Zbornik 20. mednarodne multikonference

INFORMACIJSKA DRUŽBA - IS 2017

Zvezek H

Proceedings of the 20th International Multiconference

INFORMATION SOCIETY - IS 2017

Volume H

Robotika
Robotics

Uredila / Edited by
Andrej Gams, Aleš Ude

http://is.ijs.si

9.–13. oktober 2017 / 9–13 October 2017
Ljubljana, Slovenia

Zbornik 20. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2017

Zvezek H

Proceedings of the 20th International Multiconference

INFORMATION SOCIETY – IS 2017

Volume H

Robotika
Robotics

Uredila / Edited by

Andrej Gams, Aleš Ude

http://is.ijs.si

9. - 13. oktober 2017 / 9th – 13th October 2017
Ljubljana, Slovenia

Urednika:

Andrej Gams
Odsek za avtomatiko, biokibernetiko in robotiko
Institut »Jožef Stefan«, Ljubljana

Aleš Ude
Odsek za avtomatiko, biokibernetiko in robotiko
Institut »Jožef Stefan«, Ljubljana

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič, Lana Zemljak
Oblikovanje naslovnice: Vesna Lasič

Dostop do e-publikacije:
http://library.ijs.si/Stacks/Proceedings/InformationSociety

Ljubljana, oktober 2017

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani
COBISS.SI-ID=30874151
ISBN 978-961-264-119-1 (pdf)

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2017

Multikonferenca Informacijska družba (http://is.ijs.si) je z **dvajseto** zaporedno prireditvijo osrednji srednjeevropski dogodek na področju informacijske družbe, računalništva in informatike. Letošnja prireditev je ponovno na več lokacijah, osrednji dogodki pa so na Institutu »Jožef Stefan«.

Informacijska družba, znanje in umetna inteligenca so spet na razpotju tako same zase kot glede vpliva na človeški razvoj. Se bo eksponentna rast elektronike po Moorovem zakonu nadaljevala ali stagnirala? Bo umetna inteligenca nadaljevala svoj neverjetni razvoj in premagovala ljudi na čedalje več področjih in s tem omogočila razcvet civilizacije, ali pa bo eksponentna rast prebivalstva zlasti v Afriki povzročila zadušitev rasti? Čedalje več pokazateljev kaže v oba ekstrema – da prehajamo v naslednje civilizacijsko obdobje, hkrati pa so planetarni konflikti sodobne družbe čedalje težje obvladljivi.

Letos smo v multikonferenco povezali dvanajst odličnih neodvisnih konferenc. Predstavljenih bo okoli 200 predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic. Prireditev bodo spremljale okrogle mize in razprave ter posebni dogodki, kot je svečana podelitev nagrad. Izbrani prispevki bodo izšli tudi v posebni številki revije Informatica, ki se ponaša s **40-letno** tradicijo odlične znanstvene revije. Odlične obletnice!

Multikonferenco Informacijska družba 2017 sestavljajo naslednje samostojne konference:

- Slovenska konferenca o umetni inteligenci
- Soočanje z demografskimi izzivi
- Kognitivna znanost
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Izkopavanje znanja in podatkovna skladišča
- Vzgoja in izobraževanje v informacijski družbi
- Četrta študentska računalniška konferenca
- Delavnica »EM-zdravje«
- Peta mednarodna konferenca kognitonike
- Mednarodna konferenca za prenos tehnologij - ITTC
- Delavnica »AS-IT-IC«
- Robotika

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi tudi ACM Slovenija, SLAIS, DKZ in druga slovenska nacionalna akademija, Inženirska akademija Slovenije (IAS). V imenu organizatorjev konference se zahvaljujemo združenjem in inštitucijam, še posebej pa udeležencem za njihove dragocene prispevke in priložnost, da z nami delijo svoje izkušnje o informacijski družbi. Zahvaljujemo se tudi recenzentom za njihovo pomoč pri recenziranju.

V 2017 bomo petič podelili nagrado za življenjske dosežke v čast Donalda Michija in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe bo prejel prof. dr. Marjan Krisper. Priznanje za dosežek leta bo pripadlo prof. dr. Andreju Brodniku. Že šestič podeljujemo nagradi »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je dobilo padanje slovenskih sredstev za akademsko znanost, tako da smo sedaj tretji najslabši po tem kriteriju v Evropi, jagodo pa »e-recept«. Čestitke nagrajencem!

Bojan Orel, predsednik programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2017

In its **20th year**, the Information Society Multiconference (http://is.ijs.si) remains one of the leading conferences in Central Europe devoted to information society, computer science and informatics. In 2017 it is organized at various locations, with the main events at the Jožef Stefan Institute.

The pace of progress of information society, knowledge and artificial intelligence is speeding up, and it seems we are again at a turning point. Will the progress of electronics continue according to the Moore's law or will it start stagnating? Will AI continue to outperform humans at more and more activities and in this way enable the predicted unseen human progress, or will the growth of human population in particular in Africa cause global decline? Both extremes seem more and more likely – fantastic human progress and planetary decline caused by humans destroying our environment and each other.

The Multiconference is running in parallel sessions with 200 presentations of scientific papers at twelve conferences, round tables, workshops and award ceremonies. Selected papers will be published in the Informatica journal, which has **40 years** of tradition of excellent research publication. These are remarkable achievements.

The Information Society 2017 Multiconference consists of the following conferences:

- Slovenian Conference on Artificial Intelligence
- Facing Demographic Challenges
- Cognitive Science
- Collaboration, Software and Services in Information Society
- Data Mining and Data Warehouses
- Education in Information Society
- 4th Student Computer Science Research Conference
- Workshop Electronic and Mobile Health
- 5th International Conference on Cognitonics
- International Conference of Transfer of Technologies - ITTC
- Workshop »AC-IT-IC«
- Robotics

The Multiconference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, i.e. the Slovenian chapter of the ACM, SLAIS, DKZ and the second national engineering academy, the Slovenian Engineering Academy. In the name of the conference organizers we thank all the societies and institutions, and particularly all the participants for their valuable contribution and their interest in this event, and the reviewers for their thorough reviews.

For the fifth year, the award for life-long outstanding contributions will be delivered in memory of Donald Michie and Alan Turing. The Michie-Turing award will be given to Prof. Marjan Krisper for his life-long outstanding contribution to the development and promotion of information society in our country. In addition, an award for current achievements will be given to Prof. Andrej Brodnik. The information lemon goes to national funding of the academic science, which degrades Slovenia to the third worst position in Europe. The information strawberry is awarded for the medical e-recipe project. Congratulations!

Bojan Orel, Programme Committee Chair
Matjaž Gams, Organizing Committee Chair

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, South Korea
Howie Firth, UK
Olga Fomichova, Russia
Vladimir Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Israel
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Karl Pribram, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Australia
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, Chicago, USA
Toby Walsh, Australia

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki
Mitja Lasič
Robert Blatnik
Aleš Tavčar
Blaž Mahnič
Jure Šorn
Mario Konecki

Programme Committee

Bojan Orel, chair
Franc Solina, co-chair
Viljan Mahnič, co-chair
Cene Bavec, co-chair
Tomaž Kalin, co-chair
Jozsef Györkös, co-chair
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Mitja Luštrek
Marko Grobelnik
Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič
Grega Repovš
Ivan Rozman
Niko Schlamberger
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček
Andrej Ule
Tanja Urbančič
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah

Invited lecture

AN UPDATE FROM THE AI & MUSIC FRONT

Gerhard Widmer
Institute for Computational Perception
Johannes Kepler University Linz (JKU), and
Austrian Research Institute for Artificial Intelligence (OFAI), Vienna

Abstract
Much of current research in Artificial Intelligence and Music, and particularly in the field of Music Information Retrieval (MIR), focuses on algorithms that interpret musical signals and recognize musically relevant objects and patterns at various levels -- from notes to beats and rhythm, to melodic and harmonic patterns and higher-level segment structure --, with the goal of supporting novel applications in the digital music world. This presentation will give the audience a glimpse of what musically "intelligent" systems can currently do with music, and what this is good for. However, we will also find that while some of these capabilities are quite impressive, they are still far from (and do not require) a deeper "understanding" of music. An ongoing project will be presented that aims to take AI & music research a bit closer to the "essence" of music, going beyond surface features and focusing on the expressive aspects of music, and how these are communicated in music. This raises a number of new research challenges for the field of AI and Music (discussed in much more detail in [Widmer, 2016]). As a first step, we will look at recent work on computational models of expressive music performance, and will show some examples of the state of the art (including the result of a recent musical 'Turing test').

References
Widmer, G. (2016).
Getting Closer to the Essence of Music: The Con Espressione Manifesto.
ACM Transactions on Intelligent Systems and Technology 8(2), Article 19.

KAZALO / TABLE OF CONTENTS

Zbornik 20. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2017

Zvezek H

Proceedings of the 20th International Multiconference

INFORMATION SOCIETY – IS 2017

Volume H

Robotika
Robotics

Uredila / Edited by

Andrej Gams, Aleš Ude

http://is.ijs.si

13. oktober 2017 / 13th October 2017
Ljubljana, Slovenia

PREDGOVOR

Pod okriljem multikonference »Informacijska družba« po krajšem premoru zopet organiziramo tudi konferenco Robotika, s katero nadaljujemo tradicijo raziskovalne robotike v Sloveniji.

Robotika je v vzponu in čeprav jo mnogi še zmeraj dojemajo kot znanstveno fantastiko, je tudi uporabniška robotika že nekaj časa nekaj povsem realnega in oprijemljivega, kmalu pa bo tudi že nekaj običajnega. Robotika je tudi skorajda vseprisotna. Brez robotskih manipulatorjev si ne znamo več predstavljati sodobnih industrijskih procesov. Ne presenečajo niti kirurški roboti ali servisni mobilni roboti, ki dostavljajo pakete in hrano ter čistijo in stražijo javno infrastrukturo. Domišljija in pa želje ljudi ne poznajo mej, zato se raziskovalna robotika trudi z razvojem velikih večnamenskih robotskih hišnih pomočnikov. Pri razvoju tako kompleksnih in avtonomnih sistemov, kar nekateri ocenjujejo, da je težje kot raketna znanost, je pomembna izmenjava idej in mnenj, kar je tudi namen konference Robotika.

V zborniku so zbrani prispevki raziskovalcev Odseka za avtomatiko, biokibernetiko in robotiko na Inštitutu Jožef Stefan, veseli pa smo, da imamo letos prispevke s svetovno priznane Tehniške Univerze v Muenchnu, Nemčija. Upamo, da bo izmenjava idej in raziskovalnih rezultatov vodila v nadaljnje skupne podvige, ki bodo še naprej pomagali soustvarjati trende raziskovalne robotike.

Andrej Gams in Aleš Ude

FOREWORD

Robotics conference in the scope of the Information Society is after a short break again a part of the multiconference, and continues the rich tradition of research robotics in Slovenia.

Robotics is on the rise and even though many people still perceive it as science fiction, even consumer robotics has passed from the realm of fiction to something real, tangible. Robotics is also omnipresent. Many industrial processes today simply cannot be conceived without the use of robotic manipulators. The use of surgical and mobile service robots, which deliver packages and food and clean and guard public infrastructure is not a surprise anymore. As human imagination and wishes do not know any borders, research robotics is working hard towards the development of multipurpose, autonomous, robotic household assistants. The development of such systems, which some consider more complex than rocket science, requires cooperation between researchers and the exchange of ideas and opinions. Exchange of ideas and opinions is also the main aim and goal of the Robotics conference in the scope of the Information Society multiconference.

The conference proceedings contain papers from researchers of the Department for Automatics, Biocybernetics and Robotics of Jožef Stefan Institute. We are delighted to have attracted contributions from researchers of the world-renowned Technical University of Munich, Germany. We hope that the exchange of ideas will lead to joint undertakings and will help to co-shape the trends of research robotics in the future.

Andrej Gams and Aleš Ude

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Andrej Gams

Aleš Ude

Compliant Bimanual Actions through Learning of Primitives

Andrej Gams
Humanoid and Cognitive Robotics Lab
Dept. of Automatics, Biocybernetics and Robotics
Jožef Stefan Institute
andrej.gams@ijs.si

Aleš Ude
Humanoid and Cognitive Robotics Lab
Dept. of Automatics, Biocybernetics and Robotics
Jožef Stefan Institute
ales.ude@ijs.si

ABSTRACT

Compliant Movement Primitives (CMP) provide the means to achieve low trajectory errors and compliant control without explicit dynamic models of the task. This paper addresses the application of CMPs to bimanual tasks. Besides performing the task, the robots have to maintain their relative posture even in the presence of external perturbations on any of the robots, not to exert force on the object they are carrying. Thus, they act compliantly in their absolute task, but remain stiff in their relative task. For compliant absolute behavior and stiff relative behavior we combine previously developed joint-space CMPs with the symmetric control approach. We further augment it by applying a virtual force vector at the end-effector, calculated through the measured external joint torques on the perturbed robot, to increase bimanual compliance. Experiments with two Kuka LWR-4 robots in a bimanual setting show applicability of the system.

1. INTRODUCTION

Robots were considered dangerous to humans and objects in their workspace and were thus confined to cages [7]. This is due to high stiffness and position control used to accomplish accurate execution of their given tasks. However, the notion of collaborative robotics, where both the human and robot share their workspace to accomplish a common task [8], has gone beyond that. Collaborative robotics applications go beyond the factory work-floor to everyday human environments, including bimanual and humanoid robots.

Safety of the human is the primary task in shared environments. This can be ensured through the compliance of the robot. Compliance can be active, originating from contact detection with an artificial tactile skin [11]. On the other hand, elastic elements can introduce passive compliance. Passive compliance can also be actively changed using variable stiffness actuators [1]. Appropriate active torque strategies, relying on comparing the actual torques and the required theoretical torques [9] have also been used to implement active compliance. However, such methods require the correct dynamic model of both the robot and of the task. Such models of task dynamics are often often difficult to derive.

To bypass the need to develop dynamical models, one can learn the specific required torques for the specific task with learning by demonstration (LbD). Learned torques are applied for the repetition of the exact same action. Such an approach was utilized in [5] and termed Compliant Movement Primitives (CMPs). It is applicable to robots with active torque control. In this paper we show how we can extend the CMP framework to make it applicable for bimanual task. For bimanual operation, we need to maintain the relative task. We implement this by integrating the symmetric controller into the framework. For increased compliance, which allows safe physical interaction with humans, we also utilize virtual force translation, where we copy the perturbation to the robot from one robotic arm to the other. The experimental system used is presented in Fig. 1.

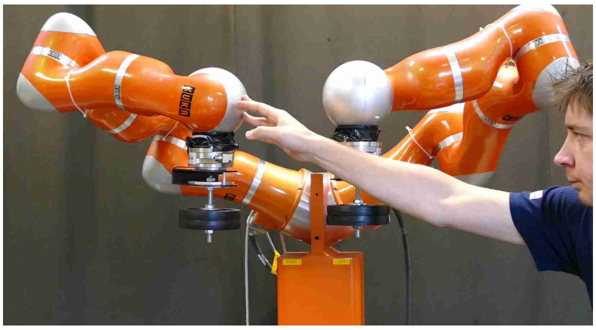


Figure 1: Bimanual robot performing a task while being perturbed by a human.

1.1 Related Work

Many papers have dealt with the topics of compliant control. It commonly relies on explicit dynamics of the robot and the task [13]. Compliant movement primitives, which represent the basic background of this paper, mitigate this need through learning of required torques. Similar approaches that rely on task-specific models have been presented. [3] used tactile sensors to determine the force of contact with the environment on the iCub robot, and then calculate the joint-torques from the measured arm pose. The calculated joint torques were used in a feed-forward manner in control. In an analogous manner in [15] the authors recorded joint torques along the kinematic trajectory and then used as the feed-forward signal for increased accuracy in the next execution of the same in-contact task.

Different approaches for bimanual control of robots have also been presented, with a basic separation in being either asymmetric or symmetric control. While the former controls each robot independently, the latter considers both robots as a single system. For example, in [6] an asymmetric control scheme using motion primitives is described. The robots are coupled through learned feed-forward signals. However, the robots themselves are completely stiff. A dynamical system that combines single or bimanual robot operation based on dynamical systems was presented in [14]. The authors use a virtual object to define the motion of robotic arms.

Asymmetric system, on the other hand, can describe a cooperative operational space. Thus it allows the user to determine the relative and the absolute tasks. This also allows to define geometrical variables at the position/orientation level [4]. An example of such is presented in [12], showing a human-robot cooperation scheme for bimanual robots. It is based on separately defining the gains for absolute and relative motion. However, trajectory tracking errors will increase considerably when the absolute gains are set low.

2. COMPLIANT MOVEMENT PRIMITIVES

Compliant movement primitives rely on the impedance controller, such as the one used in the Kuka LWR robot [2], and add feed-forward torques $\vec{\tau}_{ff}$. Thus, the low-level robot controller is given by

$$\vec{\tau}_u = \mathbf{K}_q(\vec{q}_d - \vec{q}) + \mathbf{D}_q(\dot{\vec{q}}_d - \dot{\vec{q}}) + \vec{f}_{\text{dynamic}}(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}) + \vec{\tau}_{ff}. \quad (1)$$

Here $\vec{\tau}_u$ is the control torque vector, $\mathbf{K}_q$ is a diagonal joint-stiffness matrix, $\vec{q}_d$ and $\vec{q}$ are vectors of the desired and measured joint positions, respectively, $\mathbf{D}_q$ is a diagonal damping matrix, $\dot{\vec{q}}_d$ and $\dot{\vec{q}}$ are the desired and measured vectors of joint velocities, respectively, and $\vec{f}_{\text{dynamic}}(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}})$ represents the robot dynamics and the non-linearities occurring in the robot.

If the robot is made compliant by lowering the stiffness ($\mathbf{K}_q$), this increases the trajectory tracking error. To compensate, feed-forward torques $\vec{\tau}_{ff}$ are added to the motor torque to preserve trajectory tracking. These feed-forward torques $\vec{\tau}_{ff}$ are usually calculated from an explicit dynamical model. However, for repeatable tasks, we can use previously learned torques to provide low trajectory tracking errors. That is the basic principle of the the CMP framework [5].

A CMP combines desired joint motion trajectories (joint positions $\vec{q}_d(t)$) and corresponding joint torque signals $\vec{\tau}_{ff}(t)$

$$\vec{h}(t) = [\vec{q}_d(t), \vec{\tau}_{ff}(t)]. \quad (2)$$

Joint positions for all degrees-of-freedom (DOF) are learned, for example through imitation, while joint torques are recorded from a stiff execution. Because CMPs encode only task-specific torques, we gain them by subtracting the known robot's $\vec{f}_{\text{dynamic}}(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}})$ from the actual measured torques $\vec{\tau}_m$ at robots joints

$$\vec{\tau}_{ff} = \vec{\tau}_m - \vec{f}_{\text{dynamic}}(\vec{q}, \dot{\vec{q}}, \ddot{\vec{q}}). \quad (3)$$

Joint positions are encoded as dynamic movement primitives (DMPs) [10] and the corresponding torques are encoded as a combination of radial basis functions (RBFs). We again refer the reader to [5] for details.

CMPs operate in joint space. For bimanual tasks, the relation of the robots is in task space.

3. SYMMETRIC ROBOT CONTROL

In this paper we provide a reduced description for the control of a bimanual system. The absolute coordinates describe the position and orientation of a common coordinate frame (CF) of the robots in reference to the inertial CF. The relative ones (6 DOF) describe the position and orientation of one robot end-effector relative to the other.

The Jacobian matrix of a bimanual system includes both the absolute and the relative parts.

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{abs} \\ \mathbf{J}_{rel} \end{bmatrix}. \quad (4)$$

We can iteratively calculate the inverse kinematics using

$$\dot{\vec{q}} = \mathbf{J}^{\dagger} \left(\vec{v}_d + \mathbf{K}\vec{e} \right), \quad (5)$$

where $\mathbf{J}^{\dagger}$ is the Moore-Penrose pseudo-inverse of the Jacobian matrix from (4). In (5) $\dot{\vec{q}} = \begin{bmatrix} \dot{\vec{q}}_1^T & \dot{\vec{q}}_2^T \end{bmatrix}^T$ is the vector of angular velocities, $\vec{e} = \begin{bmatrix} \vec{e}_{abs}^T & \vec{e}_{rel}^T \end{bmatrix}^T$ is the vector of task space errors, $\vec{v}_d = \begin{bmatrix} \vec{v}_{absd}^T & \vec{v}_{reld}^T \end{bmatrix}^T$ are the desired task space velocities and $\mathbf{K}$ is a 12×12 diagonal gain matrix. The absolute error is given by

$$\vec{e}_{abs} = \begin{bmatrix} \vec{p}_{absd}^{w} - \vec{p}_{abs}^{w} \\ \frac{1}{2} \left(\mathbf{S}\left(\vec{n}_{abs}^{w}\right) \vec{n}_{absd}^{w} + \mathbf{S}\left(\vec{s}_{abs}^{w}\right) \vec{s}_{absd}^{w} + \right. \\ \left. + \mathbf{S}\left(\vec{a}_{abs}^{w}\right) \vec{a}_{absd}^{w} \right) \end{bmatrix} \quad (6)$$

For the relative coordinates we have

$$\vec{e}_{rel} = \begin{bmatrix} \mathbf{R}_{abs}^{w} \vec{p}_{reld}^{abs} - \vec{p}_{r}^{w} \\ \frac{1}{2} \mathbf{R}_1^{w} \left(\mathbf{S}\left(\vec{n}_{rel}^{1}\right) \vec{n}_{reld}^{1} + \mathbf{S}\left(\vec{s}_{rel}^{1}\right) \vec{s}_{reld}^{1} + \right. \\ \left. + \mathbf{S}\left(\vec{a}_{rel}^{1}\right) \vec{a}_{reld}^{1} \right) \end{bmatrix} \quad (7)$$

Desired velocities are then

$$\vec{v}_{absd} = \begin{bmatrix} \dot{\vec{p}}_{absd}^{w} \\ \vec{\omega}_{absd}^{w} \end{bmatrix} \quad (8)$$

$$\vec{v}_{reld} = \begin{bmatrix} \mathbf{R}_{abs}^{w} \dot{\vec{p}}_{reld}^{abs} + \mathbf{S}\left(\vec{\omega}_{abs}^{w}\right) \mathbf{R}_{abs}^{w} \vec{p}_{reld}^{abs} \\ \vec{\omega}_{reld}^{1} \end{bmatrix} \quad (9)$$

where the subscript suffix d stands for desired, $\mathbf{S}(\cdot)$ is the skew-symmetric operator and $\vec{n}_i^j, \vec{s}_i^j, \vec{a}_i^j$ are, respectively, the first, second and third column of a rotation matrix. i.e. $\mathbf{R}_i^j = \begin{bmatrix} \vec{n}_i^j & \vec{s}_i^j & \vec{a}_i^j \end{bmatrix}$.

To control the joint torques of a bimanual system, we can now use

$$\vec{\tau}_{\text{biman}} = \mathbf{J}^T \left(\mathbf{K}_{\text{t}} \vec{e} + \mathbf{D}_{\text{t}} \dot{\vec{e}} \right). \quad (10)$$

Here $\mathbf{K}_{\text{t}}$ and $\mathbf{D}_{\text{t}}$ are diagonal gain matrices for stiffness and damping, respectively. Low values on their diagonal will result in compliant behavior, resulting in poor trajectory tracking and high errors.

The drawback of the controller (10) is that it changes the torques of both manipulators. By pushing on one robot, additional torques will appear in both of them to neutralize the perturbation. An unintentional collision will thus result in less compliant behavior of the bimanual system. We increase the compliancy of the bimanual system, by additionally introducing a virtual force translation.

From measured joint torques we can calculate the end-effector force using the virtual work theorem, which states

$$\vec{\tau_e} = \mathbf{J}^T \vec{f_e}. \tag{11}$$

A perturbation on one robot can thus be used to estimate the end-effector force of that manipulator $\vec{f}_{1e}$ using (11). We apply the same end-effector force through the joint torques to the other robot. Thus we have

$$\vec{f}_{1e} = \vec{f}_{2e}. \tag{12}$$

Only the virtual torques caused by the perturbation should be translated to the other robot, not the complete joint torques. These are calculated by

$$\Delta\vec{\tau}_i = \vec{\tau}_{i\_\text{expected}} - \vec{\tau}_{i\_\text{measured}}, \; i = 1, 2. \tag{13}$$

The translated torque is thus

$$\vec{\tau}_{\text{vft}} = \begin{bmatrix} \vec{\tau}_{\text{vft},1} \\ \vec{\tau}_{\text{vft},2} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1^T \left(\mathbf{J}_2^\dagger\right)^T \Delta\vec{\tau}_2 \\ \mathbf{J}_2^T \left(\mathbf{J}_1^\dagger\right)^T \Delta\vec{\tau}_1 \end{bmatrix}. \tag{14}$$

4. BIMANUL CONTROL USING CMPS

If we use low gains in (10), we introduce introduce compliance to the system, but also high trajectory tracking errors. Therefore we need to add also the feed-forward torques. Feed-forward torques $\vec{\tau}_{ff}$, which we use in (1), are now composed of three components

$$\vec{\tau}_{ff} = \begin{bmatrix} \vec{\tau}_{ff,1} \\ \vec{\tau}_{ff,2} \end{bmatrix} = \vec{\tau}_{\text{rec}} + \vec{\tau}_{\text{biman}} - \vec{\tau}_{\text{vft}}. \tag{15}$$

The pre-recorded or learned task torque $\vec{\tau}_{\text{rec}}$ ensures trajectory tracking. It is the direct output of the CMP. However, the reference joint trajectories are calculated from the task-space trajectories using (5). Note that the inverse kinematics solution needs to match the posture of the robot during the demonstration, which might become a problem with redundant tasks.

5. EXPERIMENTAL EVALUATION

We performed our experiments on two Kuka LWR-4 7 DOF robots. We locked the rotation of the 3rd axis on both robots, so that our system was not redundant for the task. The system was controlled from Matlab. The robots were controlled with (1), with joint stiffness set to 25 Nm/rad for all the used joints.

The task of the bimanual system was to perform a bimanual trajectory while being compliant. This means that the trajectory tracking error was low when there were no perturbations, but the robot was compliant in the absolute task when a perturbation occured. The robots were each carrying a 2.5 kg load. The robots were not physically coupled through holding a common object, so that the relative task conformity is clearly expressed. In this experiment we controlled the relative position of the system, but did not control the relative orientation.

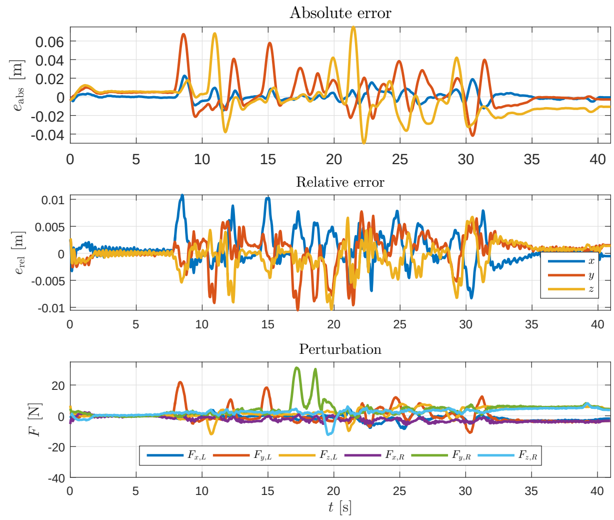


Figure 2: Absolute error (top), relative error (middle) and end-effector perturbation (calculated from measured joint torques) when using the complete controller, given by (15).

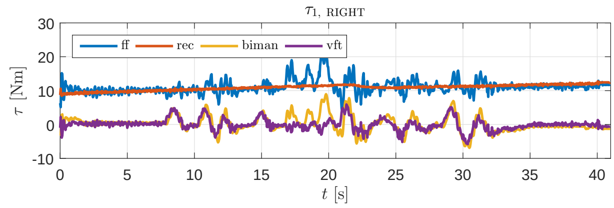


Figure 3: The feed-forward torque on the first joint of the right robot, and separate components when using the complete controller, given by (15). See the bottom plot of Fig. 2 for the perturbation.

The result of using the complete controller, given by (15) is compliant operation in the absolute task, but maintains low errors in the relative task despite the high external perturbation forces on the one side, but also maintaining low trajectory tracking errors if no perturbation is present. The results are shown in Fig. 2. The effect of including the virtual force translation in $\vec{\tau}_{ff}$ is seen also at seconds 18 – 20 in the bottom plot of Fig. 2. There, the right robot did not include $\vec{\tau}_{\text{vft}}$. Thus, a twice higher perturbation resulted in a much lower absolute error, meaning that the system was far less compliant, when $\vec{\tau}_{\text{vft}}$ was not included.

Figure 3 shows the complete torque $\tau_{1,\text{RIGHT}}$ and the contributions of separate components of the controller for the first joint of the right robot. It is evident that $\tau_{\text{vft},1}$ and $\tau_{\text{biman},1}$ are similar. Thus, when perturbing the left robot, the perturbation is not fighting $\vec{\tau}_{\text{biman}}$ of the right robot, because it only has to account for a much smaller relative error. A

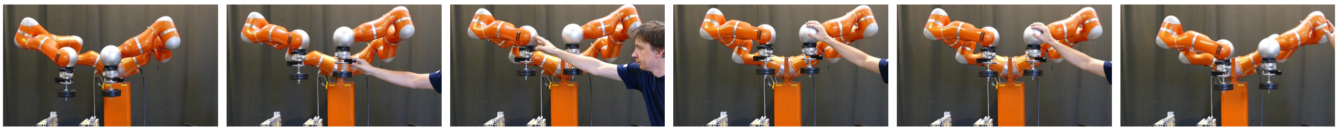


Figure 4: Bimanual robotic system performing the experimental compliant bimanual task. A person was perturbing the motion.

small relative error remains due to different postures of the robots that make force-vector copy inaccurate. We can also see that when there is no perturbation, the contribution of the bimanual symmetric torque controller and of the virtual force translation is practically 0. The plot also shows that $\tau_{\text{rec},1}$ is the actual learned feed-forward torque, while the other two react to perturbations.

Figure 4 shows a series of still photos showing the bimanual execution and physical interaction of the bimanual system with a person.

6. CONCLUSION

In this paper we have shown how one can extend the compliant movement primitives framework to include bimanual task execution. In the experiments, the robots were not physically coupled through holding a rigid object, yet we showed that the system maintains the relative posture, but is compliant in the absolute coordinate frame. Without perturbations, the system maintains low tracking error and that is the real contribution of compliant movement primitives.

The presented approach can be applied to a specific, prelearned task. On the other hand, generalization has the potential to extend it for a wider region of operations. That remains to be researched in the future. Another topic for the future is dealing with redundant robots. When the robots are redundant for the task, kinematic mapping offers numerous solutions. Learning of torques for all solutions is not viable, as there could literally be infinite. The posture of the robots needs to be maintained to match the posture of learning the torques. Going beyond this with CMPs remains an open research question.

Acknowledgments

This work was supported by the Slovenian Research Agency under grant agreement no. J2-7360.

7. REFERENCES

[1] A. Bicchi and G. Tonietti. Fast and "soft-arm" tactics [robot arm design]. *IEEE Robot. Automat. Mag*, 11(2):22–33, 2004.

[2] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger. The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing. *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–8, 2010.

[3] R. Calandra, S. Ivaldi, M. Deisenroth, and J. Peters. Learning torque control in presence of contacts using tactile sensing from robot skin. In *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 690–695, Nov 2015.

[4] P. Chiacchio and S. Chiaverini. *Kinematic control of dual-arm systems*. Springer, 1998.

[5] M. Deniša, A. Gams, A. Ude, and T. Petrič. Learning compliant movement primitives through demonstration and statistical generalization. *IEEE/ASME Transactions on Mechatronics*, PP, 2016.

[6] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude. Coupling movement primitives: Interaction with the environment and bimanual tasks. *IEEE Transactions on Robotics*, 30(4):816–830, Aug 2014.

[7] S. Haddadin. *Towards Safe Robots*. Number 90 in Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2014.

[8] S. Haddadin, A. Albu-Schaffer, and G. Hirzinger. Requirements for Safe Robots: Measurements, Analysis and New Insights. *The International Journal of Robotics Research*, 28(11-12):1507–1527, aug 2009.

[9] N. Hogan. Stable execution of contact tasks using impedance control. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 1047–1054, Raleigh, NC, USA, 1987.

[10] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural computation*, 25:328–73, 2013.

[11] P. Mittendorfer and G. Cheng. Humanoid multi-modal tactile sensing modules. *IEEE Trans. Robot.*, 27(3):401–410, 2011.

[12] B. Nemec, N. Likar, A. Gams, and A. Ude. Adaptive human robot cooperation scheme for bimanual robots. In J. Lenarcic and J. Merlet, editors, *Advances in Robot Kinematics*, pages 385–393. INRIA, 2016.

[13] J. Peters, K. Mülling, D. Kober, Jensand Nguyen-Tuong, and O. Krömer. *Robotics Research: The 14th International Symposium (ISRR)*, chapter Towards Motor Skill Learning for Robotics, pages 469–482. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[14] S. S. M. Salehian, N. Figueroa, and A. Billard. Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[15] F. Steinmetz, A. Montebelli, and V. Kyrki. Simultaneous kinesthetic teaching of positional and force requirements for sequential in-contact tasks. In *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 202–209, Nov 2015.

Collaborative Tasks Synthesis through a Hierarchical Database

Miha Deniša
Humanoid and Cognitive Robotics Lab.
Dept. of Automatics, Biocybernetics and Robotics
Jožef Stefan Institute
Ljubljana, Slovenia
miha.denisa@ijs.si

Aleš Ude
Humanoid and Cognitive Robotics Lab.
Dept. of Automatics, Biocybernetics and Robotics
Jožef Stefan Institute
Ljubljana, Slovenia
ales.ude@ijs.si

ABSTRACT
Collaborative human-robot task execution is presented in this paper. Example collaborative movements are encoded in a dual hierarchical database. The primary part of the database is used for human movement recognition, while the data in the secondary database is used to synthesize appropriate collaborative robot movements.

Keywords
Programming by Demonstration, Motion Recognition, Collaborative Tasks, Human-Robot Interaction

1. INTRODUCTION
By moving robots into an unstructured environment, programming its movements by hand becomes unfeasible. New movements can alternatively be gained through programming by demonstration (PbD) [14, 6], where a human demonstrates example movements. While multiple approaches can be used to record human movements [16, 12, 17], we use kinesthetic guidance, where a human physically moves the robot [8, 13].

Important aspect of using robots in a home environment or in a small to medium enterprise, is human-robot collaboration. In this two step process, human intentions are recognized first. Then an appropriate collaborative robot movement is synthesized and executed. Various approaches were used by several authors: Hidden Markov Models [10, 2]; extension of DMPs, called Interaction Primitives (IPs) [1]; Probabilistic Movement Primitives [11, 7]; etc. Similar to our approach, Yamane et al. [19, 18] used a binary tree database to recognize and synthesize movements.

Our presented approach is based on a dual hierarchical database of example collaborative movements. While the primary database encodes example human movements and is used for recognition, the second database encodes example corresponding robot movements and is used for cooperative movement synthesis. Previously, new movements were synthesize using a single hierarchical database [5]. An extended dual database was used to synthesize new compliant movement primitives [4]. Using a dual database for human robot collaborative tasks was preliminary evaluated through comparison [3].

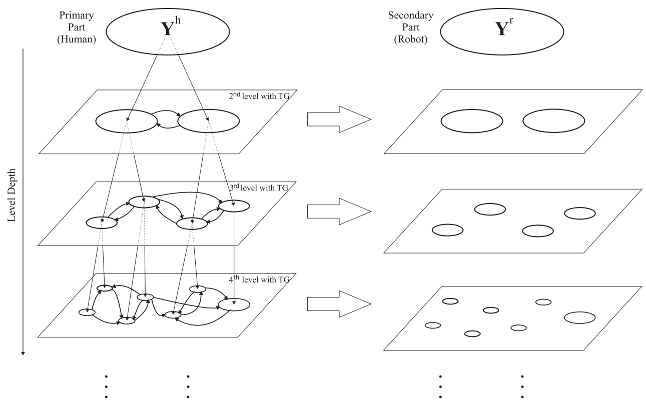


Figure 1: A simple representation of a dual database. The primary part, used for construction and motion recognition, can be seen on the left. The secondary part, seen on the right, stores corresponding robot movements.

2. METHODOLOGY
The first part of the proposed approach consists of dual hierarchical database construction. A simple representation of a dual hierarchical database is shown in Fig. 1. Its construction starts by demonstrating a set of n_S human-robot collaborative movements $\mathbf{D}$. Each one contains n_D state vectors,

$$\mathbf{D} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n_D}], \quad (1)$$

sampled at a given discrete time t_D. Each state vectors,

$$\mathbf{y}_i = [\mathbf{y}_i^h, \mathbf{y}_i^r]^T, \quad (2)$$

includes human state vectors $\mathbf{y}_i^h$, and corresponding robot state vectors $\mathbf{y}_i^r$. Human movements are used to build the primary part of the database. Clustering is used to construct multiple levels which encode all demonstrated movements at different granularities. Each level includes a weighted directed graph, i. e. a transition graph (TG). It is based on demonstrated movements and represents transitions between the nodes at that level. The secondary database encodes corresponding robot movements. At each level the nodes are mirrored from the primary database. This means that each node in the primary database has a mirrored node in the secondary database, which includes robot state vectors recorded at the same time as human state vectors.

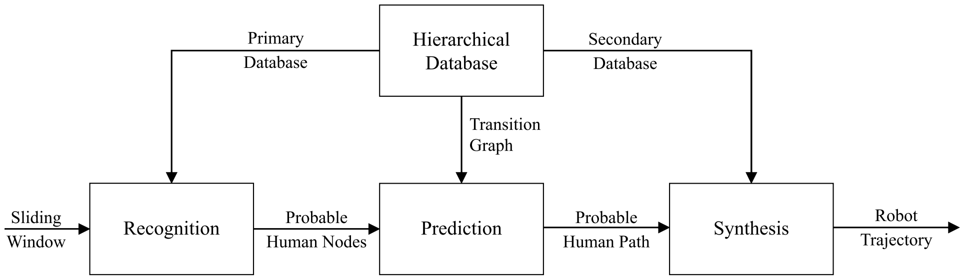


Figure 2: Recognition and cooperative movement synthesis overview.

The further details on hierarchical database construction are omitted and the reader is referred to [5, 3].

The second part of the proposed approach uses the dual database for human movement recognition and corresponding robot movement synthesis. The process, shown in Fig. 2, can be divided into three components: human movement *recognition*, which finds the most probable human nodes w.r.t. the current sliding window; human path *prediction*, which uses transition graphs to predict a full human path; and robot movement *synthesis*, which uses DMPs to interpolate between corresponding robot nodes and generates an executable participating robot trajectory.

Each recognition iteration starts by updating the sliding window, which consists of last n_W human state vectors observed in the current human movement,

$$\mathbf{Y}^o = \{\mathbf{y}^o_{n_o-n_W}, \dots, \mathbf{y}^o_{n_o-1}, \mathbf{y}^o_{n_o}\}, \quad (3)$$

where o represents *observed*, and the complete number of human state vectors observed from the start of the movement is denoted by n_o. At each iteration, recognition is done by traversing down the levels of the primary database. Several steps are done at every level l: **1)** determine the *considered* nodes at this level, i. e., the nodes who's parents were above the cut-off range at the previous level; **2)** build a matrix of considered nodes, where each row of length n_W is a permutation of considered nodes; **3)** calculate the recognition score for each permutation based on the TG and similarities between the observations and considered nodes; **4)** determine the nodes with recognition score above the cut-off range, who's children nodes will be considered as we move down a level. As we reach the last level, the permutation of nodes with the best recognition score is determined as the most probable human sequence of nodes.

Next component uses these results for robot sequence prediction. It follows the most probable human sequence on the TG w.r.t. highest weights until it reaches an end node. It then mirrors it to the secondary database and predicts the most suitable robot node sequence.

Last component is synthesis, which enhances the most suitable robot node sequence with time stamps and interpolates it. Time stamps are determined based on estimated duration of nodes. Time duration t_v of a single node v is estimated as

$$t_v = \frac{n_v}{m_v} t_D, \quad (4)$$

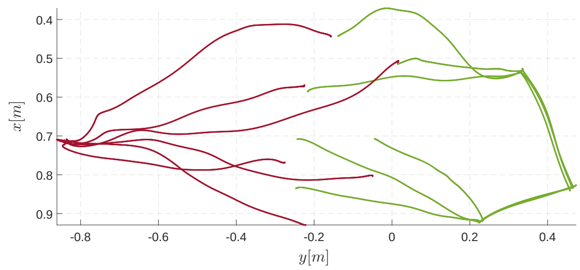


Figure 3: Set of demonstrated cooperative trajectories seen in two dimensions Robot trajectories, denoted in red, were converted to task space for presentation purposes. Human trajectories are denoted in green.

where n_v denotes the number of state vectors clustered in node v and m_v denotes the number of demonstrated trajectories passing through it. The interpolation of robot state vectors belonging to the most suitable robot node sequence, and enhanced with time stamps, is done with Dynamic Movement Primitives (DMPs). The details on DMPs are omitted and the reader is referred to [15, 9]. DMPs ensure smooth and continuous robot movements, even in the events of sudden recognition change.

3. EVALUATION

Evaluation was done on a robot system consisting of: a KUKA-LWR4 robot arm, a three-fingered Barret Hand gripper, and a passive marker based system OptiTrack. Six collaborating tasks were executed, with one demonstrator kinesthetically guiding the robot and the other executing the task with markers on his hand. Six human movements and corresponding robot movements are shown in Fig. 3. They are shown in two dimensions for presentational purposes. The robot movements were captured and encoded in the database in joint space. For this image they were converted to task space. The recorded set of collaborating movements can be divided into to two sets of three movements. Each set executes a variation of a pick and place task with one of two objects. The human part of the task was to grab up one of two object (peg or cover) and move it to one of three final position for each object. He then inserted/put

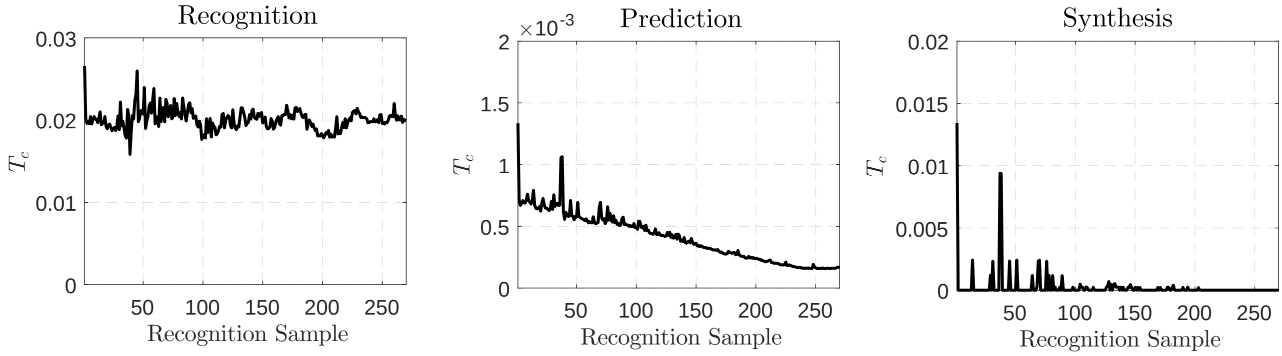


Figure 4: Computation time of components. Mean values of computation times over all 10 experiments are presented in the 3 graphs. They present computation times needed for separate components of the process.

the object in/on the cylinder. The robot's task, which was holding the cylinder, was to rotate it appropriately. As described in the previous section, the collaborative movements were used to build a dual hierarchical database.

Ten experiments were performed. In each the same dual database was used to recognize human movement and synthesize an appropriate collaborative robot movement. In each experiment the human performed a different movement: picking up an object and moving to one of the 6 final positions, picking up an object and while moving it changing its mind about the final position, and moving to pick one object, but then changing his mind and picking up the other one.

As our goal was to perform the recognition and synthesis on-line, we first evaluated the needed computation time of the approach. Mean values of calculation times over all 10 experiments presented in Fig. 4 are divided over the components: recognition of human movement, robot node sequence prediction, and robot movement synthesis. We can observe the majority of the computation time is used for recognition component. With the exception of the occasional rise above the desired 30 Hz, the sum computation times of all three components, remained approximately 0.02 s.

Images in Fig. 5 show two example executions of on-line movement recognition and cooperative task execution. Top 6 images show the human picking up the cover and placing it on the object. The bottom 6 images show another example, where the human moves the peg. We can observe the robot executing appropriate collaborative movements.

4. CONCLUSIONS

Human movement recognition and collaborative robot movements synthesis was presented in this paper. A set of human-robot collaborative movements was demonstrated and encoded in a dual hierarchical database. While the primary part was used for human movement recognition, the secondary part was used for collaborative robot movement synthesis. Incorporated DMP approach ensures a smooth and continuous trajectory for its execution on a robot.

Performed experiments showed on-line execution was viable, due to a satisfactory computation time. In all ten experiments the human movement was recognized and an appropriate collaborative robot movement was synthesized. All robot movement were smooth and continuous.

Future work involves further evaluation, including responses to sudden human movement changes. The approach will also be evaluated with a bigger set of more various demonstrated collaborative movements. Computation time needed for the recognition component could be further reduced to make the whole process perform faster.

5. REFERENCES

[1] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters. Interaction primitives for human-robot cooperation tasks. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2831–2837. IEEE, 2014.

[2] H. B. Amor, D. Vogt, M. Ewerton, E. Berger, B. Jung, and J. Peters. Learning responsive robot behavior by imitation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3257–3264. IEEE, 2013.

[3] M. Deniša, B. Nemec, and A. Ude. Cooperative movements through hierarchical database search. In *Advanced Robotics (ICAR), 2017 18th International Conference on*, pages 40–46. IEEE, 2017.

[4] M. Deniša, T. Petrič, T. Asfour, and A. Ude. Synthesizing compliant reaching movements by searching a database of example trajectories. In *In Proc. IEEE Int Conf on Humanoid Robots*, pages 540–543, Atlanta, GA, USA, 2013.

[5] M. Deniša and A. Ude. Synthesis of new dynamic movement primitives through search in a hierarchical database of example movements. *Int J Adv Robot Syst.*, 12(137), 2015.

[6] R. Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robot. Auton. Syst.*, 47(2):109–116, 2004.

[7] M. Ewerton, G. Neumann, R. Lioutikov, H. B. Amor, J. Peters, and G. Maeda. Learning multiple collaborative tasks with a mixture of interaction primitives. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1535–1542.

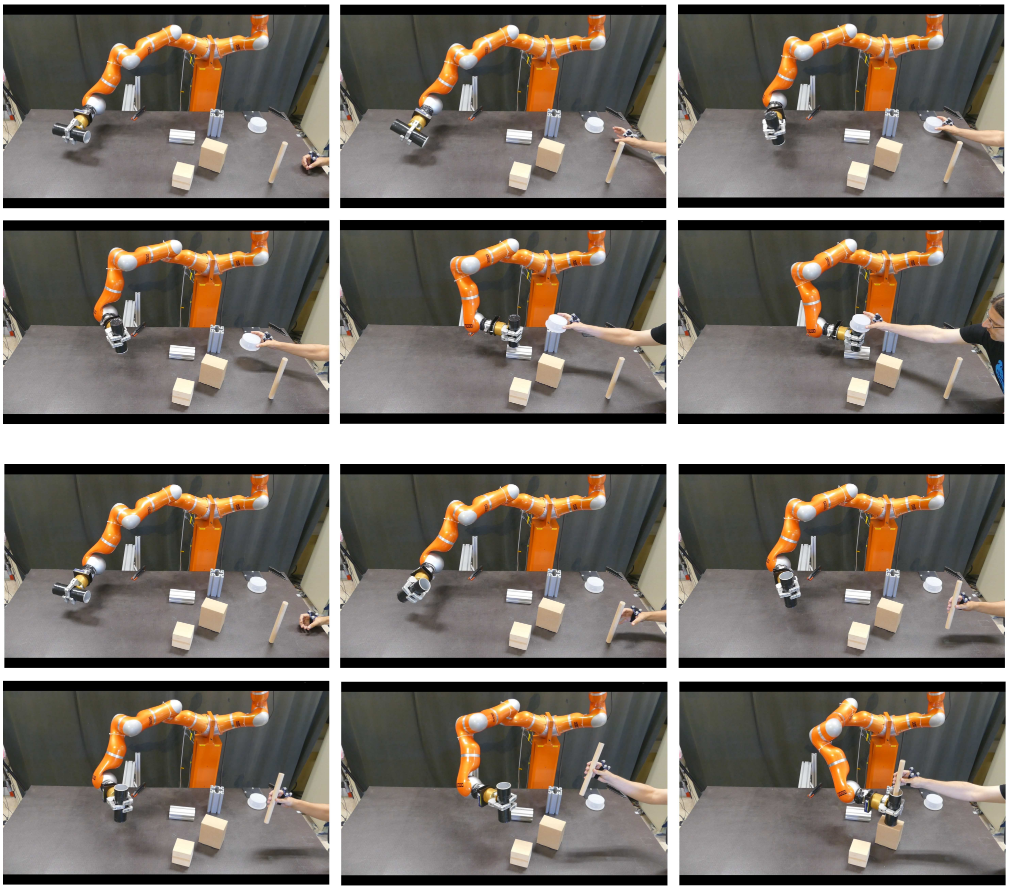


Figure 5: Images from two example executions of on-line movement recognition and cooperative task execution. The human reaching for the cover and moving it to one of the final positions, is shown in top 6 images. The robot immediately starts the suitable rotation for the cover and then moves to the appropriate end point. Bottom six images show another example execution. Here the human reaches for the peg and moves it to one of its final positions. Again, the robot executes an appropriate movement.

IEEE, 2015.

[8] M. Hersch, F. Guenter, S. Calinon, and A. Billard. Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Trans. Robot.*, 24(6):1463–1467, 2008.

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

[10] D. Lee, C. Ott, and Y. Nakamura. Mimetic communication model with compliant physical contact in human–humanoid interaction. *The International Journal of Robotics Research*, 29(13):1684–1704, 2010.

[11] G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters, and G. Neumann. Learning interaction for collaborative tasks with probabilistic movement primitives. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 527–534. IEEE, 2014.

[12] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Und.*, 104(2):90–126, 2006.

[13] L. Rozo, S. Calinon, D. Caldwell, P. Jiménez Schlegl, and C. Torras. Learning collaborative impedance-based robot behaviors. In *In Proc. AAAI Conference on Artificial Intelligence*, pages 1422–1428, Bellevue, WA, USA, 2013.

[14] S. Schaal. Is imitation learning the route to humanoid robots? *Trends Cogn. Sci.*, 3(6):233–242, 1999.

[15] S. Schaal, P. Mohajerian, and A. Ijspeert. Dynamics systems vs. optimal control–a unifying view. *Prog. Brain Res.*, 165:425–445, 2007.

[16] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, 2013.

[17] A. Ude, C. G. Atkeson, and M. Riley. Programming full-body movements for humanoid robots by observation. *Robot. Auton. Syst.*, 47(2):93–108, 2004.

[18] K. Yamane, M. Revfi, and T. Asfour. Synthesizing object receiving motions of humanoid robots with human motion database. In *In Proc. IEEE Int Conf on Robotics and Automation (ICRA)*, pages 1629–1636, Karlsruhe, Germany, 2013.

[19] Y. Yamane, K.and Yamaguchi and Y. Nakamura. Human motion database with a binary tree and node transition graphs. *Autonomous Robots*, 30(1), 2011.

Active Reconfiguration of Software and Hardware in a Robotic Workcell

Robert Bevec
Department of Automatics, Biocybernetics, and Robotics
Jožef Stefan Institute, Ljubljana, Slovenia
robert.bevec@ijs.si

Timotej Gašpar
Department of Automatics, Biocybernetics, and Robotics
Jožef Stefan Institute, Ljubljana, Slovenia
timotej.gaspar@ijs.si

Barry Ridge
Department of Automatics, Biocybernetics, and Robotics
Jožef Stefan Institute, Ljubljana, Slovenia
barry.ridge@ijs.si

Martin Bem
Department of Automatics, Biocybernetics, and Robotics
Jožef Stefan Institute, Ljubljana, Slovenia
martin.bem@ijs.si

Igor Kovač
Department of Automatics, Biocybernetics, and Robotics
Jožef Stefan Institute, Ljubljana, Slovenia
igor.kovac@ijs.si

Aleš Ude
Department of Automatics, Biocybernetics, and Robotics
Jožef Stefan Institute, Ljubljana, Slovenia
ales.ude@ijs.si

ABSTRACT

High volume production has been a prerequisite in order to invest into automation of the manufacturing process for decades. The high cost of setup and the inflexibility of the solution meant low batch productions, often present in *small and medium-sized enterprises (SMEs)*, were dismissed for automation. In order to bring automation closer to SMEs a flexible solution is required that can accommodate more than one manufacturing process, allows rapid change between them and doesn't require expertise knowledge for set up. In this paper we present a novel robotic workcell that enables active reconfiguration of software and hardware components, facilitating set up and production of several manufacturing processes on a single robotic cell. The ROS based software has been designed to be robot independent and modular. Special user interfaces have been developed for cell calibration, programming by demonstration and set up of quality control and part localization tasks. The proposed workcell is applicable in companies with product families, where manufacturing processes are similar and where fast changeover is required in order to adapt to new production requirements quickly. Due to the emphasis put on ease of use it will also be of interest to companies getting into automation for the first time.

1. INTRODUCTION

Fast changes in market demands lead to great fluctuation of orders down the manufacturing chain. Companies must react quickly, efficiently, and in an economically viable way. Robots have been successful in industrial production processes, when applied to repetitive tasks with long production runs and high unit volume. However, frequent shifts in the required product type or in the number of required products has prevented many companies from automating their manufacturing processes.

These so-called *few-of-a-kind* assembly production scenarios [7] are typical of SMEs. Since SMEs are the "backbone of the manufacturing industry", e. g. providing some ∼45% of the value added by manufacturing in the European Union [8],

Figure 1: The proposed reconfigurable workcell prototype shown at Hanover fair 2017.

it would be highly beneficial if robotic workcells could be developed specifically to address such use-cases.

The main barrier to greater adoption of robot production in SMEs are the expertise needed in setting up existing solutions and the time for testing and fine-tuning. Since SMEs usually do not have such expertise available, they avoid introducing such solutions, even when they are economically justifiable. We can recognize that these problems are due to the time costs involved in re-configuring and re-programming the robot workcell for new assembly tasks, which are often too prohibitive to make the application of robots profitable.

In this paper, we present the design of a new kind of autonomous robot workcell that is attractive not only for large production lines, but also for few-of-a-kind production. We propose reducing set-up times by exploiting a number of hardware and software technologies, some of which were partially developed in prior work, and some of which are novel contributions in this paper particular to the proposed workcell design.

The main novelty of the workcell lies in the active reconfiguration of passive fixtures and other passive elements in the cell, which can be performed by the robots installed therein. This reconfiguration process allows the robots to autonomously configure their workspace and prepare the workcell for the execution of new assembly tasks.

1.1 Related Work

Many surveys in recent years have followed the development of reconfigurable robotic systems, both in research and in industry [4]. Work of Chen [2] focuses on the modular reconfigurability aspect in particular by finding optimal module assembly configurations from a given set of module components for a specific task. His subsequent work on the design of a reconfigurable robotic workcell for rapid response manufacturing [3] is of particular relevance with respect to the workcell proposed in this paper. However, his work involved a workcell containing hardware elements that can only be reconfigured *manually*. Our proposed workcell focuses on introducing hardware elements that can be actively reconfigured *automatically* by the system itself [5].

In the work of Krüger *et al.* [7,9], a set of methods was developed to facilitate the set-up of complex automated assembly processes. The proposed set of methods included pose estimation and tracking of parts using a 3-D vision system, fast and robust robot trajectory adaptation using *dynamic movement primitives (DMP)* [10], and ROS-based software control and state machine programming. In this work we build on these approaches and add the ability to automatically reconfigure the workcell, while adding a user interfaces to facilitate task set up. The proposed system advances beyond synthetic benchmarks and demonstrates the viability of the system on actual industrial use-case.

2. RECONFIGURABLE SOFTWARE

The introduction of a robotic cell into a production line represents a big investment for SMEs. The high costs usually come from the price of the necessary hardware and the time spent for the integration of the robotic system into the production line. One of the time-consuming aspects is the programming of task sequences for the robots involved in the production process. The programming is normally done via on-line programming using a robot teach pendant connected to the robot controller, or via off-line programming in a simulation environment. Because specific robot system knowledge is required for both approaches, we developed a software system that would facilitate the programming of robot tasks regardless of the robot system. The software system is designed to be distributed, modular and offers seamless reconfiguration of the robot cell. The package also provides the necessary tools to enable simple, intuitive programming of robot tasks.

Our system was build using *Robot Operating System (ROS)* framework, where the hard real-time components were developed using a *Matlab Simulink Real-Time (SLRT)* server, since ROS in its current form does not provide any form of hard real-time implementation. This is a crucial requirement for reliable and accurate robot control.

2.1 System Architecture Overview

Elements of the software architecture of a typical workcell design can be seen in Figure 2. The "Robot Module"s, represent a robot with its robot control SLRT server, one additional measurement unit, and all of its tools and grippers. The "ROS Master Computer" refers to the computer in the system that runs the ROS core and our basic nodes. In order to access various periphery from the workcell a "Digital Interface Unit" is used for bridging the connection from PLC to ROS. Other typical modules include a "Vision Module" and a "Simulation Computer". The Vision Module represents a user programmable processing unit for typical vision tasks like quality control and part localization. The Simulation Computer offers a dynamic simulation environment provided by the VEROSIM software package, where the production can be planned and evaluated. Depending on requirements, the workcell design can be adapted by adding or removing various modules.

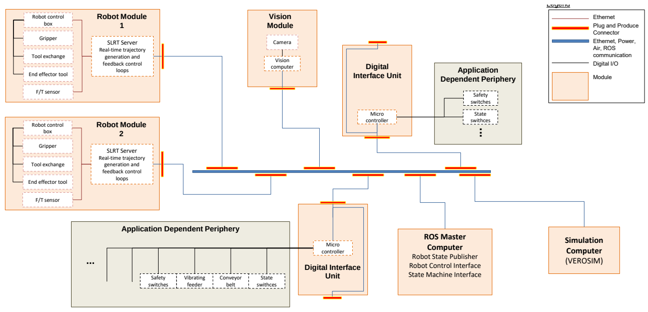


Figure 2: Schematics of the workcell software and hardware architecture.

2.2 Simulink Real-Time Server

The SLRT server, responsible for robot control, connects directly to the robot controller via Ethernet. Standard robot controllers usually only offer basic control methods and not state-of-the-art trajectory generation methods implemented on the SLRT server. Importantly this approach also makes our system independent of the robot. Compatible robots must enable receiving a joint stream over Ethernet and all it takes is a modification of the kinematics model on the SLRT server in order to integrate a new robot. The SLRT server also connects to some other measurement units (e.g. force/torque sensors) that can be used for closed loop control policies (e.g. force control).

The trajectory generation algorithms that have been implemented so far cover the most common robot motion needs in the context of automated assembly. These are: trapezoidal speed profile in joint space; minimum jerk for position and minimum jerk SLERP for orientation trajectories in Cartesian space; admittance force control [1]; joint space dynamic movement primitives for free-form movements [6]; Cartesian space dynamic movement primitives free-form movements in Cartesian space [10].

2.3 ROS Software Package

To allow the robot workcell to be accessed, controlled and calibrated within the ROS environment, various ROS nodes have been developed to offer an interface to the SLRT server and other modules in the workcell.

SLRT State Publisher to read joint positions, velocities, payload, tool information, force/torque sensor data from the SLRT server and publish it via ROS topics using standard ROS messages.

Action Servers handle communication to the SLRT server and are used to trigger robot motion and monitor the progress of the trajectory using *actionlib*. Each trajectory generation algorithm that is implemented on the SLRT server is offered as a separate action server with its own goal, feedback and result messages. The low level logic of all of the action servers ensures that only one motion can be executed at a time.

ROS Services for handling short duration tasks such as changing the state of digital outputs. Our ROS package includes services for changing the robot mode from position control to gravity compensation mode, triggering direct joint control on the SLRT server and setting digital outputs on the robot controller.

Database for keeping track of the workcell state at all times, be it during operation or downtime. We followed a common approach with wide support in the community and implemented a MongoDB database for persistent storage.

Robot Capture Program for capturing and storing various robot related configurations in the database. It is commonly used in conjunction with the kinesthetic guiding of the robot, where the programmer of the robot workcell can freely move the robot in its workspace and then save the points of interest. The functionality is commonly used for calibrating the workcell state (reconfigurable fixture positions, tool pick-up slots) and for saving pick-and-place poses of the robot assembly task.

Programming by Demonstration as an intuitive method for teaching the robot how to move to either points in Cartesian or joint space or over whole trajectories. This method has increased in popularity in recent years and more robot manufacturers are starting to implement the functionality on their robots.

Adaptation of Learned Trajectories by using admittance control. The displacement due to the force error is used as a correcting offset to our DMP encoded trajectory, when the learned trajectory is not ideal or optimal [1].

3. RECONFIGURABLE HARDWARE

The proposed robotic workcell is in large part constructed of modular hardware that allows for fast and easy reconfiguration; from the structural frame to the fixtures, end-effectors, tool exchange system, P&P connectors, and other peripheral devices. With this approach we make it possible to use the proposed workcell in a wide range of industrial applications and environments. Furthermore, we also make it relatively easy to alter its shape and purpose within those environments. The workcell follows the notion that automatic reconfiguration is achieved using robots, which actively reconfigure the passive elements inside the workcell. This concept drives down the cost of the cell making it more affordable for SMEs. The following technologies and solutions were used to achieve said hardware reconfigurability.

Reconfigurable Frame made of rectangular steel beams that are connected via the *BoxJoint* patented modular frame coupling technology. The advantage of this technology is that a workcell frame can be easily configured into a large variety of shapes.

Tool Exchange System for the robots to equip themselves with different grippers needed for different assembly tasks. Tools are introduced into the workcell on trolleys that connect to the P&P connector. If reconfiguration of the cell is needed to assemble a different workpiece, a new trolley with different end-effectors can be introduced into the workcell.

Passive Sensor-less Reconfigurable Fixtures designed in a Stewart platform-esque configuration with six legs, named "hexapod". These fixtures can be actively reconfigured by the robot arms on demand by connecting a robot to a fixture via the tool exchange system, releasing the fixture brakes, manipulating the fixture into the desired pose, re-applying the brakes, and disconnecting the robot from the fixture.

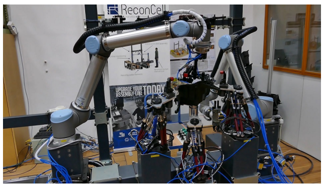
Passive Sensor-less Linear Unit, on which the robot is mounted, to enlarge the work area of the robot within the work cell. The robot is used to propel itself along the linear axis by connecting the end-effector to the frame and moving the base to a new position. Compared to conventional actuated solutions that are much more expensive, this approach is appropriate for applications where the need to move the robot is relatively infrequent.

4. USE CASE EVALUATION

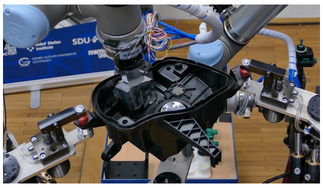
In order to evaluate our proposed robotic cell, we implemented a real industrial use-case. The industrial partner is involved in automotive light production, where the demand for different lights can vary substantially in a single year. The total production of each light housing is typically between 100,000-300,000 units per item. However, these lights are not assembled in one batch. Following the just-in-time production paradigm, a switch from production of one automotive light type to another is often necessary. The company uses product specific assembly devices, which are stored in a warehouse, when not in use. The devices must be stored to produce spare parts for at least the next five years. Production of spare parts in particular is a low quantity production scenario and usually occurs only a few times per year. Each changeover of production lasts several hours and presents a significant cost. It would therefore be extremely useful for suppliers to have a single robot cell available which is capable of assembling many different types of lights, while also being rapidly reconfigurable for alternating production scenarios.

The assembly device are currently operated by people, who insert parts in the machine and check the quality after assembly. Manual work and quality is highly dependent on workers' qualifications, skills and their knowledge of the assembly process. Customers expect that the supplier company is very flexible in coping with changes in demand. This is why SMEs seek to time every task carefully and look for optimizations. A fully automated assembly procedure also implements quality control checks and integrates in the company's business intelligence infrastructure providing key performance indicators (KPIs). Defective products can be detected before delivery and the KPIs can be used for analysis and prevention of defects and production optimization.

(a) Active reconfiguration of passive fixtures.



(b) Housing insertion and part pickup.



(c) Part assembly.

(d) Finished product removal

Figure 3: Key production steps for assembly of an automotive light using active reconfiguration.

The technologies described in this paper have been implemented for light assembly in the following way (*c.f.* Fig. 3). Before the start of the production of a new light housing model, the reconfigurable fixtures are actively placed in the appropriate configuration by robots, to accommodate the housing (main body of the headlamp) that comes directly from an injection molding machine. This step happens only once per production of a single type of headlamp (*c.f.* Fig. 3a). In the next step, the robots equip themselves with grippers with which it will pick up assembly parts. The assembly parts are detected using a calibrated visual localization system. The robot places the light housing into the fixture and inserts the remaining parts into the housing one by one (*c.f.* Fig. 3b,3c). After the assembly the robot grasps a camera to inspect the quality of the assembly. Finally, the other robot removes the housing from the fixtures, moving it on to the next step of the production process.

5. CONCLUSION AND FUTURE WORK

In this paper we presented a reconfigurable robotic workcell that targets the manufacturing industry with small production batches where changes in demand happen rapidly. The developed workcell consist of both modular software components and reconfigurable hardware elements. Affordable passive elements are actively reconfigured via robot manipulation to accommodate a different manufacturing process. To demonstrate the benefit of using such a workcell in an actual industrial scenario, a case study has been implemented with a partner from the automotive industry. In our experiments we demonstrated that the developed reconfigurable robot workcell provides the much needed flexibility and fast changeover characteristics for automated assembly processes in the context of automotive lights product family. In the future, we will focus on methods for finding a workcell configuration of reconfigurable components for assembly of a new product automatically, taking into account the constraints of the assembly procedure and the current workcell components.

6. ACKNOWLEDGMENTS

This work has been funded by the Horizon 2020 ICT-FoF Innovation Action no 680431, ReconCell (A Reconfigurable robot workCell for fast set-up of automated assembly processes in SMEs) and by the GOSTOP programme, contract no C3330-16-529000, co-financed by Slovenia and the EU under the ERDF.

7. ADDITIONAL AUTHORS

Additional authors: Žiga Gosar (Elvez d.o.o., Višnja Gora, Slovenia, email: `ziga.gosar@elvez.si`).

8. REFERENCES

[1] F. J. Abu-Dakka, B. Nemec, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, and A. Ude. Adaptation of manipulation skills in physical contact with the environment to reference force profiles. *Autonomous Robots*, 39(2):199–217, 2015.

[2] I.-M. Chen. *Theory and applications of modular reconfigurable robotic systems.* PhD thesis, California Institute of Technology, 1994.

[3] I.-M. Chen. A Rapidly Reconfigurable Robotics Workcell and Its Applications for Tissue Engineering. `https://dspace.mit.edu/handle/1721.1/3753`, 2003.

[4] M. Fulea, S. Popescu, E. Brad, B. Mocan, and M. Murar. A literature survey on reconfigurable industrial robotic work cells. *Applied Mechanics and Materials*, 762:233, 2015.

[5] T. Gašpar, B. Ridge, R. Bevec, M. Bem, I. Kovač, A. Ude, and Ž. Gosar. Rapid hardware and software reconfiguration in a robotic workcell. In *IEEE The 18th International Conference on Advanced Robotics (ICAR)*, pages 229–236, Hong Kong, China, 2017.

[6] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. *Advances in Neural Information Processing Systems*, pages 1523–1530, 2002.

[7] N. Krüger, A. Ude, H. G. Petersen, B. Nemec, L.-P. Ellekilde, T. R. Savarimuthu, J. A. Rytz, K. Fischer, A. G. Buch, D. Kraft, W. Mustafa, E. E. Aksoy, J. Papon, A. Kramberger, and F. Wörgötter. Technologies for the Fast Set-Up of Automated Assembly Processes. *KI - Künstliche Intelligenz*, 28(4):305–313, Nov. 2014.

[8] European Factories of the Future Research Association. Factories of the Future: Multi-Annual Roadmap for the Contractual PPP under Horizon 2020. Technical report, Publications office of the European Union: Brussels, Belgium, 2013.

[9] T. R. Savarimuthu, A. G. Buch, C. Schlette, N. Wantia, J. Rossmann, D. Martínez, G. Alenyá, C. Torras, A. Ude, B. Nemec, A. Kramberger, F. Wörgötter, E. E. Aksoy, J. Papon, S. Haller, J. Piater, and N. Krüger. Teaching a Robot the Semantics of Assembly Tasks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017. doi: 10.1109/TSMC.2016.2635479.

[10] A. Ude, B. Nemec, T. Petrič, and J. Morimoto. Orientation in cartesian space dynamic movement primitives. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2997–3004, Hong Kong, 2014.

SMACHA: An API for Rapid State Machine Assembly

Barry Ridge
Humanoid and Cognitive Robotics Laboratory
Department of Automatics, Biocybernetics and Robotics
Jožef Stefan Institute, Ljubljana, Slovenia
barry.ridge@ijs.si

ABSTRACT

Given the burgeoning complexity and diversity of both the hardware and software components of robotic systems, software libraries that use *state machines* as a basis for robot control by seamlessly connecting between low-level imperative task scripting and higher-level task planning have been in active development over the past decade or so. However, while they provide much in terms of power and flexibility, their overall task-level simplicity can often be obfuscated at the script-level by boilerplate code, intricate structure and lack of code reuse between state machine prototypes. To address these issues, we propose a code generation, templating and meta-scripting methodology for state machine assembly, as well as an accompanying application programming interface (API) for the rapid, modular development of robot control programs. The API has been developed within the ROS ecosystem to function effectively as either a front-end for concise scripting or a back-end for code generation for visual programming systems. Its capabilities are demonstrated in experiments using the Baxter robot simulator.

1. INTRODUCTION

The Robot Operating System (ROS) has, in recent years, become a popular choice of middleware for communication and control when designing robotic applications, and various packages within its ecosystem have come to the fore as being especially useful for dictating control flow. The SMACH high-level executive [3], in particular, has proven to be an exceptionally versatile and robust task-level architecture for state machine construction in ROS-based systems. It allows for the description of nested hierarchical state machines in Python in which parent container states contain child state sequences. State machines may describe lists of different possible outcomes and transitions are specified between states that depend on the outcomes in order to specify the control flow. These transitions are easily remapped across different depth levels in the hierarchy. Data may be passed between states as defined by a *userdata* object and the inputs and outputs of states may be remapped to userdata variables in order to control the flow of data.

While the ideas encapsulated by SMACH are conceptually simple, its usage still demands a significant degree of domain-specific expertise and prototyping time in order to define a functional state machine for a given robot control application. Another library that builds on the functionality of SMACH named FlexBE [5] aims at addressing this by providing a visual programming interface from which code may be generated. However, the generated code is language-specific and would therefore be brittle with respect to any significant changes to the programmatic approach.

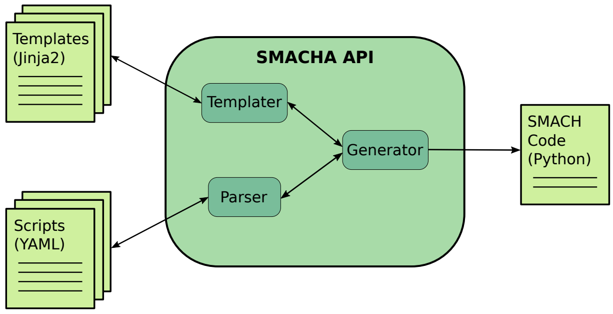


Figure 1: SMACHA API overview.

Here we present an application programming interface (API) named SMACHA[1] that aims at distilling the task-level simplicity of SMACH into compact scripts in the foreground, while retaining all of its power and flexibility in code templates and a custom code generation engine in the background. One of the major potential advantages of SMACHA is that it is is designed to be both language and framework agnostic. Although this has not yet been implemented, it would be possible, for example, to design templates to generate FlexBE code instead of SMACH code, or even state machine code written in a language other than Python, while maintaining the same scripting front-end.

2. SMACHA API OVERVIEW

The SMACHA API is composed of three main components as depicted in Fig. 1: a *parser*, a *templater* and a *generator*. The parser parses simple data-oriented scripts that describe the high-level arrangement of state machines to be constructed into operational program code by the generator and templater. We refer to this concept as *meta-scripting* and it is described below in Section 2.1. The templater retrieves and renders code templates as required by the generator in order to produce the code, and is described in Section 2.2. The generator recursively processes the parsed script and generates the final program code using the templater. It is described in Section 2.3. The relationship between the scripting and templating functionality, as well as the overall recursive code generation process, is depicted in Fig. 2.

[1] https://github.com/ReconCell/smacha

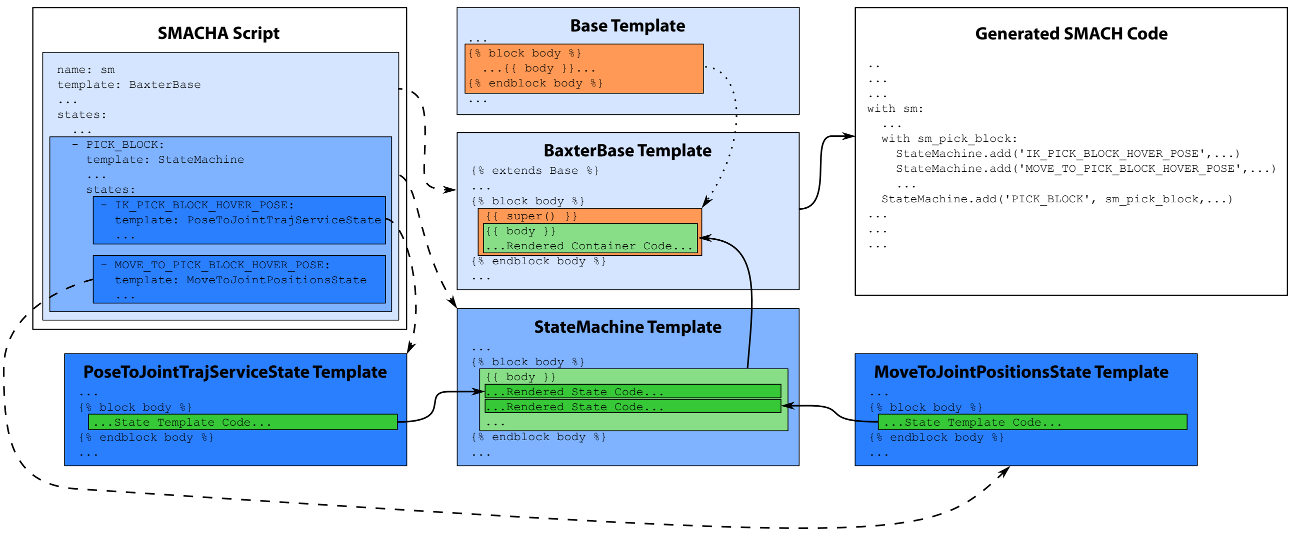


Figure 2: SMACHA recursive meta-scripting, templating and code generation pipeline example. Dashed arrows show nested state template selection from the SMACHA script and the blue shaded boxes indicate the depth level in the state hierarchy. Solid arrows and green shaded boxes show recursive template rendering flow, from child state templates at bottom-left and bottom-right, to a parent container *StateMachine* template at bottom-centre, to its parent *BaxterBase* template in the middle, to the final generated SMACH code on the right. Template inheritance is indicated by the dotted arrow and orange boxes.

2.1 Meta-Scripting

One of the core ideas behind the development of SMACHA is that state machines are essentially simple entities that can be almost entirely described via natural language constructs, perhaps augmented by some essential additional information necessary to describe how transitions should occur and how data should be passed between states. With this in mind, in order to transcribe the high-level logic of state machine description in as simple a manner as possible with a view towards offloading the more complex aspects to be processed by a code generation system working in the background we selected YAML (YAML Ain't Markup Language) as our scripting front-end [2]. YAML scripts are data-oriented and so are built around constructs such as lists and associative arrays that may be easily translated into corresponding machine code constructs and, more importantly for our purposes, can be used to represent both sequences of states and their individual data representations respectively. They can also represent data hierarchies very effectively, and are therefore well-suited to describing SMACH container states and nested state hierarchies. Thus, SMACHA scripts are YAML files that are used to describe how SMACHA should generate SMACH code. An example of a script that was written for a pick and place demonstration for the Baxter simulator can be see in Listing 1.

```
1 --- # Modular SMACHA pick and place test script for the Baxter simulator.
2 name: sm
3 template: BaxterBase
4 node_name: baxter_smach_pick_and_place_test
5 outcomes: [succeeded, aborted, preempted]
6 userdata:
7   hover_offset: [[0.0, 0.0, 0.15], [0.0, 0.0, 0.0, 1.0]]
8 states:
9   - LOAD_TABLE_MODEL:
10      template: LoadGazeboModelState
11      model_name: cafe_table
12      model_path: rospkg.RosPack().get_path('baxter_sim_examples') +
13                  '/models/cafe_table/model.sdf'
14      userdata:
15        table_model_pose_world: Pose(position=Point(x=1.0, y=0.0, z=0.0))
16        table_model_ref_frame: world
17      remapping: {pose: table_model_pose_world,
18                  reference_frame: table_model_ref_frame}
19      transitions: {succeeded: LOAD_BLOCK_MODEL}
20
21  - LOAD_BLOCK_MODEL:
22      template: LoadGazeboModelState
23      model_name: block
24      model_path: rospkg.RosPack().get_path('baxter_sim_examples') +
25                  '/models/block/model.urdf'
26      userdata:
27        block_model_pick_pose_world: [[0.6725, 0.1265, 0.7825],
28                                      [0.0, 0.0, 0.0, 1.0]]
29        block_model_pick_ref_frame: world
30        block_model_pick_pose: [[0.7, 0.15, -0.129],
31                                [-0.02496, 0.99965, 0.00738, 0.00486]]
32        block_model_place_pose: [[0.75, 0.0, -0.129],
33                                 [-0.02496, 0.99965, 0.00738, 0.00486]]
34      remapping: {pose: block_model_pick_pose_world,
35                  reference_frame: block_model_pick_ref_frame}
36      transitions: {succeeded: MOVE_TO_START_POSITION}
37
38  - MOVE_TO_START_POSITION:
39      template: MoveToJointPositionsState
40      limb: left
41      userdata: {joint_start_positions:
42          [-0.08000, -0.99998, -1.18997, 1.94002, 0.67000, 1.03001, -0.50000]}
43      remapping: {positions: joint_start_positions}
44      transitions: {succeeded: PICK_BLOCK}
45
46  - PICK_BLOCK:
47      script: pick_block
48      remapping: {pick_pose: block_model_pick_pose,
49                  hover_offset: hover_offset}
50      transitions: {succeeded: PLACE_BLOCK}
51
52  - PLACE_BLOCK:
53      script: place_block
54      remapping: {place_pose: block_model_place_pose,
55                  hover_offset: hover_offset}
56      transitions: {succeeded: succeeded}
```

Listing 1: SMACHA pick and place demo script.

2.1.1 Base Variables

The base of a main SMACHA script file specifies the following variables: *name* (a name for the overall state machine), *template* (the name of its base template), *manifest* (an optional ROS manifest name), *node_name* (a name for its associated ROS node), *outcomes* (a list of its possible outcomes) and *states* (a list of its constituent states). Each of the states in the base script may, in turn, specify similar variables of their own, as discussed in the following.

2.1.2 States

Each state, including the base, must specify a template from which its respective code should be generated (see *e.g.* lines 3, 10, 22 and 39 of Listing 1). States may be specified as lists specifying their transition order (see *e.g.* lines 8, 9, 21, 38, 46 and 52 of Listing 1), and may also be nested as described in the SMACH documentation using appropriate combinations of template and state specifications. Possible state outcomes may be specified as a list in the base state machine and in each container state (see *e.g.* line 5 of Listing 1). Possible state transitions may be specified as an associative array in each state (see *e.g.* lines 19, 36, 44, 50 and 56 of Listing 1). Input and output remappings of user data may be specified as an associative array in each state (see *e.g.* lines 17, 34, 43, 48 and 54 of Listing 1).

2.1.3 Modularity

Modularity is achieved at the scripting level by allowing useful subroutines wrapped in container states to be saved as separate YAML script files called *sub-scripts* which can be included in a main script as states. Examples of this can be seen in lines 46–50 and 52–56 of Listing 1, where the sub-scripts "pick_block" and "place_block" are included in the main pick and place state machine script to define its sub-states. The input and output userdata keys expected by the container states in the sub-scripts may be remapped as appropriate in the main script along with their state transitions. The use of this functionality encourages low coupling and high cohesion, while allowing for extremely rapid and easily specified reuse of common patterns.

2.2 Templating

Code templating is implemented using the Jinja2 templating library [1]. Core templates are provided by default to support standard SMACH states and custom templates may be defined for particular use cases.

2.2.1 Core Templates

SMACHA provides default core templates for many of the SMACH states and containers, as well as for other useful constructs. At the time of writing, the following core templates are present and functional: *Base* (Python script skeleton), *State* (contains functionality common to all states, *e.g.* userdata specification), *StateMachine* (container), *Concurrence* (container), *ServiceState* (generic state), *SimpleActionState* (generic state), *ReadTopicState* (custom state used for reading messages from ROS topics) and *TF2ListenerState* (custom state used for reading TF2 transforms).

2.2.2 Code Generation Variables and Code Blocks

There are a number of core code generation variables and code blocks present in the core templates that enable SMACHA to produce code in the appropriate places. In most cases, a code block contains a variable of the same name within it to indicate where code from child state templates should be rendered into. The main code blocks are as follows: *base_header* (for code that must appear near the top of the program script), *defs* (for function definitions), *class_defs* (for class definitions), *main_def* (for the main function definition), *header* (for code that is to be rendered into the *header* variable the parent template), *body* (for code that is to be rendered into the *body* variable of the parent template), *footer* (for code that is to be rendered into the *footer* variable of the parent template), *execute* (for the code necessary for executing the state machine), *base_footer* (for any code that must appear near the bottom of the program script) and *main* (for the code necessary to execute the main function). The most important block for most state templates is the *body* block and its associated *body* variable, as it is where the state template should render the code necessary to add the state to the parent state machine, which will either be some container state or the base state machine itself. Note that all of the above code generation variables and code blocks may be either removed, modified or arbitrarily customized within the API for particular use cases. The code insertion order may also be specified within the API, i.e. code may be either prepended or appended to a variable. An example of how code generation variables work together with code blocks is depicted in Fig. 2.

2.2.3 Template Inheritance

Jinja2 provides powerful functionality, including the ability to extend templates via template inheritance, such that their constituent code blocks may be overridden or extended. SMACHA aims to incorporate as much of this functionality as possible, thus the core templates may be overridden or augmented by custom user-specified templates via the usual Jinja2 template inheritance mechanisms, with some caveats. This works in the usual way using the following Jinja2 variables and expressions: `{% extends "<template_name>" %}` (to inherit code blocks from the parent template specified by `<template_name>`), `{{ super() }}` (when this appears in a block, the code from the same block in the parent template as specified by `{{ extends }}` will be rendered at its position) and `{% include "<template_name>" %}` (to include all code from the template specified by `<template_name>`).

Regarding the aforementioned caveats, there is a behaviour that is specific to SMACHA that goes beyond the usual capabilities of Jinja2 and that was designed as a means of dealing with the recursive state machine processing required by this particular use case. If a state template contains blocks, but does not contain an `{{ extends }}` expression at the top of a template, it is implied that the code for the blocks will be rendered into variables and blocks with the same names as the blocks in the state template as dictated by the SMACHA script and as defined usually either by the base template or container templates. In the current implementation, only base templates use the `{% extends %}` inheritance mechanism, whereas state and container templates use the `{% include %}` mechanism to inherit code from other templates. This is partially illustrated in Fig. 2.

2.3 Code Generation

The SMACHA code generator is a custom-designed engine for recursively generating state machine code based on the scripts described in Section 2.1 and using the templates described in Section 2.2. Recursive processing was necessary given the potentially arbitrary depth levels of state machine nesting that are possible under the SMACH API. The basic operation scheme behind the code generator is thus to iterate through the data constructs of a parsed script, evaluate them based on their type, and determine whether they should be rendered as code using the appropriate templates, passed on for recursive processing, or some combination of both. When iteratively processing a script, data items that are encountered are either lists or associative arrays. When a list is encountered, it is assumed that it is a list of states and is passed on for further recursive processing. When processing an associative array, there are three main cases that

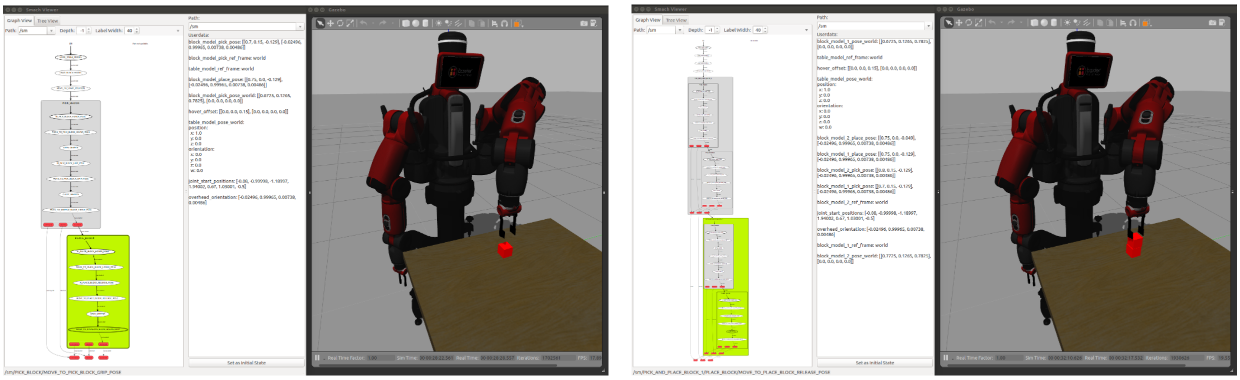


Figure 3: Pick and place (left) and stacking (right) tasks running on the Baxter simulator using SMACHA-generated code.

need to be handled separately: *container states*, *sub-script states* and *leaf states*. The recursive processing of container and leaf states is partially illustrated in Fig. 2.

3. EXPERIMENTS

For the experiments, we chose to use the Rethink Robotics Baxter robot [4] simulator which uses the Gazebo simulation system and comes equipped with extensive ROS support by default. Custom code templates were designed to facilitate the development of the necessary states required for the experiments. Two experiments were performed in total using these templates: a pick and place experiment and a block stacking experiment[2]. The first of these is a replication of the pick and place demo that comes as standard with the Baxter SDK. It was initially re-programmed from scratch in order to make use of SMACH and such that the control logic of the demo could be specified using a state machine. After that it was possible to design the necessary code templates and script the demo using SMACHA. Once the custom templates and the SMACHA script had been created for the first demo, it was possible to reuse both of them to very rapidly script the second experiment for block stacking. In both cases, it was possible to run the Python SMACH code generated by SMACHA without further modification with both experiments completing successfully.

The Baxter SMACHA package[3] includes the following custom code templates: *BaxterBase* (extends the core *Base* template), *LoadGazeboModelState* (allows allows a specified Gazebo model to be loaded into the simulator), *MoveToJointPositionsState* (moves a Baxter limb to a specified set of joint positions), *PoseToJointTrajServiceState* (uses inverse kinematics to calculate a set of joint positions from a specified end-point pose), and *GripperInterfaceState* (either opens or closes a specified gripper). In the inital states of the pick and place experiment state machine, as specified by the SMACHA script in Listing 1, a table model must be loaded into the simulator using the *LoadGazeboModelState* template, followed by a block model placed at a specified pose on the table, and the left limb of the robot must be moved to a starting position using the *MoveToJointPositionsState* template. Subsequently, the robot enters a "PICK_BLOCK" state as specified by the "pick_block" sub-script in order to pick the block from the table, followed by a "PLACE_BLOCK" state as specified by the "place_block" sub-script in order to place the block at a given placement pose. The stacking experiment initialises similarly to the pick and place experiment, only in this case, two block models are loaded instead of one, and the robot is tasked with stacking one on top of the other. This essentially involves two pick and place sequences, one for each block, so the "pick_block" and "place_block" sub-scripts used in the previous experiment are reused. The results of both experiments are depicted in Fig. 3.

[2]Video available at: https://youtu.be/WFp_keDsA6M
[3]https://github.com/abr-ijs/baxter_smacha

4. CONCLUSION

We have developed an API for the rapid assembly of state machines for modular robot control using a meta-scripting, code templating and code generation paradigm. It has been demonstrated on a simulated humanoid robot platform in two different experiments.

5. ACKNOWLEDGEMENTS

This work has been funded by the Horizon 2020 ICT-FoF Innovation Action no 680431, ReconCell (A Reconfigurable robot workCell for fast set-up of automated assembly processes in SMEs) and by the GOSTOP programme, contract no C3330-16-529000, co-financed by Slovenia and the EU under the ERDF.

6. REFERENCES

[1] Jinja2 (The Python Template Engine). `http://jinja.pocoo.org/`. (Last accessed 2017-07-24).
[2] YAML Ain't Markup Language (YAML™) Version 1.1. `http://yaml.org/spec/1.1/`. (Last accessed 2017-07-25).
[3] J. Bohren and S. Cousins. The SMACH High-Level Executive [ROS News]. *IEEE Robotics Automation Magazine*, 17(4):18–20, Dec. 2010.
[4] E. Guizzo and E. Ackerman. How rethink robotics built its new baxter robot worker. *IEEE spectrum*, page 18, 2012.
[5] P. Schillinger, S. Kohlbrecher, and O. von Stryk. Human-robot collaborative high-level control with application to rescue robotics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2796–2802, May 2016.

Extending the Workspace of Pseudo-Linear Variable-Lever Variable Stiffness Actuator

Miha Dežman
Institute Jožef Stefan
Jamova cesta 39
Ljubljana, Slovenia
miha.dezman@ijs.si

Andrej Gams
Institute Jožef Stefan
Jamova cesta 39
Ljubljana, Slovenia
andrej.gams@ijs.si

ABSTRACT

Among the different mechanically compliant actuators, variable stiffness actuators have the possibility to change their mechanical compliance on the fly, which is favorable in many force control applications. Though their control and mechanics are of higher complexity, their simplification is in the research focus of many groups. In this work we propose a modification by adding additional springs to our novel variable stiffness principle, which solves the rising stiffness torque drawback of the previous design. We performed a parameter search over two modification parameters to find the combination with the most suitable workspace. The proposed mechanical modification extends the torque/deflection workspace of the original principle, while keeping its favorable properties, i.e. pseudo-linear torque deflection characteristics and low power to change stiffness.

Keywords

mechanically compliant actuator, variable stiffness, optimization, parameter search

1. INTRODUCTION

Robots using mechanical compliance have several advantages when compared to classical stiff actuators [1], including safer human-robot interaction and higher energy efficiency. The compliant element is a low-pass filter [2] that reduces the peak gear forces, however, at the same time it also reduces the bandwidth of the actuator. Improved force accuracy in higher stability is another advantage.

Much of the research is focused in this field because of the favorable properties of such devices. Higher mechanical and control complexity offer many simplification possibilities in combination with a wide array of possible applications. Researchers focus on the novel device architectures development and prototype manufacturing due to commercial unavailability of such devices. We point the reader to some articles that provide an overview over different design architectures, for example, [3], [4] and [5].

There already exist many interesting applications that utilize the principle of mechanical compliance. One of them is a small jumping robot called Salto [6], that is capable of jumping to a 1m height, trying to mimic a jumping animal called Senegal bushbaby (Galago senegalensis). Another interesting example is the passive-ankle exoskeleton [7], which is able to reduce the users walking effort for 10% in a passive manner without using motors as power inputs by utilizing springs. We developed a similar exoskeleton, where we implemented a mechanism to close the clutch using a small motor [8], which increased its operation reliability.

This work is the development continuation of a novel variable stiffness actuator principle presented in [9]. A quasi-linear torque-deflection is the main advantage of the proposed configuration. Another advantage is the simplicity of mechanical configuration. However, a drawback of the device is the growing pressure angle at lower stiffness positions, which increases the torque required to change the stiffness and limits the devices maximum deflection.

We begin the article with a description of the original operation principle in section 2 and explain the modification in section 3. In section 4 we explain the parameter optimization and show its results in section 5. We conclude the work in section 6.

2. VSA OPERATION PRINCIPLE

The development of a variable stiffness actuator (VSA) encompasses a complex design process with many parameters [10]. The number of parameters is higher than with classical stiff actuators. Furthermore, mechanical convenience is also important to keep the design simple and easy to manufacture, and is hard to achieve due to so many parameters. The devices requirements also differ based on the desired application.

The goal of our design is to keep the structure simple, with a favorable torque deflection characteristic. The weight of the device and its power requirements should also be minimal.

The core of our device is a combination of a cam mechanism and a variable lever principle (see Fig. 1). A curved lever arm rotates (φ_d) around the pivot A and compresses the spring follower. This provides a reaction torque to the external load. Via the follower rotation (φ_s) around pivot B, the effective radius of the spring to the pivot A changes, thus changing the stiffness of the mechanism. The system is in principle unidirectional, since the spring can be compressed only. To achieve a bidirectional application, a system of gears or cables can be used to change the bidirectional motion of the external link to unidirectional motion for spring compression. At zero deflection, the follower can be rotated with minimal torque. The reason is, that the device uses a pretension-less spring principle [10].

The analytical model of the torque/deflection characteristics was

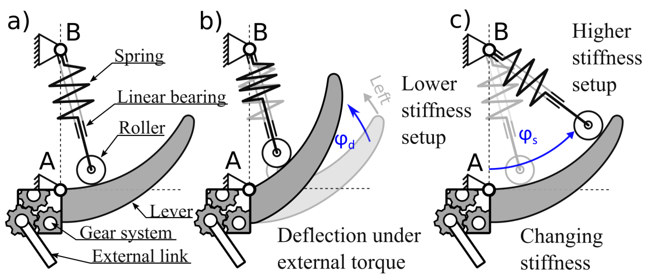


Figure 1: The actuator working principle: a) equilibrium position, b) deflection under external load, and c) varying stiffness.

derived in [9]. To test the principle, a rapid-prototype was also developed. In this article, we consider a case of a real prototype, with the theoretical data specified in Table 1.

Table 1: Calculation and prototype parameters

Variable	Name	Value	Unit
R_s	Cam curve radius	50	[mm]
k_{lin}	Linear spring stiffness	72.6	[N/mm]
φ_d	Deflection angle (min/max)	$\pm$ 40	[^{o}]
φ_s	Stiffness setup angle range	0 - 90	[^{o}]
T_{pn}	Pos. motor nominal torque	15	[Nm]
T_{pp}	Pos. motor peak torque	22.5	[Nm]
T_{sn}	Stiff. motor nominal torque	5	[Nm]
T_{sp}	Stiff. motor peak torque	6.3	[Nm]

Due to nominal and peak torque limits of the position motor and stiffness motor gearboxes, a torque/deflection graph for the selected mechanism with marked operation areas can be created. The blue area marks the workspace where both motors operate below the nominal torque, while the red area represents the torque levels between the nominal and peak torque limits.

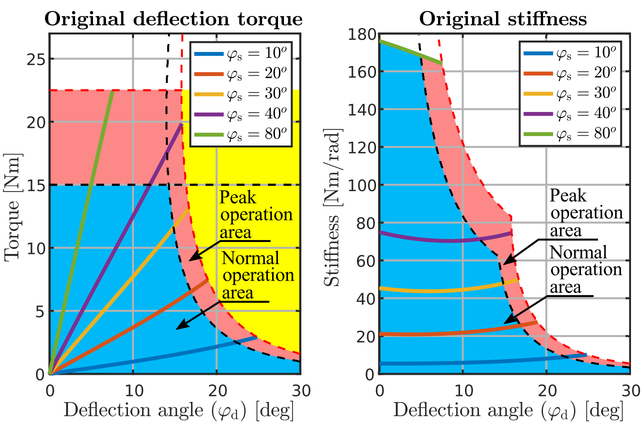


Figure 2: Normal torque/deflection workspace and the corresponding stiffness graph.

One drawback of the proposed system is the rising pressure angle at lower stiffness presets and higher deflections. Due to the curvature of the deflection lever, no torque is theoretically needed to change the stiffness at zero deflection. However, as the deflection increases, the stiffness torque rises due to an increasing pressure angle. This is one of the drawbacks of the current system architecture.

The stiffness torque required to change the stiffness is shown in Fig. 3 for different stiffness presets. One can see that the stiffness motor torque theoretically easily gets higher than 30 Nm, which is above the peak torques of the stiffness motor and the position motor. Though a stronger motor could be used, a mechanical solution would be far more convenient. Seeing Fig. 2, the torque limits separate the torque/deflection graph into three areas. The yellow workspace area is where the stiffness torque overpowers the stiffness motor. It is possible to stay outside of the yellow area by keeping the stiffness angle above $\varphi_s > 40^o$. Here the actuator operates reliably. However, by implementing two additional springs, the workspace can be extended.

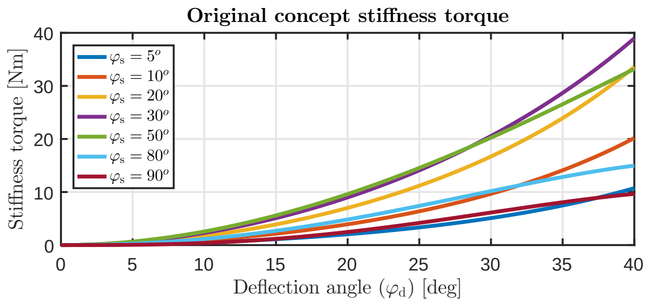


Figure 3: The torque required to change the device stiffness for the original case.

3. EXTENDING THE VSA WORKSPACE

In order to extend the torque/deflection workspace, we propose to add two more springs into the system. With this we are able to keep the torque required to change stiffness low while extending the devices workspace. The modification is shown in Fig. 4.

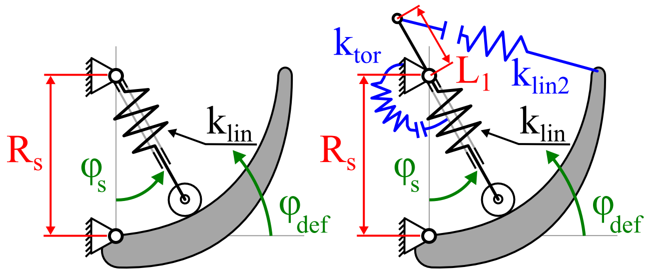


Figure 4: The main device parameters (left) with the two proposed modifications (right).

The first modification is the addition of a torsion spring k_{tor} (see Fig. 4). The stiffness torque at zero deflection is negligible and it rises with the deflection, whereas it is higher at lover stiffness positions and lower at high stiffness positions. We can thus increase the workspace by adding a torsion spring to create a negative stiffness torque offset. This way, the compensation stiffness torque (M_{cs}), due to spring k_{tor}, rises linearly from $\varphi_s = 90^o$ to $\varphi_s = 5^o$ as described in equation

$$M_{sc}(\varphi_s) = \frac{M_0}{90^o - 5^o} \cdot (90^o - \varphi_s)\,, \tag{1}$$

where the M_{cs} is the compensation torque, T_0 is the maximum torque and φ_s the stiffness angle. The maximum stiffness compensation torque is reached at $\varphi_s = 5^o$. The workspace below $\varphi_s = 5^o$ is deemed unusable. By calculating the new stiffness torque, an increase in the workspace can be observed as seen Fig. 5. The black area represents the workspace before and the (red and blue area) after the addition of the torsion spring.

Second modification is the addition of a linear spring (k_{lin2}), which is not as trivial as before. It is best, if the torque required to assist the stiffness motor comes directly from the external torque. Therefore, the second modification is a spring connected between the curved deflection arm and the follower. Note, that since the follower can be rotated, the spring system is compression based only. This way, at lower stiffness angles the assistance is the strongest, while at the higher stiffness presets the assistance is minimal or there is none. The combination of the L_1 and k_{lin2} parameter has an optimal solution. To find it, we need a new mathematical model.

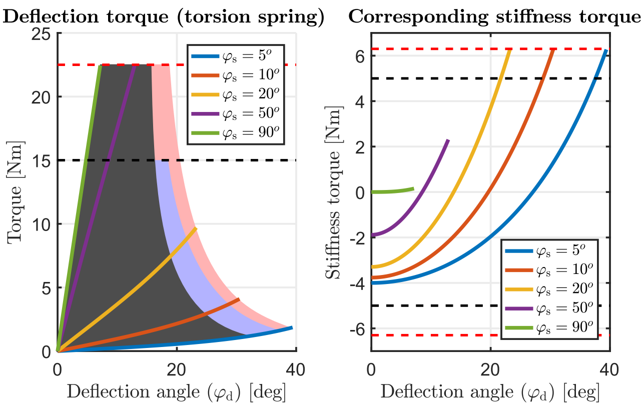


Figure 5: Normal deflection torque (black area) and the new workspace with torsion spring (red-blue area). The corresponding stiffness torque is on the right graph.

Considering the two additional springs, the new torque/deflection characteristics can be calculated as

$$M_{\text{dnew}} = M_{\text{d}} + M_{\text{dk2}}, \tag{2}$$

where the M_{d} presents the torque from the old system, derived in [9]. The torque needed to compress the new spring k_{lin2} is M_{dk2} and can be calculated as a vector product

$$M_{\text{dk2}} = \left|\vec{r}_{01} \times \vec{F}_{\text{k2}}\right|, \tag{3}$$

where the $\vec{F}_{\text{k2}}$ is the force from the k_{lin2} spring compression and $\vec{r}_{01}$ the lever vector, as can be seen from Fig. 6. The new stiffness torque is calculated as

$$M_{\text{snew}} = M_{\text{s}} - M_{\text{sc}} - M_{\text{sk2}}, \tag{4}$$

where the M_{s} is adapted from [9] and the M_{sc} from (1). The M_{sk2} represents the stiffness torque resulting from the spring compression and is calculated as the vector product

$$M_{\text{sk2}} = \left|\vec{r}_{34} \times \vec{F}_{\text{k2}}\right|. \tag{5}$$

Again, see Fig. 6. Note, that when the stiffness motor rotates (φ_s), the spring k_{lin2} is only active when the distance between points (x_4, y_4) and (x_1, y_1) is smaller than the no-load spring length.

4. PARAMETER OPTIMIZATION

In order to determine the optimal parameters r and k_{lin}, we performed a parameter search. Fig. 7 represents the workspace for normal PLVL-VSA configuration and the one with additional springs.

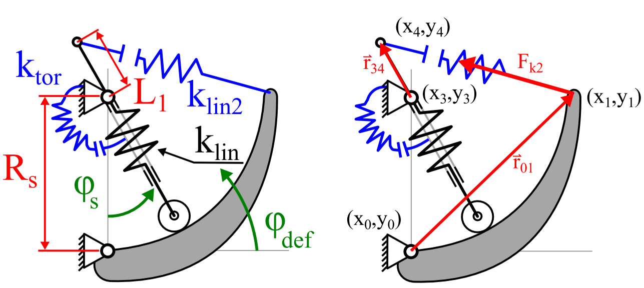


Figure 6: A scheme of the proposed mechanism with the force and distance vectors.

The gray (black) area represents the workspace, where the stiffness

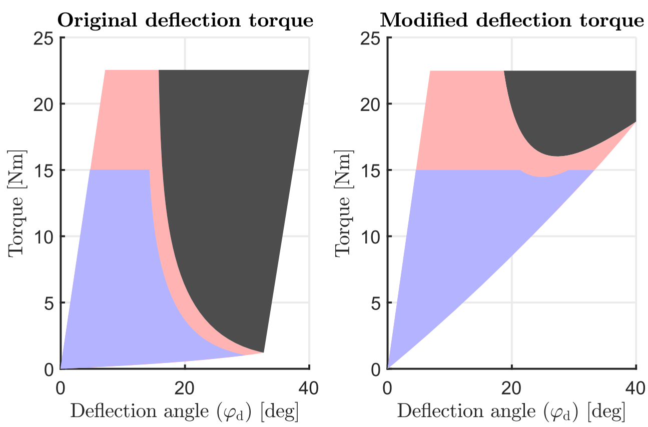


Figure 7: Workspace comparison between the normal and modified mechanism version, where red-blue area is desired and black undesired.

setup motor does not have enough torque to move. One can see, that by addition of the springs, the black area shrinks. In the blue area, the motor torque is below nominal and in the red, the motor torque is between nominal and peak operation limits. The size of the areas can be numerically calculated and compared for different parameter combinations. The parameters T_{pn}, T_{pp}, T_{sn} and T_{sp} represent the limits of the position motors and stiffness motors gearboxes. Their values can be found in Table 1. These parameters limit the torque/deflection workspace of our actuator.

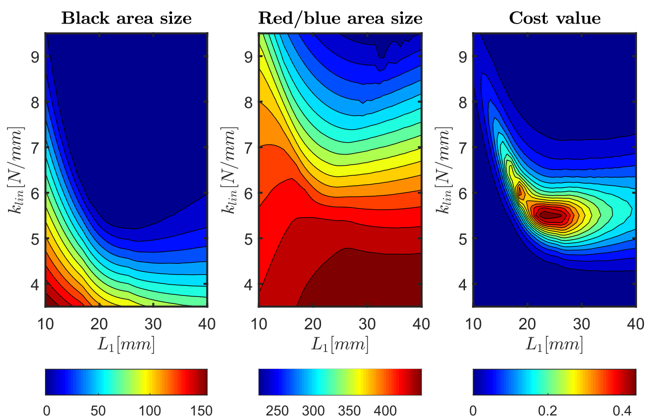


Figure 8: Black, red-blue area sizes and the corresponding cost value for different k_{lin} and L_1 parameter combinations.

Our goal is to find the optimal combination of parameters L_1 and k_{lin2}, where the black area of the actuator is of minimal size and the red-blue area is of maximal size. We cycle through both parameters in range $L_1 = 10 - 40$mm and $k_{lin2} = 1 - 20$N/mm. For each parameter set of L_1 and k_{lin2}, we use the theoretical model and calculate the torque/deflection graph with nominal/peak torque limits. We numerically measure the areas of the workspace areas and collect them into a matrix. We determine the optimal configuration by taking the normalized versions of areas using the following cost function

$$\delta = A_{redblue} * (1 - A_{black})^{10}, \tag{6}$$

where $A_{redblue}$ represents the area of the sum of red and blue workspaces and the A_{black} represents the black workspace. The 10th exponential makes the local maxima easier to see. The optimization graphs are shown in Fig. 8. Looking at the graph titled Cost value, one can spot a local maxima near parameter combination $L_1 = 25$ mm and $k_{\text{lin}} = 5.5$ N/mm. A rounded approximation close to the real optimal value is better suited for manufacturing. At the chosen parameters, the sum of red-blue area is at maximum while the black area is at minimum.

5. OPTIMIZED WORKSPACE

The optimal parameter torque/deflection graphs are shown in Fig. 9. One can see, that the system still preserves the quasi-linearity, which

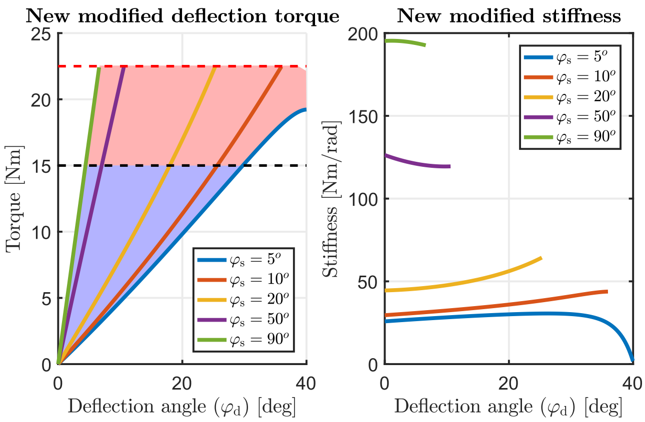


Figure 9: The torque/deflection workspace for the new mechanism with the corresponding stiffness.

is favorable from the control perspective. The deflection range is now extended to 40^o at lower stiffness presets. As can be observed, we can vary the stiffness between 30 to 200 Nm/rad. The stiffness torque, required to change the stiffness, is shown in Fig. 10. As predicted, due to the modifications, the new stiffness torque is kept between the peak torque limits. This shows that the proposed modifications increases the performance of the device while using the same motors as in the original concept. Since the motors contribute a lot of weight to the overall system, keeping the motors small also helps keeping the overall device light, small and compact.

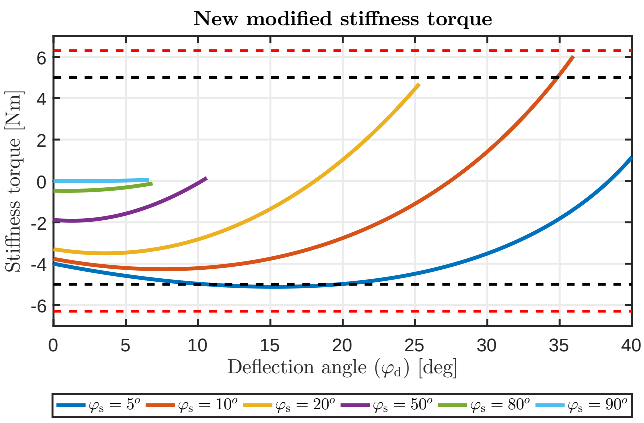


Figure 10: The stiffness torque of the modified mechanism.

6. CONCLUSIONS

We presented the continuing development of a novel variable stiffness actuator. The proposed modifications of the original operation principle extends the devices torque/deflection workspace, while keeping the same stiffness and position motors. The modification successfully minimizes the effects of the original principles main drawback, the rising stiffness torque due to the rising pressure angle. In the future, the proposed modifications will be implemented in a real-world prototype.

7. REFERENCES

[1] Ronald Van Ham, Thomas G Sugar, Bram Vanderborght, Kevin W Hollander, and Dirk Lefeber. Compliant actuator designs. *IEEE Robotics & Automation Magazine*, 16(3), 2009.

[2] Gill A Pratt and Matthew M Williamson. Series elastic actuators. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 1, pages 399–406. IEEE, 1995.

[3] Bram Vanderborght, Alin Albu-Schäffer, et al. Variable impedance actuators: A review. *Robotics and autonomous systems*, 61(12):1601–1614, 2013.

[4] Nevio Luigi Tagliamonte, Fabrizio Sergi, et al. Double actuation architectures for rendering variable impedance in compliant robots: A review. *Mechatronics*, 22(8):1187–1203, 2012.

[5] Ronald Van Ham, T Sugar, et al. Review of actuators with passive adjustable compliance/controllable stiffness for robotic apllications. *IEEE Robotics and Automation mag.*, 16(3):81–94, 2009.

[6] Duncan W Haldane, MM Plecnik, Justin K Yim, and Ronald S Fearing. Robotic vertical jumping agility via series-elastic power modulation. *Science Robotics*, 1(1):eaag2048, 2016.

[7] Steven H Collins et al. Reducing the energy cost of human walking using an unpowered exoskeleton. *Nature*, 522(7555):212–215, 2015.

[8] Miha Dežman, Jan Babič, and Andrej Gams. Qualitative assessment of a clutch-actuated ankle exoskeleton. In *International Conference on Robotics in Alpe-Adria Danube Region*, pages 778–786. Springer, 2017.

[9] Miha Dežman and Andrej Gams. Pseudo-linear variable lever variable stiffness actuator: Design and evaluation. In *Advanced Intelligent Mechatronics (AIM), 2017 IEEE International Conference on*, pages 785–790. IEEE, 2017.

[10] S. Wolf, G. Grioli, O. Eiberger, et al. Variable stiffness actuators: Review on design and components. *IEEE/ASME Transactions on Mechatronics*, 21(5):2418–2430, Oct 2016.

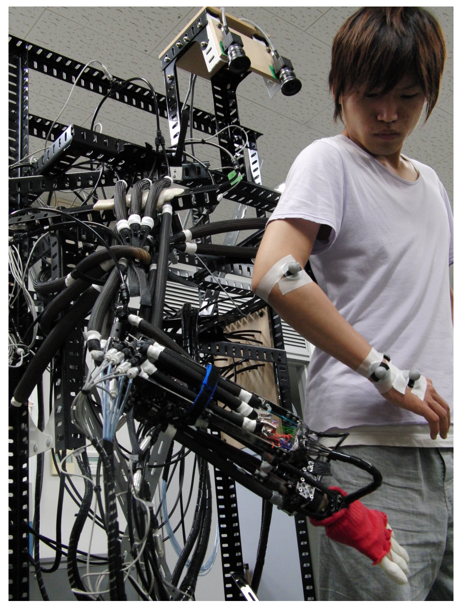
Soft Humanoid Robots

Koh Hosada
Department of System Innovation, Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531/ Japan
hosoda@sys.es.osaka-u.ac.jp

ABSTRACT

A human can exhibit intelligent behavior. Yet, we do not completely understand the mechanism how the behavior emerges. Adaptive behavior is obviously generated by the brain, but brain alone cannot explain everything. Key components are soft tissue, muscles, bones, and skin.

This talk will introduce our challenges to build soft humanoid robots consisting of muscles, bones, and skin so that we can constructively understand the human's adaptive intelligence.



Semantic Reasoning under Realistic Conditions

Karinne Ramirez-Amaro and Gordon Cheng
Institute for Cognitive Systems, Technical University of Munich, Germany
Arcisstrasse 21, 80333, Munchen
karinne.ramirez@tum.de and gordon@tum.de

ABSTRACT

One fundamental issue of autonomous robots in task domains is its capability to learn new skills and to re-use past experiences under different situations as efficient, intuitive and reliable as possible. A promising mechanism to achieve that is via learning from demonstrations or observations. In this abstract, we present a novel learning method that generates compact and general semantic models to infer human activities. We propose a method that allows robots to obtain and determine a higher-level understanding of a demonstrator's behavior via semantic representations [4]. First, the low-level information is extracted from the sensory data, then a meaningful semantic description, the high-level, is obtained by reasoning about the intended human behaviors [6]. The introduced method has been assessed on different robots, e.g. the iCub [3], REEM-C [5], and TOMM [2], with different kinematic chains and dynamics. Furthermore, the robots use different perceptual modalities, under different constraints and in several scenarios ranging from making a sandwich to driving a car assessed on different domains (home-service and industrial scenarios). Each of the studied scenarios poses distinct and challenging levels of complexity to demonstrate, that our method does not depend on the analyzed task, thus presenting a major benefit compared to classical reasoner approaches. Another important aspect of our approach is its scalability and adaptability toward new activities, which can be learned on-demand.

Our semantic reasoning method can be extended to a more higher level for the generation of tasks. Since our system is able to on-line recognize and learn new activities on-demand, thus we can produce a graph of all related activities that produce a task. Therefore, our learning system extracts general task structures which together with the obtained knowledge can improve and accelerate the teaching of new tasks. Furthermore, we improved and validate this approach using a Virtual Reality system, which presents a realistic, cluttered, space in which a variety of tasks can be accomplished. Using task graphs also allowed the robot to utilize sparse sets of instructions to construct the complete set of steps necessary to carry out complex tasks [1]. Overall, the presented compact and flexible solutions are suitable to tackle complex and challenging problems for autonomous robots.

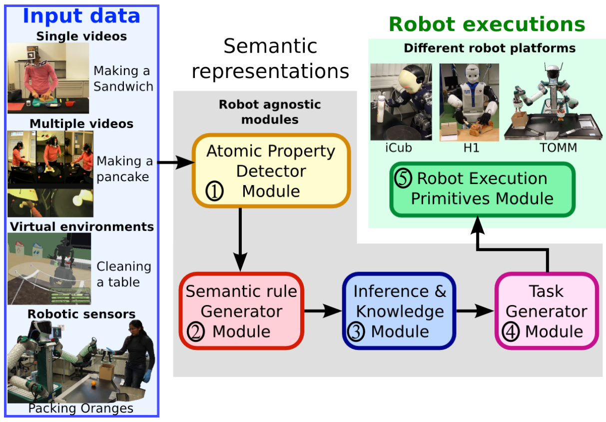


Figure 1: Main modules of our framework for the segmentation and recognition of human everyday activities using multiple sensors.

1. REFERENCES

[1] T. Bates, K. Ramirez-Amaro, T. Inamura, and G. Cheng. On-Line Simultaneous Learning and Recognition of Everyday Activities from Virtual Reality Performances. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, Sept 2017.

[2] E. Dean, K. Ramirez-Amaro, F. Bergner, I. Dianov, P. Lanillos, and G. Cheng. Robotic technologies for fast deployment of industrial robot systems. In *42nd IEEE Industrial Electronics Conference (IECON)*. IEEE, October 2016.

[3] K. Ramirez-Amaro, M. Beetz, and G. Cheng. Understanding the intention of human activities through semantic perception: observation, understanding and execution on a humanoid robot. *Advanced Robotics*, 29(5):345–362, 2015.

[4] K. Ramirez-Amaro, M. Beetz, and G. Cheng. Transferring skills to humanoid robots by extracting semantic representations from observations of human activities. *Artificial Intelligence*, 247:95–118, June 2017.

[5] K. Ramirez-Amaro, E. C. Dean-Leon, and G. Cheng. Robust semantic representations for inferring human co-manipulation activities even with different demonstration styles. In *Humanoids*, pages 1141–1146. IEEE, 2015.

[6] K. Ramirez-Amaro, H. N. Minhas, M. Zehetleitner, M. Beetz, and G. Cheng. Added Value of Gaze-Exploiting Semantic Representation to Allow Robots Inferring Human Behaviors. *ACM Transactions on Interactive Intelligent Systems*, 7(1):5:1–5:30, 2017.

Integrating Multi-modal Tactile Signals to a Compliant Control to Improve physical HRI

Emmanuel Dean-Leon and Gordon Cheng
Institute for Cognitive Systems, Technical University of Munich, Germany
Arcisstrasse 21, 80333, Munchen
dean@tum.de and gordon@tum.de

ABSTRACT

The development of breakthrough technologies helps the deployment of robotic systems in the industry. The implementation and integration of such technologies will improve productivity, flexibility and competitiveness, in diverse industrial settings specially for small and medium enterprises. In this talk, we present a framework [1] that integrates three novel technologies, namely safe robot arms with multi-modal and auto-calibrated robot skin [4], an end-to-end approach for transforming multi-modal tactile signals into a compliant control to generate different dynamic robot behaviors [2], and an intuitive and fast teaching by demonstration method that segments and recognizes the robot activities on-line based on re-usable semantic descriptions [6]. A key component of our framework is a robot parametric modeling based on the artificial skin multi-modal sensors (proximity, force and acceleration) [5]. We validate our approach in our robot TOMM [3] with an industrial application.

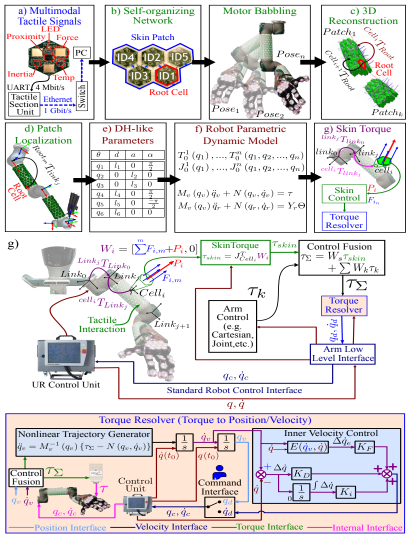


Figure 1: Main modules of our approach.

1. REFERENCES

[1] E. Dean, K. Ramirez-Amaro, F. Bergner, I. Dianov, P. Lanillos, and G. Cheng. Robotic technologies for fast deployment of industrial robot systems. In *42nd IEEE IECON*. IEEE, Oct. 2016.

[2] E. Dean-Leon, F. Bergner, K. Ramirez-Amaro, and G. Cheng. From multi-modal tactile signals to a compliant control. In *2016 IEEE-RAS Humanoids*.

[3] E. Dean-Leon, B. Pierce, F. Bergner, P. Mittendorfer, K. Ramirez-Amaro, W. Burger, and G. Cheng. Tomm: Tactile omnidirectional mobile manipulator. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2441–2447, May 2017.

[4] P. Mittendorfer, E. Dean, and G. Cheng. 3D spatial self-organization of a modular artificial skin. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3969–3974, Sept 2014.

[5] P. Mittendorfer, E. Dean, and G. Cheng. Automatic robot kinematic modeling with a modular artificial skin. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 749–754, Nov 2014.

[6] K. Ramirez-Amaro, M. Beetz, and G. Cheng. Transferring skills to humanoid robots by extracting semantic representations from observations of human activities. *AI*, 247:95–118, June 2017.

Event-Driven Systems for Efficient Reactive Control with Large Scale Robot Skin

Florian Bergner, Emmanuel Dean-Leon and Gordon Cheng
Institute for Cognitive Systems, Technical University of Munich, Germany
Arcisstrasse 21, 80333, Munchen
florian.bergner@tum.de, dean@tum.de and gordon@tum.de

ABSTRACT

Tactile human-robot interaction is essential for collaborative robots in industrial, health-care and household application scenarios and increases the robot's intuitiveness, flexibility and safety. One key element of enabling intuitive interactions is the ability to manually guide the robot by touching it for e.g. in teach-in scenarios [4]. A promising approach to upgrade existing robot systems with such abilities is to use skin for robots and implement multi-contact controllers. For taking full advantage of skin, the skin has to cover the robot completely. However, the application of large scale skin induces new challenges: 1) solving issues in transmitting huge amounts of tactile information with low-latency and 2) processing huge amounts of tactile information in real-time. In this abstract we present our new event-driven approach to tackle these challenges. We developed a novel multi-modal event-driven robot skin [1, 2] and combined it with our novel efficient event-driven reactive skin controller for large scale robot skin [3]. Event-driven systems only sample, transmit and process information when the novelty of the information is guaranteed. This increases their efficiency in comparison to synchronous systems. We evaluated our system in a comprehensive performance evaluation with our robot TOMM. TOMM has two UR5 robot arms, each covered with 253 multi-modal skin cells. The results show that the event-driven reactive skin controller always outperforms the synchronous reference controller while both controllers show exactly the same response. When the robot is not moving then the event-driven controller reduces the CPU usage by 78% in comparison to the synchronous reference controller. When the robot is responding to contacts then the CPU usage reduces by 66%.

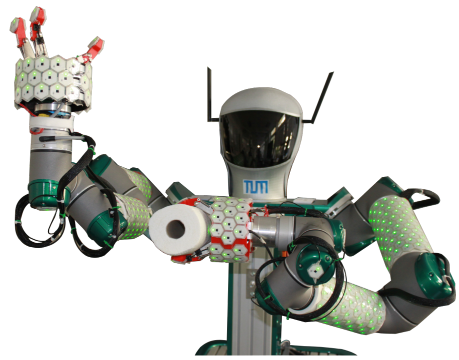


Figure 1: The robot TOMM [5] with two arms and two grippers covered with skin; the robot holds a paper towel in its left gripper which it uses to push the right arm; the right arm is controlled by our novel event-driven tactile reaction controller; it tries to avoid contacts and moves to the right.

1. REFERENCES

[1] F. Bergner, P. Mittendorfer, E. Dean-Leon and G. Cheng, "Event-based signaling for reducing required data rates and processing power in a large-scale artificial robotic skin", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2124-2129, 2015.

[2] F. Bergner, E. Dean-Leon and G. Cheng, "Event-based signaling for large-scale artificial robotic skin - Realization and performance evaluation", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4918-4924, 2016.

[3] F. Bergner, E. Dean-Leon and G. Cheng, "Efficient Event-Driven Reactive Control for Large Scale Robot Skin", in *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, pp. 394-400, 2017.

[4] E. Dean-Leon, K. Ramirez-Amaro, F. Bergner, I. Dianov, P. Lanillos and G. Cheng, "Robotic Technologies for Fast Deployment of Industrial Robot Systems", in *The 42nd Annual Conference of IEEE Industrial Electronics Society (IECON)*, pp. 6900-6907, 2016.

[5] E. Dean-Leon, B. Pierce, P. Mittendorfer, F. Bergner, K. Ramirez-Amaro, W. Burger and G. Cheng, "TOMM: Tactile Omnidirectional Mobile Manipulator", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, In Press.

Indeks avtorjev / Author index

Konferenca / Conference

Uredila / Edited by

Robotika / Robotics

Andrej Gams, Aleš Ude