



Industrial Robot: An International Journal

Solving peg-in-hole tasks by human demonstration and exception strategies

Fares J. Abu-Dakka Bojan Nemeč Aljaž Kramberger Anders Glent Buch Norbert Krüger Aleš Ude

Article information:

To cite this document:

Fares J. Abu-Dakka Bojan Nemeč Aljaž Kramberger Anders Glent Buch Norbert Krüger Aleš Ude , (2014), "Solving peg-in-hole tasks by human demonstration and exception strategies", *Industrial Robot: An International Journal*, Vol. 41 Iss 6 pp. 575 - 584

Permanent link to this document:

<http://dx.doi.org/10.1108/IR-07-2014-0363>

Downloaded on: 04 November 2014, At: 02:55 (PT)

References: this document contains references to 26 other documents.

To copy this document: permissions@emeraldinsight.com

The fulltext of this document has been downloaded 16 times since 2014*

Users who downloaded this article also downloaded:

Honghai Liu, Jian S. Dai, Lakmal D. Seneviratne, (2008), "A model-based approach to cooperative operation of multirobot systems", *Industrial Robot: An International Journal*, Vol. 35 Iss 1 pp. 37-45

Access to this document was granted through an Emerald subscription provided by Token:JournalAuthor:D144736C-CC29-4680-A5FF-E4870E41FE76:

For Authors

If you would like to write for this, or any other Emerald publication, then please use our Emerald for Authors service information about how to choose which publication to write for and submission guidelines are available for all. Please visit www.emeraldinsight.com/authors for more information.

About Emerald www.emeraldinsight.com

Emerald is a global publisher linking research and practice to the benefit of society. The company manages a portfolio of more than 290 journals and over 2,350 books and book series volumes, as well as providing an extensive range of online products and additional customer resources and services.

Emerald is both COUNTER 4 and TRANSFER compliant. The organization is a partner of the Committee on Publication Ethics (COPE) and also works with Portico and the LOCKSS initiative for digital archive preservation.

*Related content and download information correct at time of download.

Solving peg-in-hole tasks by human demonstration and exception strategies

Fares J. Abu-Dakka

Humanoid and Cognitive Robotics Lab Department, Jožef Stefan Institute, Ljubljana, Slovenia

Bojan Nemeč and Aljaž Kramberger

Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia

Anders Glent Buch and Norbert Krüger

Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense M, Denmark, and

Aleš Ude

Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia

Abstract

Purpose – The purpose of this paper is to propose a new algorithm based on programming by demonstration and exception strategies to solve assembly tasks such as peg-in-hole.

Design/methodology/approach – Data describing the demonstrated tasks are obtained by kinesthetic guiding. The demonstrated trajectories are transferred to new robot workspaces using three-dimensional (3D) vision. Noise introduced by vision when transferring the task to a new configuration could cause the execution to fail, but such problems are resolved through exception strategies.

Findings – This paper demonstrated that the proposed approach combined with exception strategies outperforms traditional approaches for robot-based assembly. Experimental evaluation was carried out on Cranfield Benchmark, which constitutes a standardized assembly task in robotics. This paper also performed statistical evaluation based on experiments carried out on two different robotic platforms.

Practical implications – The developed framework can have an important impact for robot assembly processes, which are among the most important applications of industrial robots. Our future plans involve implementation of our framework in a commercially available robot controller.

Originality/value – This paper proposes a new approach to the robot assembly based on the Learning by Demonstration (LbD) paradigm. The proposed framework enables to quickly program new assembly tasks without the need for detailed analysis of the geometric and dynamic characteristics of workpieces involved in the assembly task. The algorithm provides an effective disturbance rejection, improved stability and increased overall performance. The proposed exception strategies increase the success rate of the algorithm when the task is transferred to new areas of the workspace, where it is necessary to deal with vision noise and altered dynamic characteristics of the task.

Keywords Learning from demonstration, Manipulation and compliant assembly

Paper type Research paper

1. Introduction

Automatic assembly has been a challenging area of research over the past decades. Generally, it requires an extremely tight positional accuracy and a good knowledge of parts' geometrical characteristics. The inaccuracy of the positioning devices used during part mating adds to the complexity of the process. One approach to help the assembly and avoid generation of large impact forces between the parts is to employ compliance and online adaptation of the desired trajectories (Nemeč *et al.*, 2013a). Human operators perform compliant movements easily by using force feedback and tactile information. Contacts and forces that occur during assembly are sensed and used to comply with the environment and adapt the assembly procedures. This

information may be also used in the subsequent operations to speed up the assembly process (Nemeč *et al.*, 2013b). There has been a significant amount of work to provide such ability for industrial robots using force feedback and modifying the position or velocity of the tool-tip. The process, however, often results in task-specific solutions that are application-oriented and, thus, unsatisfactory as general assembly solutions.

One of the standard operations needed for the automatic assembly is peg insertion, also referred to as peg-in-hole task (PiH). There are many approaches to solve the PiH problem, most of them are based on force feedback control (Yun, 2008, Hirana *et al.*, 2002, Giordano *et al.*, 2008, Newman *et al.*, 1999).

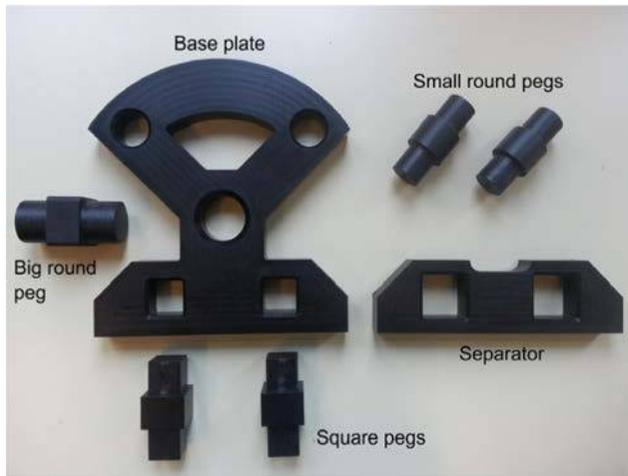
In this paper, the tasks for evaluation were taken from the Cranfield benchmark, (Figure 1) (Collins *et al.*, 1985), which constitutes a standardized assembly task in robotics. It requires the insertion of different pegs (round and square) into

The current issue and full text archive of this journal is available at www.emeraldinsight.com/0143-991X.htm



Industrial Robot: An International Journal
41/6 (2014) 575–584
© Emerald Group Publishing Limited [ISSN 0143-991X]
[DOI 10.1108/IR-07-2014-0363]

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 269959, IntellAct, and no. 600578, ACAT.

Figure 1 Cranfield benchmark

the corresponding holes of the base plate and the placement of a separator onto the two previously inserted pegs.

Visual serving and reactive task controllers can also play an important role (Hamner *et al.*, 2010). Because the appropriate strategy depends on the geometry of the manipulated workpieces, in most traditional approaches an engineer has to hardcode a different strategy for every new workpiece geometry (Bruyninckx *et al.*, 1995; Xiao, 1997).

In this paper, an approach has been developed for learning of force-based skills. This approach consists of four stages:

- 1 Kinesthetic guiding is first used to perform the desired task in a specific part of the robot workspace. Besides positions, velocities and accelerations, we also acquire the associated force and torque profiles that led to the successful task execution. Note that unlike standard programming by demonstration (PbD) (ARGALL *et al.*, 2009), we also make use of force/torque profiles that occur during the task execution to learn the appropriate behavior. There are only few other works that exploit forces and torques acquired during human demonstration. For example, Kormushev *et al.* (2011), Rozo *et al.* (2013) designed a system where a haptic interface is utilized to demonstrate the desired movements and the associated forces and torques.
- 2 If necessary, the acquired force/torque profiles are refined via reinforcement learning (Kalakrishnan *et al.*, 2011), and the optimal stiffness parameters can be learned.
- 3 Once the new workpiece pose is acquired by vision, the demonstrated skill is transferred to new configurations, and the transferred movement is adapted to the trained forces and torques.
- 4 Exception strategies are applied when the peg fails to enter the hole due to errors in the estimation of the new workpiece pose by the three-dimensional (3D) vision system.

The main focus of this paper is on Item 4, i. e. the application of exception strategies to resolve situations where the robot fails to accomplish the task outright. We also explain how Item 3 is solved. We focus on situations where the PiH task needs to be executed in a different part of the workspace than during training. In our system, a new workpiece pose is estimated by vision, which introduces noise, thus simply transforming the

previously trained Cartesian space trajectory and replaying it in a new configuration does not lead to a successful execution of the task. Such an approach is guaranteed to fail due to various reasons including vision noise and uncertainties in the grasping posture of the peg. Moreover, Cartesian space tracking errors can occur due to a different joint space configuration of the robot arm.

In our approach, we, therefore, first try to improve the transformed control policy by matching it with the previously trained performance in the force/torque space using force feedback control. The main idea is to match the recorded forces and torques with the currently measured ones by iteratively modifying the transformed Cartesian space trajectory. Through subsequent repetitions of the task execution, our learning procedure moves the force feedback errors to the encoded trajectory, thus improving the execution of the task and making it closer to the originally trained performance. Our experimental results show that with the proposed approach the robot can effectively learn and transfer assembly operations to new situations (Nemec *et al.*, 2013a). To further improve the effectiveness of the algorithm, we introduce exception strategies, which can be utilized to solve the problem of missing the hole due to vision errors, thereby increasing the success rate of the algorithm. When the vision system acquires a new pose of the baseplate of the Cranfield benchmark (Figure 1), the algorithm adapts the Cartesian space trajectory to this new pose. However, due to the improper calibration and vision error, the initial pose is often shifted and rotated with respect to the true initial (hole) pose. We solved the abovementioned problem by integration of exception strategies into the learning framework.

The paper consists of six sections. In Section 2, we briefly present the data acquisition procedure used for acquiring trajectories and forces for an assembly operation. In Section 3, we formulate the problem and present the dynamic movement primitives (DMP) framework for encoding of discrete trajectories. Moreover, an approach to iteratively adapt the learned trajectory to improve the task performance is reviewed. Section 4 presents the main result of this paper, which is the exception strategies developed to improve the success ratio of the PiH operation. Results of the experimental evaluation are given in Section 5, followed by final remarks, summary of main contributions and conclusions in Section 6.

2. Programming by demonstration for force-based tasks

In this section, we briefly describe the basic procedure of learning PiH operations by a human demonstration. The human demonstrations are performed by kinesthetic guiding using the Kuka LWR arm in gravity compensation mode (Figure 2). The human guides the robot's tool center point along the desired trajectory, for example, so that the peg insertion task is successfully executed. We measure the Cartesian space trajectory by proprioception during the execution. In the Kuka LWR arm, the force sensors are located in the robot joints and consequently the forces exerted from the human operator during the demonstration affect the measured forces and torques. Therefore, to obtain the net forces and torques at the robot tool center point, we repeat the measured trajectory with the robot, record the resulting joint

Figure 2 Kinesthetic guiding using Kuka LWR arm in gravity compensation mode



torques and transform them into the corresponding tool center point forces and torques. We take care that the workspace configuration does not change during this process and since our robot can accurately track the joint space trajectories, the measured forces and torques provide a good reference for later adaptation.

With this system, each successful demonstration of PiH task, the algorithm acquires the Cartesian space tool center point positions and orientations (represented as quaternions), velocities and accelerations:

$$\mathcal{G}_d = \{\mathbf{p}_i, \mathbf{q}_i, \dot{\mathbf{p}}_i, \omega, \dot{\mathbf{q}}_i, \dot{\omega}\}_{i=0}^T, \quad (1)$$

and the associated forces and torques:

$$\mathcal{F}_d = \{\mathbf{F}_i, \mathbf{M}_i\}_{i=0}^T, \quad (2)$$

all acquired at times t_i , $i = 0, \dots, T$. Positions and orientations are represented as a frame $\mathcal{Q} = \{\mathbf{p}, \mathbf{q}\}$, where $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{q} = v + \mathbf{u}$ is a unit quaternion where $v \in \mathbb{R}$ and $\mathbf{u} \in \mathbb{R}^3$.

3. Problem formulation

In this section, technologies used for the trajectory adaptation of an assembly process will be introduced. The robot trajectories are represented with DMPs framework (Ijspeert et al., 2013). The main feature of this framework for our purposes is that it enables to slow down or speed up the trajectories. In Section 3.1, this framework is presented. In Section 3.2, we explain how the demonstrated trajectories are modified to successfully execute a task in different parts of the robot workspace. In Section 3.3, we used the Iterative Learning Control to learn and adapt the trajectories. This adaptation is performed in several iterations, where the robot repeats the same task. Finally, a scheme for the overall procedure is introduced (Section 3.4).

3.1 DMPs for Cartesian space

The demonstrated Cartesian space trajectories that result in a successful execution of the PiH task are encoded by DMPs. A DMP for a single df trajectory y is defined by the following set of nonlinear differential equations (Ijspeert et al., 2013):

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x) \quad (3)$$

$$\tau \dot{y} = z, \quad (4)$$

$$\tau \dot{x} = -\alpha_x x, \quad (5)$$

where x is the phase variable, z is an auxiliary variable and τ is the time variable. Parameters α_z and β_z define the behavior of the second-order system described by equations (3) and (4). The evolution of the phase variable x is defined by equation (5). With the choice of the time constant $\tau > 0$, $\alpha_z = 4\beta_z$ and $\alpha_x > 0$, the convergence of the underlying dynamic system to a unique attractor point at $y = g$; $z = 0$ is ensured (Ijspeert et al., 2013). $f(x)$ is defined as a linear combination of N nonlinear radial basis functions, which enables the robot to follow any smooth trajectory from the initial position y_0 to the final configuration g (goal):

$$f(x) = \frac{\sum_{i=1}^N w_i \Psi_i(x)}{\sum_{i=1}^N \Psi_i(x)} x(g - y_0), \quad (6)$$

$$\Psi_i(x) = \exp(-h_i(x - c_i)^2), \quad (7)$$

where Ψ_i are fixed basis functions and w_i are adjustable weights. c_i are the centers of Gaussians distributed along the phase of the trajectory with h_i are their widths. For a given N and setting the time constant $\tau = t_T$, we can define $c_i = \exp(-\alpha_x i - 1/N - 1)$, $h_i = 1/(c_{i+1} - c_i)^2$, $h_N = h_{N-1}$, $i = 0, \dots, N$. For each Cartesian df, the weights w_i are estimated from the measured data equation (1) using regression (Ude et al., 2010), while g is the last-recorded configuration on the trajectory. Thus, the resulting DMPs encode the desired peg insertion trajectory.

In the basic DMP equations (3) – (4), each df of the position/orientation dimension is encoded as a single DMP with a common phase variable x . Note that we encode eh quaternion component as a single DMP; therefore, the resulting quaternion has to be normalized after each integration step. The exact solution is proposed in Pastor et al.'s study (2011), but because we confirmed experimentally that the differences between both approaches are negligible, we use equation (3) – (4) also for quaternion integration. Such an approach simplifies the implementation.

3.2 Error feedback and DMP phase stopping

When the robot executes the demonstrated trajectory, the resulting forces and torques differ from the ones that were measured during human demonstration. This happens due to small displacements arising from inaccurate pose estimation, reduced accuracy of the robot tracking control and uncertainties in the placement of the peg in the gripper. This could worsen or even prevent the successful execution of the PiH task. To adapt to a new situation, we propose to modify the demonstrated trajectory according to the stiffness control law (Villani and Schutter, 2008):

$$\mathbf{p}_r(x) = \mathbf{K}_{s1} \mathbf{e}_p(x) + \mathbf{p}'_{DMP}(x), \quad (8)$$

$$\mathbf{q}_r(x) = \mathcal{R}(\mathbf{K}_{s2} \mathbf{e}_q(x)) * \mathbf{q}'_{DMP}(x), \quad (9)$$

where $\mathbf{p}_r(x)$ is the position vector fed to the robot controller, $(\mathbf{p}'_{DMP}, \mathbf{q}'_{DMP})$ is the displaced demonstrated trajectory

computed by integrating the learned DMPs (\mathbf{p}_{DMP} , \mathbf{q}_{DMP}) and applying t workpiece displacement ($\Delta\mathbf{t}_w$, $\Delta\mathbf{q}_w$),

$$\mathbf{p}'_{DMP}(x) = \Delta\mathbf{q}_w * \mathbf{p}_{DMP}(x) * \overline{\Delta\mathbf{q}_w} + \Delta\mathbf{t}_w \quad (10)$$

$$\mathbf{q}'_{DMP}(x) = \Delta\mathbf{q}_w * \mathbf{q}_{DMP}(x) \quad (11)$$

\mathbf{K}_{s1} and \mathbf{K}_{s2} are 3×3 diagonal matrices, $\mathbf{q}_i(x)$ is the orientation quaternion fed to the robot controller. $\mathbf{q}_{DMP}(x)$ is the reference quaternion obtained integrating the learned DMP, and $\overline{\Delta\mathbf{q}_w}$ denotes conjugated quaternion $\Delta\mathbf{q}_w$. The quaternion product “*”, defined as $\mathbf{q}_1 * \mathbf{q}_2 = (v_1, \mathbf{u}_1) * (v_2, \mathbf{u}_2) = (v_1v_2, v_2\mathbf{u}_1 + v_1\mathbf{u}_2 + \mathbf{u}_1 \times \mathbf{u}_2)$, has been used in equation (9). The feedback error terms, i.e. $\mathbf{K}_{s1}\mathbf{e}_p(x)$ and $\mathcal{R}(\mathbf{K}_{s2}\mathbf{e}_q(x))$, respectively, provide force/torque feedback control. The errors $\mathbf{e}_p(x)$ and $\mathbf{e}_q(x)$ are defined as:

$$\mathbf{e}_p(x) = \mathbf{q}(x) * (\mathbf{F}_d(x) - \mathbf{F}) * \overline{\mathbf{q}(x)}, \quad (12)$$

$$\mathbf{e}_q(x) = \mathbf{q}(x) * (\mathbf{M}_d(x) - \mathbf{M}) * \overline{\mathbf{q}(x)}, \quad (13)$$

where $\mathbf{F}_d(x)$, $\mathbf{M}_d(x)$ are the desired reference forces and torques at phase x as acquired from human demonstration, $\mathbf{F}_d = [F_{d,x}, F_{d,y}, F_{d,z}]^T$ and $\mathbf{M}_d = [M_{d,x}, M_{d,y}, M_{d,z}]^T$. Linear equations systems similar to equation (17) are needed to be solved to estimate \mathbf{F}_d and \mathbf{M}_d from the measured force/torque data equation (2). Subscript s (\cdot)_{X,Y,Z} denote the components of \mathbf{F}_d vectors in directions of Cartesian axes or rotation components of \mathbf{M}_d around Cartesian axes, respectively. Please note that the uppercase letters X, Y, Z denote Cartesian axis, while the lower case letters x, y, z denote DMP variables.

\mathbf{F} and \mathbf{M} are the current measured forces and torques, $\mathbf{q}(x)$ is the quaternion specifying the current orientation of the robot’s tool, and $\overline{\mathbf{q}(x)}$ denotes conjugated quaternion $\mathbf{q}(x)$. \mathcal{R} denotes the transformation which maps an angular velocity to a unit quaternion describing the resulting rotation within the sampling time Δt :

$$\mathcal{R}(\omega) = \left(\cos \frac{\omega \Delta t}{2}, \frac{\omega}{\omega} \sin \frac{\omega \Delta t}{2} \right), \quad (14)$$

The proposed controller tracks simultaneously the desired position and orientations and forces and torques. Force/torque adaptation requires low gains (matrices \mathbf{K}_s) for stable and robust operation. Thus, force adaptation is usually slow. To effectively minimize the force/torque error, we propose slowing down the trajectory execution using DMP slowdown feedback. For DMP phase stopping (Schaal et al., 2007), the original equation for phase equation (5) is replaced with:

$$\tau \dot{x} = - \frac{\alpha_x x}{1 + \alpha_{px} \mathbf{e}}, \quad (15)$$

where \mathbf{e} is the combined force/torque error,

$$\mathbf{e} = [\mathbf{e}_p^T, \mathbf{e}_q^T]. \quad (16)$$

Note that in the case of large forces or torques, the error \mathbf{e} becomes large which, in turn, makes the phase change \dot{x} small.

Thus, the phase evolution is stopped until the robot reduces the force/torque error (Nemec et al., 2013a).

During the execution of the demonstrated trajectory, the resulting positions and orientations offsets are captured depending on the phase variable x , which ensures that the sampling is independent of the trajectory duration (Nemec et al., 2013a). Thus, the trajectory is sampled exactly the same number of times as during the human demonstration, even when the phase is slowed down during the execution according to equation (15).

3.3 Trajectory adaptation

The goal of learning is to iteratively modify in sequential steps the positional and orientation part of the demonstrated trajectory so that the robot-generated trajectory results in the same forces and torques as during the human demonstration. For this purpose, we apply the Iterative Control Learning (ILC) framework, where the control signal from the previous iteration cycle is reused in the next iteration (Bristow et al., 2006). For that, we simply calculate new updated trajectory using control law equations (10) and (11), save entire trajectory and calculate new DMP weights $w_{k,p}$ for positional part $\mathbf{p}'_{DMP}(x)$ and $w_{k,q}$ for rotational part $\mathbf{q}'_{DMP}(x)$, respectively. Subscript $(\cdot)_k$ refers to the k^{th} component of positional vector and quaternion, respectively.

The optimal weights $w_{k,p}$ are then computed by solving the following linear system of equations:

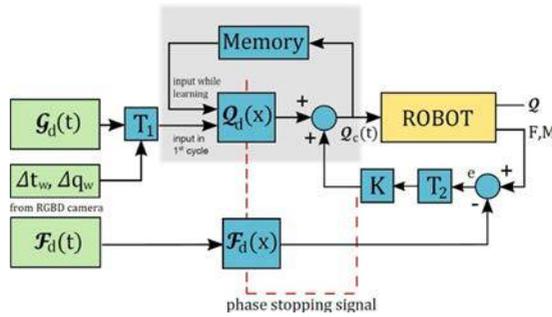
$$\begin{aligned} \mathbf{A} \mathbf{w}_{k,p} &= \mathbf{p}_{k,p} \\ \mathbf{w}_{k,p} &= \begin{bmatrix} w_{1,k,p} \\ \vdots \\ w_{M,k,p} \end{bmatrix}, \mathbf{p}_k = \begin{bmatrix} \mathbf{p}_k(x_0) \\ \vdots \\ \mathbf{p}_k(x_T) \end{bmatrix}, \\ \mathbf{A} &= \begin{bmatrix} \frac{\Psi_1(x_0)x_0}{\sum_{i=1}^M \Psi_i(x_0)} & \cdots & \frac{\Psi_M(x_0)x_0}{\sum_{i=1}^M \Psi_i(x_0)} \\ \vdots & \ddots & \vdots \\ \frac{\Psi_1(x_T)x_T}{\sum_{i=1}^M \Psi_i(x_T)} & \cdots & \frac{\Psi_M(x_T)x_T}{\sum_{i=1}^M \Psi_i(x_T)} \end{bmatrix} \quad (17) \end{aligned}$$

where Ψ_i are the Gaussian kernel functions from equation (7) and \mathbf{p}_k are signals sampled from equation (8). Similarly we compute also the weights $w_{k,q}$. Reference position vectors $\mathbf{p}'_{DMP}(x)$ and quaternions $\mathbf{q}'_{DMP}(x)$ are then computed by integrating equations (3)-(5). Note that the quaternion part of the trajectory \mathbf{q}'_{DMP} has to be normalized before being used in equation (9). An attention should be paid also when sampling signals from equations (8) and (9). They should not be sampled at each control cycle, as the phase variable is modified according to the equation (15). Instead, they have to be sampled exactly at $x_i = \exp(-\alpha_x t_i)$, t_i , $i = 0, \dots, T$.

3.4 Control scheme

Figure 3 shows the basic implementation scheme for the proposed learning control. In this scheme, $\mathcal{G}_d(t)$ [equation (1)] denotes the original demonstrated trajectory, which is encoded by DMPs. \mathcal{Q}_d refers to the output signal obtained by integrating. Equations (4) and (5) and applying the workpiece displacement ($\Delta\mathbf{t}_w$, $\Delta\mathbf{q}_w$) as estimated by vision which occurs in block \mathbf{T}_1 in equations (10) and (11). Note that this input is calculated and used only in the first learning cycle. \mathcal{F}_d

Figure 3 Basic learning scheme



[equation (2)] denotes the forces and torques captured during the execution of the demonstrated trajectory. Due to the noise induced by the errors in the vision system, robot tracking errors and imperfect grasping, the measured forces/torques \mathbf{F} , \mathbf{M} differ from the desired demonstrated forces \mathcal{F}_d [equation (2)]. Our goal is to minimize that error. In block \mathbf{T}_2 , the error in equations (12) and (13) is transformed into robot base coordinates. Using admittance control in equations (8) and (9), the new reference trajectory is computed. Motor commands \mathcal{Q}_c are given as the aggregation of the DMP generated trajectory and force feedback equations (8) and (9). In following learning cycles, the reference trajectory from the previous learning cycle is used to calculate $\mathcal{Q}_d(x)$ using equations (19), (3)-(7) instead of the reference trajectory as calculated in the first cycle. The whole procedure is repeated until the measured forces match the desired forces.

4. Exception strategies

The proposed approach is robust to the small to moderate deviations of the workpieces positions and dimensions. But can fail at large deviations. We investigated the human behavior in such situations and found out, that humans in such cases mostly interrupt the pre-learned actions and perform additional actions, called exception strategies. Exception strategies are generally triggered together with the phase stopping at high force/torque errors. Offset learned by the exception strategies is added to the offset learned with the iterative learning framework. Eventual jumps in trajectory are automatically filtered by applying DMP framework for offset encoding. Exception strategies can be either deterministic or stochastic. When sufficient *a priori* knowledge of the process/environment is available, deterministic strategies are generally faster than stochastic. Stochastic exception strategy, which basically performs random motions in small increments, does not require any previous knowledge and is more robust to the sensor noise. As an example, we will describe the stochastic exception strategy, which is used to compensate the error of the initial pose of the work-piece. These errors are mostly due the vision subsystem and are normally more evident in the extremes of the working space. In other words, due to calibration errors and distortion the pose estimation is less accurate at the boundaries.

The main uncertainties in our system are related to the vision aspect. We use a Kinect sensor for acquiring 3D scene data, and we apply an existing system (Buch *et al.*, 2013) for

estimating the pose of the known objects based on point cloud data. For the objects shown in Figure 1, we have the corresponding CAD models, and these are resampled to point clouds for use by the pose estimation algorithm.

The primary source of uncertainties is the internal calibration of the Kinect sensor. It has been shown recently (Herrera *et al.*, 2012) that the depth error for this sensor is up to 5 mm at a distance of 1 m.

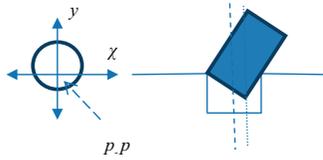
The second source of errors lies in the external calibration between the Kinect and the robot. We performed the external hand-eye calibration by a manual process where the robot is placed in several random configurations, followed by a measurement of the position of a known fixation point at the tool center point in both the robot base frame (using forward kinematics) and in the camera sensor frame (using a graphical interface which allows the user to mark the fixation point in the point cloud). From the corresponding position pairs, the relative transformation between the frames can be estimated by the singular value decomposition of the point covariance matrix (Kabsch, 1976). Although this method provides an optimal estimate, the result is still greatly affected by biases and noise in the measurements.

Finally, the pose estimation system introduces errors. Although pose refinement (Besl and McKay, 1992) is applied to each pose estimate, the accuracy of the end result is still affected by the sensor noise. Even for an optimal alignment, the pose estimation can thus still produce a translation error in the range of the sensor noise, i.e. up to 5 mm at a distance of 1 m from the sensor.

To handle all these uncertainties in a principled manner, and to increase the success rate of the algorithm, we introduced an exception strategy to the algorithm. This exception strategy is combined by two phases:

- 1 *Verification phase*: This phase is divided into following steps:
 - *Approaching*: Move the robot tool down in global z -coordinate using implicit force control to detect the surface of the baseplate and overcome any pose estimation errors in z -coordinates.
 - *Verification*: Due to the limited precision of the vision system, we never know whether the robot is at the required starting position above the hole which required for a successful accomplishment of the PiH operation. Therefore, first step should be verification of the required starting pose. Because we cannot rely on vision, we use force-sensing instead. Verification is executed in following steps, where we assume that the workpiece lies in the X - Y plane (Figure 4):
 - establish the contact with the environment using hybrid force-torque control law [Equations (19) and (20)] by prescribing the desired force in the Z direction;
 - perform linear motion in the global X coordinate starting from the current point with the displacement of dx and $-dx$;
 - determine minimum of Z coordinate and move to that point;
 - perform linear motion in the global Y coordinate with the displacement dy and $-dy$; and

Figure 4 Verification phase: robot searches for the maximal penetration point in global X and Y coordinates



- determine the minimum of Z coordinate. If minimum exists, we have hit the hole and the resulting pose p_p corresponds to the pose of the maximal penetration of the peg into a hole.

We might not found a minimum in Z coordinate during this verification procedure. In this case, we have to proceed with the next step, which is hole-search exception strategy. Note that maximal penetration point p_p does not correspond to the center of the hole.

If the previously described procedure has not succeeded to detect the maximal penetration point p_p , we proceed with the following exception strategy.

- 2 *Searching phase (Figure 5)*: Using stochastic search algorithm to detect the hole. This algorithm does small random increments ε in X and Y coordinates. The values of ε are taken randomly in a circle with a specific diameter. In the experiment that diameter was 1.5 cm:

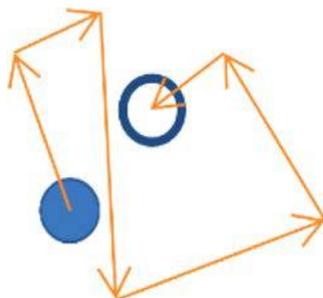
$$\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix} \quad (18)$$

In the searching phase, linear DMPs are used to encode the trajectories between those random positions. Linear DMPs imply that $f(x) = 0$ in equation (3). During the execution of these trajectories, a force feedback is used to update the learned trajectory to maintain constant contact force with the surface of the baseplate. The end-effector velocity is calculated by:

$$\mathbf{v}_r = \mathbf{S}_v \mathbf{v}_v + \mathbf{K}_F \mathbf{S}_F (\mathbf{F}_m - \mathbf{F}_d) \quad (19)$$

where \mathbf{v}_r stands for the resolved velocities vector, \mathbf{S}_v for the velocity selection matrix, \mathbf{v}_v for the desired velocities vector, \mathbf{K}_F for the force gain matrix, \mathbf{S}_F for the force selection matrix,

Figure 5 Stochastic search



\mathbf{F}_m for the measured force and \mathbf{F}_d denotes the force offset which determines the behavior of the robot when it is not in contact with the environment. To get the desired positions, we use:

$$\mathbf{y} = \mathbf{y}_r + \int \mathbf{v}_r dt \quad (20)$$

where \mathbf{y}_r is the desired initial position/orientation and \mathbf{y} is the actual position/orientation. Using this approach, we can online modify the trajectory of the learned movement by updating the selected df. Matrices \mathbf{S}_v and \mathbf{S}_F are often diagonal and are used to select the df that are updated using force feedback. This leads to what is known as hybrid control. In a hybrid control scheme, the manipulator is required to follow the environment surface at a given speed while maintaining a desired contact force \mathbf{F}_d . Such control schemes can be written with a proper selection of vectors σ , for example, $\sigma = [1; 1; 1; 0; 0; 0]^T$, which denotes that the translational dfs of a 6-df manipulator are trajectory-controlled, while the rotations are force-controlled. Therefore, equation (19) becomes:

$$\mathbf{v}_r = \text{diag}(\sigma) \mathbf{v}_v + \mathbf{K}_F (\mathbf{I} - \text{diag}(\sigma)) (\mathbf{F}_m - \mathbf{F}_d),$$

$$\text{diag}(\sigma) = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_6 \end{bmatrix} \quad (21)$$

Note that the velocity selection matrix \mathbf{S}_v and the force selection matrix \mathbf{S}_F are complementary; $\mathbf{S}_F = \mathbf{I} - \mathbf{S}_v$. Technically this means that a single df cannot be at the same time force- and trajectory-controlled.

The force gain matrix \mathbf{K}_F , from equation (19), multiplied by the measured force (or measured force minus the desired force), gives the end-effector velocity. Physically, the force gain matrix can be compared to the stiffness matrix (usually denoted by K), which means how much the environment or the robot deforms/moves when a force is applied to it. The stiffness and the compliance matrix (usually denoted by C) are related so that $C = K^{-1}$.

For the case of wiping a flat surface, where only the height (the Z direction) is needed to be updated, the equations are considerably simplified. By using $\mathbf{S}_v = \text{diag}(1; 1; 0; 1; 1; 1)$, $\mathbf{K}_F = \text{diag}(0; 0; k_F; 0; 0; 0)$ and $\mathbf{S}_F = \text{diag}(0; 0; 1; 0; 0; 0)$. The desired end-effector height z in each discrete time step Δt becomes:

$$\dot{z}(t) = k_F (F_Z(t) - F_d) \quad (22)$$

$$z(t) = z_0 + \dot{z}(t) \Delta t \quad (23)$$

Here, z_0 is the starting height, k_F is the force gain [kg/s], F_Z is the measured force in the Z direction and F_d is the desired force that the robot needs to press on the object. Such formulation of the movement ensures constant movement in the $-Z$ direction or constant contact when an object is encountered.

5. Experimental evaluation

The task was to evaluate the success rate of the insertion of the square and round peg into a hole of the Cranfield benchmark with and without using exception strategies. Trajectories were obtained as described in Section 2. We executed the trajectory for different randomly selected positions of the base plate. The measured trajectory data was translated and rotated into the new configurations using the pose of the base plate as estimated by vision. In this scenario, the uncertainties come from vision. Therefore, the resulting forces ring the execution of the demonstrated trajectory may exceed the expected ones.

The learning procedure was implemented in Matlab, which communicated with the Kuka LWR controller using Fast Research Interface (Schreiber *et al.*, 2010). Kuka LWR is capable of measuring joint torques and can switch to Cartesian compliance control. This experiment is comprised of eight groups. Each group contains 50 trial execution of the PiH algorithm (Tables I and II). In each trial, the base plate was randomly put at different locations in the area of $0.5 \times 0.2 \text{ m}^2$ on the table within the robot's workspace. The poses of the base plate were estimated by vision. From the tables, it can be clearly noticed that the exception strategy significantly improved the success ratio especially in case of square and tight round pegs. The average time needed for the execution of the exception strategy was 25 seconds over the time for the standard ones. The insertion time is always greater than or equal to the demonstrated trajectories time. In case of the shaft, the demonstrated trajectory time is 5 seconds, while the average insertion time of the experiments was 6.7 seconds. This time after the adaptation will be decreased to be more close to the demonstrated one.

Figure 6 shows graphically the improvement of the success ratio of the PiH insertion with Cranfield benchmark's different objects.

In the next Figure 7, it can be observed that the offset of the error between the actual pose of the hole (indicated by red

Table I Success ration of PiH insertion without using exception strategy

Without exception strategy	No. of trials	No. of success trials	Success ratio (%)
Shaft peg	50	48	96.00
Small round peg	50	48	96.00
Small tight round peg	50	38	76.00
Square peg	50	33	66.00

Table II Success ration of PiH insertion using exception strategy

With exception strategy	No. of trials	No. of success trials	Success ratio (%)
Shaft peg	50	50	100.00
Small round peg	50	50	100.00
Small tight round peg	50	50	100.00
Square peg	50	48	96.00

circle) in the baseplate and the pose estimated by vision. Those segments represent the Euclidean distance between the pose of the robot tip and the pose of the hole. As it can be shown from the figure, the offsets are more evident at the extremes of the workspace.

Further experiments were executed on the Kuka LWR robot arm to test the impact of the algorithm when phase stopping is disabled. In these experiments, the shaft and round peg were used with the same setup in the first and second groups in Table I. Running the experiments, it can be clearly observed that the application of phase stopping improves the reliability of our approach. The success rate achieved without phase stopping was 66 per cent. In these experiments, 100 per cent success rate was achieved by enabling phase stopping and integrating exception strategy for finding the hole into the system.

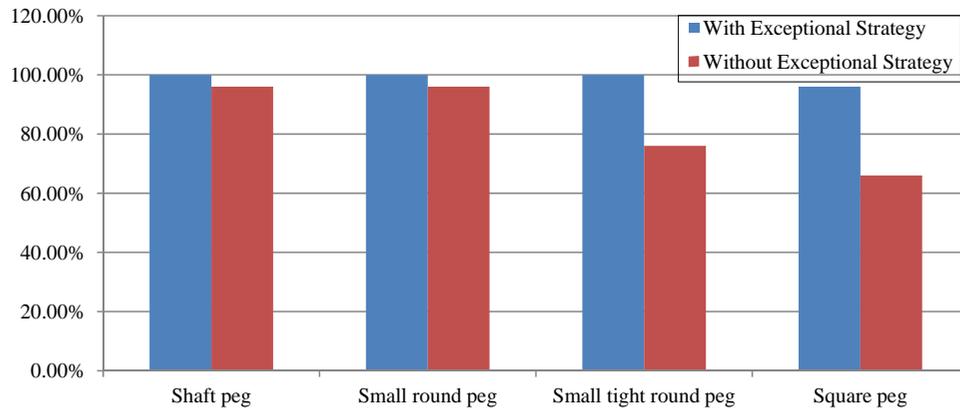
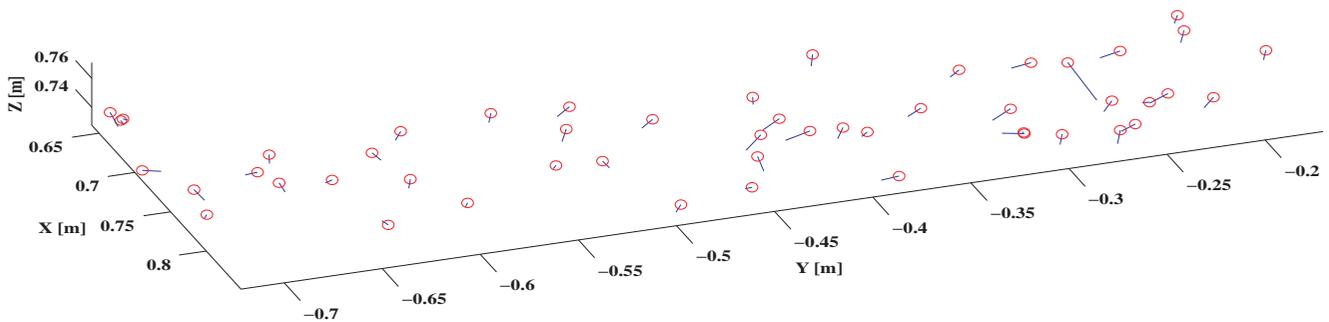
Additional experiments were conducted on the Universal UR5 robot that uses high-gain non-compliant controller. The learning procedure was implemented in C++ and linked to the robot using ROS (Quigley *et al.*, 2009). From the Tables III and IV, it can be clearly noticed that the exception strategy significantly improved the success ratio especially in the case of square and tight round pegs.

Figure 8 shows graphically the improvement of the success ratio of the PiH insertion with Cranfield benchmark's different objects. From figure and tables, we can notice that the success ratio obtained by Kuka robot is slightly higher than the one obtained by UR5. That was due to some factors, such as the grasping way in Kuka gripper was considered as fixed grasping which gives more stable grasping rather than the SCHUNK SDH. Moreover, the UR5 is stiff robot, which implies that in some cases the peg get stuck, and because of the forces, it would move inside the hand before it gives enough time for the controller to do corrections.

6. Conclusion

In this paper, we show how to integrate force-based programming by demonstration and exception strategies to adapt tasks in contact with the environment. We start with a human demonstration of the task because humans can effectively perform tasks that involve force feedback. Thus, their performance can be taken as baseline for robot performance. The effectiveness of the approach was demonstrated on tasks that cannot be performed without force feedback, such as peg insertion. Robust adaptation and exception strategies are essential for such tasks because even small errors can cause the robot execution to fail. The robot first adapts its movement in such a way that the adapted movement generates the same forces and torques as during demonstration. The core of our proposed strategy is not to develop different strategies for every shape of the object that needs to be inserted. Instead, we developed a general strategy that relies on the demonstrated policy, which includes positions/orientation and forces/torques. To overcome the abovementioned uncertainties, exception strategies were developed and integrated into the algorithm. The developed exception strategy is stochastic and performs random motions in small increments. This strategy does not require any previous knowledge and is robust to the force sensor noise.

Experimental evaluation of the proposed approach includes the use of 3D vision system and workpieces of

Figure 6 Success ratio comparison in case of using/without using exception strategy**Figure 7** Offset between hole pose estimated from vision and the actual pose of it**Table III** Success ration of PiH insertion without using exception strategy

Without exception strategy	No. of trials	No. of success trials	Success ratio (%)
Shaft peg	50	46	92.00
Small round peg	50	46	92.00
Small tight round peg	50	36	72.00
Square peg	50	31	62.00

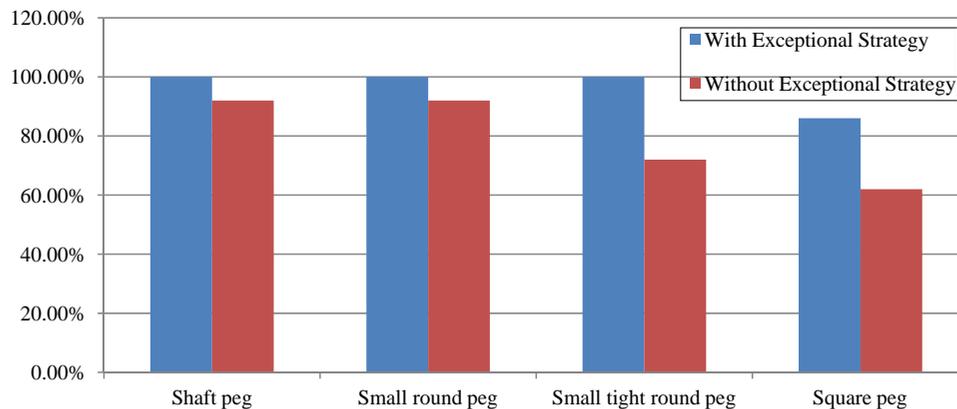
Table IV Success ration of PiH insertion using exception strategy

With exception strategy	No. of trials	No. of success trials	Success ratio (%)
Shaft peg	50	50	100.00
Small round peg	50	50	100.00
Small tight round peg	50	50	100.00
Square peg	50	43	86.00

different geometry and different number of holes. Our experimental results show that the developed system can effectively learn complex manipulation tasks that involve contact with the environment. New manipulation behaviors can quickly be learned by human demonstration and later be adapted to different configurations of the workspace. The flexibility of the approach is demonstrated by training a variety of assembly tasks involving different workpiece geometries and different number of holes. Figures 6 and 8 show that the success ratio of the PiH operation has been improved significantly by introducing the exception strategies. The proposed strategy is robust and can be used in industrial assembly applications. The tests that we conducted show the hardware independence of the proposed strategy. In this way, we proved the robustness and the reliability of the proposed method.

Majority of the existing algorithms try to solve the PiH problem without learning. Note that in the first learning cycle our algorithm is similar to most classical strategies with the exception that we applied phase stopping strategy, which actually raised the success rate substantially. After that, our algorithm improves through cycles, which is evident from the results. Therefore, it can be concluded that the existing algorithms are at most as good as our algorithm in the first cycle.

In our future work, we will investigate the possibility to accelerate the demonstrated assembly policy, which is a challenging task. Namely, the accelerated movements change also the interaction forces/torques, which have to be adjusted accordingly. Moreover, a thorough investigation can be done between the influences of the various vision-related uncertainties on the resulting PiH success rate.

Figure 8 Success ratio comparison in case of using/without using exception strategy

References

- Argall, B.D., Chernova, S., Veloso, M. and Browning, B. (2009), "A survey of robot learning from demonstration", *Robotics and Autonomous Systems*, Vol. 57 No. 5, pp. 469-483.
- Besl, P.J. and McKay, N.D. (1992), "A method for registration of 3-D shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14 No. 2, pp. 239-256.
- Bristow, D.A., Tharayil, M. and Alleyne, A.G. (2006), "A survey of iterative learning control", *IEEE Control Systems Magazine*, Vol. 26 No. 3, pp. 96-114.
- Bruyninckx, H., Dutre, S. and De Schutter, J. (1995), "Peg-on-hole: a model based solution to peg and hole alignment", *IEEE International Conference on Robotics and Automation*, Nagoya, pp. 1919-1924.
- Buch, A.G., Kraft, D., Kamarainen, J.-K., Petersen, H.G. and Kruger, N. (2013), "Pose estimation using local structure-specific shape and appearance context", *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, pp. 1050-1079.
- Collins, K., Palmer, A.J. and Rathmill, K. (1985), "The development of a European benchmark for the comparison of assembly robot programming systems", in Rathmill, K., Macconail, P., O'Leary, S. and Browne, J. (Eds), *Robot Technology and Applications*, Springer-Verlag, New York, NY.
- Giordano, P.R., Stemmer, A., Arbter, K. and Albu-Schaffer, A. (2008), "Robotic assembly of complex planar parts: an experimental evaluation", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, pp. 3775-3782.
- Hamner, B., Koterba, S.K., Shi, J., Simmons, R. and Singh, S. (2010), "An autonomous mobile manipulator for assembly tasks", *Autonomous Robots*, Vol. 28 No. 1, pp. 131-149.
- Herrera, C.D., Kannala, J. and Heikkila, J. (2012), "Joint depth and color camera calibration with distortion correction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34 No. 10, pp. 2058-2064.
- Hirana, K., Suzuki, T. and Okuma, S. (2002), "Optimal motion planning for assembly skill based on mixed logical dynamical system", *7th International Workshop on Advanced Motion Control*, Maribor, Slovenia, pp. 359-364.
- Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P. and Schaal, S. (2013), "Dynamical movement primitives: learning attractor models for motor behaviors", *Neural Computation*, Vol. 25 No. 2, pp. 328-373.
- Kabsch, W. (1976), "A solution for the best rotation to relate two sets of vectors", *Acta Crystallographica Section A*, Vol. 32 No. 5, pp. 922-923.
- Kalakrishnan, M., Righetti, L., Pastor, P. and Schaal, S. (2011), "Learning force control policies for compliant manipulation", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, pp. 4639-4644.
- Kormushev, P., Calinon, S. and Caldwell, D.G. (2011), "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input", *Advanced Robotics*, Vol. 25 No. 5, pp. 581-603.
- Nemec, B., Abu-Dakka, F., Rytz, J.A., Savarimuthu, T.R., Ridge, B., Krüger, N., Petersen, H.G., Jouffroy, J. and Ude, A. (2013a), "Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile", *16th International Conference on Advanced Robotics (ICAR)*, Montevideo.
- Nemec, B., Gams, A. and Ude, A. (2013b), "Velocity adaptation for self-improvement of skills learned from user demonstrations", *13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Atlanta, GA, pp. 423-428.
- Newman, W., Branicky, M., Podgurski, H., Chhatpar, S., Huang, L., Swaminathan, J. and Zhang, H. (1999), "Force-responsive robotic assembly of transmission components", *IEEE International Conference on Robotics and Automation*, Detroit, MI, pp. 2096-2102.
- Pastor, P., Righetti, L., Kalakrishnan, M. and Schaal, S. (2011), "Online movement adaptation based on previous sensor experiences", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, pp. 365-371.
- Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Bergery, E., Wheeler, R. and Ng, A. (2009), "ROS: an open-source robot operating system", *ICRA Workshop on Open Source Software*, Kobe.
- Rozo, L., Jiménez, P. and Torras, C. (2013), "A robot learning from demonstration framework to perform forcebased manipulation tasks", *Intelligent Service Robotics*, Vol. 6 No. 1, pp. 33-51.

- Schaal, S., Mohajerian, P. and Ijspeert, A. (2007), "Dynamics systems vs optimal control – a unifying view", *Progress in Brain Research*, Vol. 165 No. 1.
- Schreiber, G., Stemmer, A. and Bischoff, R. (2010), "The fast research interface for the KUKA lightweight robot", *ICRA Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications – How to Modify and Enhance Commercial Controllers*, Anchorage, AK.
- Ude, A., Gams, A., Asfour, T. and Morimoto, J. (2010), "Tasks-specific generalization of discrete and periodic dynamic movement primitives", *IEEE Transactions on Robotics and Autonomous Systems*, Vol. 26 No. 5, pp. 800-815.
- Villani, L. and Schutter, J.D. (2008), "Force control", in Siciliano, B. and Khatib, O. (Eds), *Springer Handbook of Robotics*, Springer-Verlag, Berlin, Heidelberg.
- Xiao, J. (1997), "Goal-contact relaxation graphs for contact-based fine motion planning", *IEEE International Symposium on Assembly and Task Planning, Marina del Rey, CA*, pp. 25-30.
- Yun, S.-K. (2008), "Compliant manipulation for peg-in-hole: Is passive compliance a key to learn contact motion?", *IEEE International Conference on Robotics and Automation*, Pasadena, CA, pp. 1647-1652.

About the authors



Fares J. Abu-Dakka received a BSc in Mechanical Engineering from the Birzeit University, Palestine, in 2003, MSc and PhD in Manipulators Motion Planning from the Polytechnic University of Valencia (UPV), Spain, in 2011. He was awarded the FPI Fellowship from the UPV and a Research Fellowship Grant from the Biomechanical Institute of Valencia. His research activities include robot motion planning, robot dynamics, biomechanical researches, robot control and learning. Fares J. Abu-Dakka is the corresponding author and can be contacted at: fares.abudakka@gmail.com



Bojan Nemec is Senior Research Associate at Humanoid and Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute. He received BSc, MSc and PhD from the University of Ljubljana. He spent his sabbatical leave at the Institute for Real-Time Computer Systems and Robotics,

University of Karlsruhe. His research interests include robot control, robot learning, sensor-guided control, service robots and biomechanical measurements in sport.



Aljaž Kramberger received his master's degree in Mechatronics from the Faculty of Mechanical Engineering at the University of Maribor, Slovenia, in 2013. He is currently a PhD student at the Faculty of electrical Engineering, University of Ljubljana. He works as a researcher at the Department of Automation, Biocybernetics and Robotics at Jožef Stefan Institute, Ljubljana. His research interests include force-based robot interaction with environment and humanoid robotics.



Anders Glent Buch received his MSc from the Maersk Mc-Kinney Moller Institute, University of Southern Denmark, in 2010. He is currently pursuing his PhD on efficient methods for 3D shape-based matching. During the course of this, he has been a visiting researcher at Tampere University of Technology, Finland. He has published in notable conferences within the fields of computer vision and robotics. His research activities include computer vision, 3D perception, computational geometry and robot control.



Norbert Krüger is a Professor at the Mærsk McKinney Møller Institute, University of Southern Denmark. He holds an MSc from the Ruhr- Universität Bochum, Germany, and a PhD from the University of Bielefeld. He leads the Cognitive Vision Lab which focuses on computer vision and cognitive systems, in particular the learning of object representations in the context of grasping.



Aleš Ude received a degree in Applied Mathematics from the University of Ljubljana, Slovenia, and a PhD from the Faculty of Informatics, University of Karlsruhe, Germany. He was awarded the Science and Technology Agency fellowship for postdoctoral studies in ERATO Kawato Dynamic Brain Project, Japan. He is currently the head of Humanoid and Cognitive Robotics Lab at Jožef Stefan Institute, Ljubljana. His research interests include autonomous robot learning, imitation learning, humanoid robot vision and humanoid robotics in general.