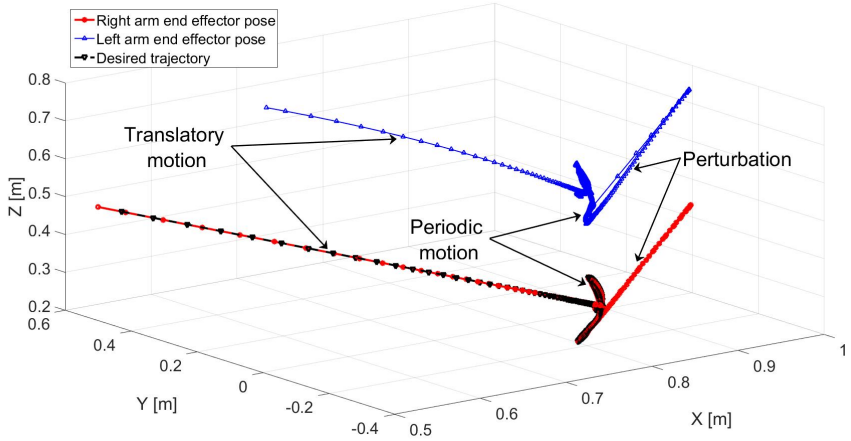

Fig. 7.3: 3D representation of the behaviors of both the arms when one of the arms is perturbed.

The response of the system to the perturbations is shown in Fig. 7.2. As one of the arms is perturbed, the other arm deviates from its desired trajectory in order to synchronize with the perturbed arm. It can be observed from Fig. 7.2 that the arms synchronize within 0.16 seconds and both the arms converge to the desired trajectory simultaneously. The 3D representation of both the arms' behaviors after the transformation is shown in Fig. 7.3.

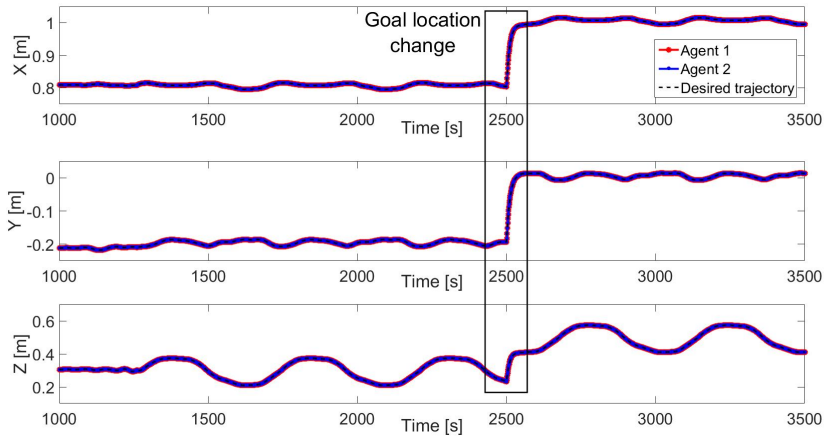

Fig. 7.4: Behaviors of both the arms when goal location is changed.

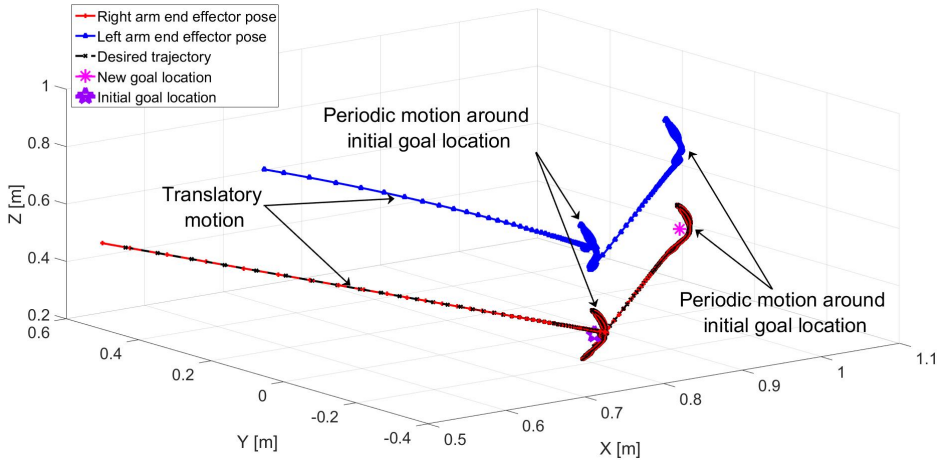

Fig. 7.5: 3D representation of the behaviors of both the arms when goal location is changed.

7.5.2 *Experiment 2*

In the second experiment, the adaption to the goal location changes while performing a task is demonstrated. The control gains are chosen to be of the same values as described in Experiment 1. The behaviors of the robot arms with the change in goal location are shown in Fig. 7.4. Both the arms are able to adapt to the change in goal location and carry out the task. The 3D representation of the behaviors of the arms is shown in Fig. 7.5.

7.5.3 *Experiment 3*

For the third experiment, tying an Overhand knot is considered. For simplicity, the two ends of a rope are assumed to be spliced together but not fully tightened. Baxter's two manipulators that are equipped with the parallel grippers, are used to grasp the two ends and pull them simultaneously in the opposite directions to tighten the knot. A single kinesthetic demonstration of one of the Baxter's arm pulling on one end of a rope is obtained. During the demonstration, the joints' trajectory of the manipulator that takes the end effector to the location where the rope is at a fully tightened state (goal location) is recorded by guiding the associated arm to the maximum length of the rope. The recorded angular position and velocity measurements performing the rope pulling task are then used to obtain the end effector pose and ultimately the position, velocity and acceleration estimates of the end effector pose. A set of 6 DMPs is used to learn the discrete movements of the rope pulling task using discrete movements as explained in 7.3. During the learning phase of the discrete movement, the gains of the attactor system of each DMP are chosen to be $\alpha_v = \alpha_y = 16$, $\beta_v = \beta_y = 8$. Also to satisfy the contraction condition in 7.11 the following gains are selected $K_1 = I_{6\times6}$, $K_2 = 0.5I_{6\times6}$ and $D = 10I_{6\times6}$. Once the DMPs of one arm are learned, the dynamics of the other arm are obtained by mirroring the points on the horizontal axis of the Baxter robot coordinate frames, namely y, by the vertical axis x while the other five dimensions (DMPs) remain fixed. The behaviors of the robot arms is depicted in Fig. 7.6, and can be observed that the arms synchronize and both converge to the goal location at the same time. Note that in Fig. 7.6 the trajectory of the second arm in y dimension is positive which implies that the second arm moves in the opposite direction from the first one.

Fig. 7.7 shows the proposed method's ability to successfully learn the motion dynamics of the bimanual manipulation task of tightening of an Overhand knot. As it is shown in Fig. 7.7 Baxter holds the two ends and simultaneously pulls them in the opposite direction to tighten the knot.

7.6 Conclusion

Literature review shows that the bimanual manipulation is an important problem for many manufacturing applications. The control laws to achieve tracking and

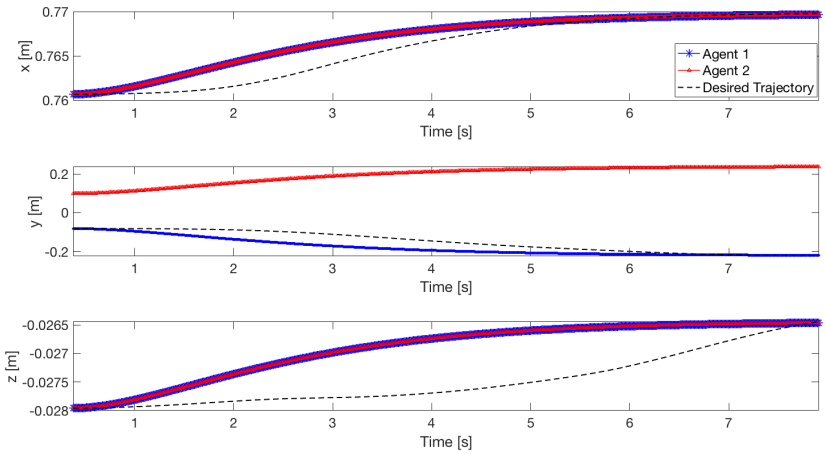

Fig. 7.6: Behavior of both the arms while tying an Overhand knot.

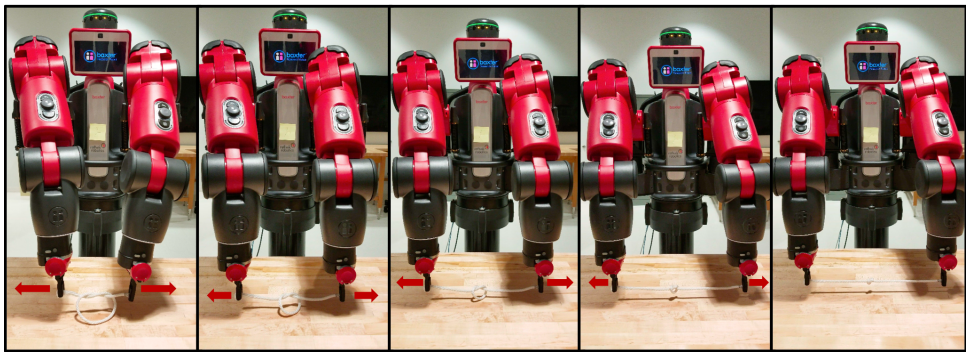

Fig. 7.7: Sequences of images showing the Baxter robot tying an Overhand knot using the tracking and synchronization controller.

synchronization of dual arm manipulator performing bimanual manipulation tasks are developed. The designed control laws are robust to perturbations and instantaneously adapt to changes in the goal location. The experimental results suggest that the agents of the system in (7.17) are able to synchronize within 0.16 seconds. Experiments by implementation of the controller on a robot shows the usefulness of the controllers on a real system. In future, demonstration of the controllers for a real manufacturing application such as wire harness assembly will be shown.

Acknowledgments

Research was sponsored by the Office of the Secretary of Defense and was accomplished under Agreement Number W911 NF-17-3-0004. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of the Secretary of Defense or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. The authors would like to thank Ryan Saltus for his help with the robot implementation.

Bibliography

Balkcom, D. J. and Mason, M. T. (2008). Robotic origami folding, *The International Journal of Robotics Research* **27**, 5, pp. 613–627.

Bandyopadhyay, S., Chung, S.-J., and Hadaegh, F. Y. (2017). Probabilistic and distributed control of a large-scale swarm of autonomous agents, *IEEE Transactions on Robotics* **33**, 5, pp. 1103–1123.

Barraquand, J. and Ferbach, P. (1994). A penalty function method for constrained motion planning, in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on* (IEEE), pp. 1235–1242.

Basile, F., Caccavale, F., Chiacchio, P., Coppola, J., and Curatella, C. (2012). Task-oriented motion planning for multi-arm robotic systems, *Robotics and Computer-Integrated Manufacturing* **28**, 5, pp. 569–582.

Bien, Z. and Lee, J. (1992). A minimum-time trajectory planning method for two robots, *IEEE Transactions on Robotics and Automation* **8**, 3, pp. 414–418.

Bitzer, S. and Vijayakumar, S. (2009). Latent spaces for dynamic movement primitives, in *9th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 574–581.

Bonitz, R. and Hsia, T. C. (1996a). Internal force-based impedance control for cooperating manipulators, *IEEE Transactions on Robotics and Automation* **12**, 1, pp. 78–89.

Bonitz, R. G. and Hsia, T. C. (1996b). Robust dual-arm manipulation of rigid objects via palm grasping-theory and experiments, in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, Vol. 4 (IEEE), pp. 3047–3054.

Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., and Billard, A. G. (2010). Learning and reproduction of gestures by imitation, *IEEE Robotics & Automation Magazine* **17**, 2, pp. 44–54.

Chung, S.-J., Bandyopadhyay, S., Chang, I., and Hadaegh, F. Y. (2013). Phase synchronization control of complex networks of lagrangian systems on adaptive digraphs, *Automatica* **49**, 5, pp. 1148–1161.

Chung, S.-J. and Slotine, J.-J. E. (2009). Cooperative robot control and concurrent synchronization of lagrangian systems, *IEEE Transactions on Robotics* **25**, 3, pp. 686–700.

Chwa, D., Dani, A. P., and Dixon, W. E. (2016). Range and motion estimation of a monocular camera using static and moving objects, *IEEE Transactions on Control Systems Technology* **24**, 4, pp. 1174–1183, doi:10.1109/TCST.2015.2508001.

Corke, P. I. (2011). *Robotics, Vision & Control: Fundamental Algorithms in Matlab* (Springer), ISBN 978-3-642-20143-1.

Dani, A. P. and Dixon, W. E. (2010). Single camera structure and motion estimation, in

Visual Servoing via Advanced Numerical Methods (Springer), pp. 209–229.

Dani, A. P., Fischer, N. R., and Dixon, W. E. (2012). Single camera structure and motion, *IEEE Transactions on Automatic Control* **57**, 1, pp. 238–243.

Dauchez, P., Fraisse, P., and Pierrot, F. (2005). A vision/position/force control approach for performing assembly tasks with a humanoid robot, in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on* (IEEE), pp. 277–282.

Dogar, M., Spielberg, A., Baker, S., and Rus, D. (2015). Multi-robot grasp planning for sequential assembly operations, in *IEEE International Conference on Robotics and Automation* (IEEE), pp. 193–200.

Doulgeri, Z. and Golfakis, A. (2006). Nonlinear manipulation control of a compliant object by dual fingers, *Journal of dynamic systems, measurement, and control* **128**, 3, pp. 473–481.

Erhart, S. and Hirche, S. (2013). Adaptive force/velocity control for multi-robot cooperative manipulation under uncertain kinematic parameters, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE), pp. 307–314.

Erhart, S., Sieber, D., and Hirche, S. (2013). An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE), pp. 315–322.

Ernesti, J., Righetti, L., Do, M., Asfour, T., and Schaal, S. (2012). Encoding of periodic and their transient motions by a single dynamic movement primitive, in *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 57–64.

Gans, N. R., Dani, A., and Dixon, W. E. (2008). Visual servoing to an arbitrary pose with respect to an object given a single known length, in *American Controls Conference* (Seattle, WA, USA), pp. 1261–1267.

Gribovskaya, E., Khansari-Zadeh, S. M., and Billard, A. (2010). Learning non-linear multivariate dynamics of motion in robotic manipulators, *The International Journal of Robotics Research* **30**, 1, pp. 80–117.

Gueaieb, W., Karray, F., and Al-Sharhan, S. (2007). A robust hybrid intelligent position/force control scheme for cooperative manipulators, *IEEE/ASME Transactions on Mechatronics* **12**, 2, pp. 109–125.

Harada, K., Tsuji, T., and Laumond, J.-P. (2014). A manipulation motion planner for dual-arm industrial manipulators, in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE), pp. 928–934.

Hirata, Y., Kume, Y., Wang, Z.-D., and Kosuge, K. (2003). Decentralized control of multiple mobile manipulators based on virtual 3-d caster motion for handling an object in cooperation with a human, in *IEEE International Conference on Robotics and Automation*, Vol. 1 (IEEE), pp. 938–943.

Huebner, K., Welke, K., Przybylski, M., Vahrenkamp, N., Asfour, T., Kragic, D., and Dillmann, R. (2009). Grasping known objects with humanoid robots: A box-based approach, in *Advanced Robotics, 2009. ICAR 2009. International Conference on* (IEEE), pp. 1–6.

Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors, *Neural Computation* **25**, 2, pp. 328–373.

Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2003). Learning attractor landscapes for learning motor primitives, in *Advances in neural information processing systems*, pp. 1547–1554.

Khansari-Zadeh, S. M. and Billard, A. (2014). Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions, *Robotics and Autonomous Systems* **62**, 6, pp. 752–765.

Knepper, R. A., Layton, T., Romanishin, J., and Rus, D. (2013). Ikeabot: An autonomous multi-robot coordinated furniture assembly system, in *IEEE International Conference on Robotics and Automation* (IEEE), pp. 855–862.

Kulvicius, T., Ning, K., Tamosiunaite, M., and Worgotter, F. (2012). Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting, *IEEE Transactions on Robotics* **28**, 1, pp. 145–157.

Kume, Y., Hirata, Y., and Kosuge, K. (2007). Coordinated motion control of multiple mobile manipulators handling a single object without using force/torque sensors, in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (IEEE), pp. 4077–4082.

Langsfeld, J. D., Kabir, A. M., Kaipa, K. N., and Gupta, S. K. (2016). Robotic bimanual cleaning of deformable objects with online learning of part and tool models, in *IEEE International Conference on Automation Science and Engineering* (IEEE), pp. 626–632.

Li, Z., Ge, S. S., and Wang, Z. (2008). Robust adaptive control of coordinated multiple mobile manipulators, *Mechatronics* **18**, 5-6, pp. 239–250.

Lian, K.-Y., Chiu, C.-S., and Liu, P. (2002). Semi-decentralized adaptive fuzzy control for cooperative multirobot systems with h/sup/spl infin//motion/internal force tracking performance, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **32**, 3, pp. 269–280.

Lohmiller, W. and Slotine, J.-J. E. (1998). On contraction analysis for non-linear systems, *Automatica* **34**, 6, pp. 683–696.

Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. (2010). Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding, in *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (IEEE), pp. 2308–2315.

Matsubara, T., Hyon, S.-H., and Morimoto, J. (2010). Learning stylistic dynamic movement primitives from multiple demonstrations, in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1277–1283.

Matsubara, T., Hyon, S.-H., and Morimoto, J. (2011). Learning parametric dynamic movement primitives from multiple demonstrations, *Neural Networks* **24**, 5, pp. 493–500.

Namiki, A. and Yokosawa, S. (2015). Robotic origami folding with dynamic motion primitives, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE), pp. 5623–5628.

Nemec, B. and Ude, A. (2012). Action sequencing using dynamic movement primitives, *Robotica* **30**, 05, pp. 837–846.

Park, D.-H., Pastor, P., Schaal, S., *et al.* (2008). Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields, in *8th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 91–98.

Ravichandar, H. and Dani, A. (2018). Learning pose dynamics from demonstrations via contraction analysis, *Autonomous Robots* **43**, 4, pp. 897–912, doi:https://doi.org/10.1007/s10514-018-9758-x.

Ravichandar, H. and Dani, A. P. (2015). Learning contracting nonlinear dynamics from human demonstration for robot motion planning, in *ASME Dynamic Systems and Control Conference (DSCC)*.

Ravichandar, H., Salehi, I., and Dani, A. (2017). Learning partially contracting dynamical systems from demonstrations, in *Proceedings of the 1st Annual Conference on Robot Learning, PMLR*, Vol. 78, pp. 369–378.

Ravichandar, H., Thota, P. K., and Dani, A. P. (2016). Learning periodic motions from human demonstrations using transverse contraction analysis, in *IEEE American*

Control Conference (ACC).

Sarkar, N., Yun, X., and Kumar, V. (1997). Dynamic control of 3-d rolling contacts in two-arm manipulation, *IEEE Transactions on Robotics and Automation* **13**, 3, pp. 364–376.

Schneider, S. A. and Cannon, R. H. (1992). Object impedance control for cooperative manipulation: Theory and experimental results, *IEEE Transactions on Robotics and Automation* **8**, 3, pp. 383–394.

Schulman, J., Ho, J., Lee, C., and Abbeel, P. (2016). Learning from demonstrations through the use of non-rigid registration, in *Robotics Research* (Springer), pp. 339–354.

Sezgin, U., Seneviratne, L. D., and Earles, S. (1997). Collision avoidance in multiple-redundant manipulators, *The International Journal of Robotics Research* **16**, 5, pp. 714–724.

Sieber, D., Deroo, F., and Hirche, S. (2013). Formation-based approach for multi-robot cooperative manipulation based on optimal control design, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE), pp. 5227–5233.

Smith, C., Karayiannidis, Y., Nalpantidis, L., Gratal, X., Qi, P., Dimarogonas, D. V., and Kragic, D. (2012). Dual arm manipulation - a survey, *Robotics and Autonomous Systems* **60**, 10, pp. 1340–1353.

Sun, D. and Mills, J. K. (2002). Adaptive synchronized control for coordination of multirobot assembly tasks, *IEEE Transactions on Robotics and Automation* **18**, 4, pp. 498–510.

Tamosiunaite, M., Nemec, B., Ude, A., and Wörgötter, F. (2011). Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives, *Robotics and Autonomous Systems* **59**, 11, pp. 910–922.

Thota, P. K., chaandar Ravichandar, H., and Dani, A. P. (2016). Learning and synchronization of movement primitives for bimanual manipulation tasks, in *IEEE 55th Conference on Decision and Control* (IEEE), pp. 945–950.

Tinós, R., Terra, M. H., and Ishihara, J. Y. (2006). Motion and force control of cooperative robotic manipulators with passive joints, *IEEE Transactions on Control Systems Technology* **14**, 4, pp. 725–734.

Umlauft, J., Sieber, D., and Hirche, S. (2014). Dynamic movement primitives for cooperative manipulation and synchronized motions, in *IEEE International Conference on Robotics and Automation* (IEEE), pp. 766–771.

Van Den Berg, J., Miller, S., Duckworth, D., Hu, H., Wan, A., Fu, X.-Y., Goldberg, K., and Abbeel, P. (2010). Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations, in *2010 IEEE International Conference on Robotics and Automation* (IEEE), pp. 2074–2081.

Wang, W. and Slotine, J.-J. E. (2005). On partial contraction analysis for coupled nonlinear oscillators, *Biological Cybernetics* **92**, 1, pp. 38–53.

Watanabe, T., Harada, K., Jiang, Z., and Yoshikawa, T. (2005). Object manipulation under hybrid active/passive closure, in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (IEEE), pp. 1013–1020.

Yang, J., Dani, A., Chung, S.-J., and Hutchinson, S. (2015). Vision-based localization and robot-centric mapping in riverine environments, *Journal of Field Robotics* .

Yun, X. and Kumar, V. R. (1991). An approach to simultaneous control of trajectory and interaction forces in dual-arm configurations, *IEEE Transactions on Robotics and Automation* **7**, 5, pp. 618–625.

Zhao, Y. and Cheah, C. C. (2009). Neural network control of multifingered robot hands using visual feedback, *IEEE Transactions on Neural Networks* **20**, 5, pp. 758–767.

Zhu, W.-H. (2005). On adaptive synchronization control of coordinated multirobots with flexible/rigid constraints, *IEEE Transactions on Robotics* **21**, 3, pp. 520–525.

Zöllner, R., Asfour, T., and Dillmann, R. (2004). Programming by demonstration: dual-arm manipulation tasks for humanoid robots. in *IROS*, pp. 479–484.