

IS
20
19

Zbornik 22. mednarodne multikonference

INFORMACIJSKA DRUŽBA

Zvezek G

Proceedings of the 22nd International Multiconference

INFORMATION SOCIETY

Volume G

Robotika

Robotics

Uredila / Edited by
Andrej Gams, Aleš Ude

<http://is.ijs.si>

11. oktober 2019 / 11 October 2019
Ljubljana, Slovenia

Zbornik 22. mednarodne multikonference
INFORMACIJSKA DRUŽBA – IS 2019
Zvezek G

Proceedings of the 22nd International Multiconference
INFORMATION SOCIETY – IS 2019
Volume G

Robotika
Robotics

Uredila / Edited by

Andrej Gams, Aleš Ude

<http://is.ijs.si>

11. oktober 2019 / 11 October 2019
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

Na naslovnici je uporabljena slika robota podjetja 

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

Ljubljana, oktober 2019

Informacijska družba
ISSN 2630-371X

Katalogni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani COBISS.SI-ID=302474240 ISBN 978-961-264-167-2 (epub) ISBN 978-961-264-168-9 (pdf)

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2019

Multikonferenca Informaci družba (<http://is.ijs.si>) je z dvaindvajseto zaporedno prireditvijo tradicionalni osrednji srednjeevropski dogodek na področju informacijske družbe, računalništva in informatike. Informacijska družba, znanje in umetna inteligenca so - in to čedalje bolj – nosilci razvoja človeške civilizacije. Se bo neverjetna rast nadaljevala in nas ponesla v novo civilizacijsko obdobje? Bosta IKT in zlasti umetna inteligenca omogočila nadaljnji razcvet civilizacije ali pa bodo demografske, družbene, medčloveške in okoljske težave povzročile zadušitev rasti? Čedalje več pokazateljev kaže v oba ekstrema – da prehajamo v naslednje civilizacijsko obdobje, hkrati pa so notranji in zunanji konflikti sodobne družbe čedalje težje obvladljivi.

Letos smo v multikonferenco povezali 12 odličnih neodvisnih konferenc. Zajema okoli 300 predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic in 500 obiskovalcev. Prireditvev bodo spremljale okrogle mize in razprave ter posebni dogodki, kot je svečana podelitev nagrad. Izbrani prispevki bodo izšli tudi v posebni številki revije Informatica (<http://www.informatica.si/>), ki se ponaša z 42-letno tradicijo odlične znanstvene revije.

Multikonferenco Informacijska družba 2019 sestavljajo naslednje samostojne konference:

- 6. študentska računalniška konferenca
- Etika in stroka
- Interakcija človek računalnik v informacijski družbi
- Izkopavanje znanja in podatkovna skladišča
- Kognitivna znanost
- Kognitonika
- Ljudje in okolje
- Mednarodna konferenca o prenosu tehnologij
- Robotika
- Slovenska konferenca o umetni inteligenci
- Srednje-evropska konferenca o uporabnih in teoretičnih računalniških znanostih
- Vzgoja in izobraževanje v informacijski družbi

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 institucijam, 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 2019 bomo sedmič podelili nagrado za življenjske dosežke v čast Donalda Michieja in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe je prejel prof. dr. Marjan Mernik. Priznanje za dosežek leta pripada sodelavcem Odseka za inteligentne sisteme Instituta »Jožef Stefan«. Podeljujemo tudi nagradi »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je dobil sistem »E-zdravje«, jagodo pa mobilna aplikacija »Veš, kaj ješ?!«. Čestitke nagrajencem!

Mojca Ciglarič, predsednica programskega odbora

Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2019

The Information Society Multiconference (<http://is.ijs.si>) is the traditional Central European event in the field of information society, computer science and informatics for the twenty-second consecutive year. Information society, knowledge and artificial intelligence are - and increasingly so - the central pillars of human civilization. Will the incredible growth continue and take us into a new civilization period? Will ICT, and in particular artificial intelligence, allow civilization to flourish or will demographic, social, and environmental problems stifle growth? More and more indicators point to both extremes - that we are moving into the next civilization period, and at the same time the internal and external conflicts of modern society are becoming increasingly difficult to manage.

The Multiconference is running parallel sessions with 300 presentations of scientific papers at twelve conferences, many round tables, workshops and award ceremonies, and 500 attendees. Selected papers will be published in the *Informatica* journal with its 42-years tradition of excellent research publishing.

The Information Society 2019 Multiconference consists of the following conferences:

- 6. Student Computer Science Research Conference
- Professional Ethics
- Human – Computer Interaction in Information Society
- Data Mining and Data Warehouses
- Cognitive Science
- International Conference on Cognitonics
- People and Environment
- International Conference of Transfer of Technologies – ITTC
- Robotics
- Slovenian Conference on Artificial Intelligence
- Middle-European Conference on Applied Theoretical Computer Science
- Education in Information Society

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 fifteenth year, the award for life-long outstanding contributions will be presented in memory of Donald Michie and Alan Turing. The Michie-Turing award was given to Prof. Marjan Mernik for his life-long outstanding contribution to the development and promotion of information society in our country. In addition, a recognition for current achievements was awarded to members of Department of Intelligent Systems of Jožef Stefan Institute. The information lemon goes to the “E-Health” system, and the information strawberry to the mobile application “Veš, kaj ješ?!” (Do you know what you eat?!). Congratulations!

Mojca Ciglarič, Programme Committee Chair
Matjaž Gams, Organizing Committee Chair

KONFERENČNI ODBORI

CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, Južna Afrika
Heiner Benking, Nemčija
Se Woo Cheon, Južna Koreja
Howie Firth, Škotska
Olga Fomichova, Rusija
Vladimir Fomichov, Rusija
Vesna Hljuz Dobric, Hrvaška
Alfred Inselberg, Izrael
Jay Liebowitz, ZDA
Huan Liu, Singapur
Henz Martin, Nemčija
Marcin Paprzycki, ZDA
Claude Sammut, Avstralija
Jiri Wiedermann, Češka
Xindong Wu, ZDA
Yiming Ye, ZDA
Ning Zhong, ZDA
Wray Buntine, Avstralija
Bezalel Gavish, ZDA
Gal A. Kaminka, Izrael
Mike Bain, Avstralija
Michela Milano, Italija
Derong Liu, Chicago, ZDA
Toby Walsh, Avstralija

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki
Marjetka Šprah
Mitja Lasič
Blaž Mahnič
Jani Bizjak
Tine Kolenik

Programme Committee

Mojca Cigliarič, chair
Bojan Orel, co-chair
Franc Solina
Viljan Mahnič
Cene Bavec
Tomaž Kalin
Jozsef Györkös
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
Vladislav Rajkovič
Grega Repovš
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
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

KAZALO / TABLE OF CONTENTS

Robotika / Robotics	1
PREDGOVOR / FOREWORD	3
PROGRAMSKI ODBORI / PROGRAMME COMMITTEES	5
Autonomous learning of assembly policy / Simonič Mihael, Ude Aleš, Nemeč Bojan	7
Grasp Detection for Human-to-Robot Object Handover / Wohlhart Lucas	11
Autonomous adaptation to changes in production demands with a reconfigurable robot workcell / Gašpar Timotej, Deniša Miha, Radanovič Primož, Ude Aleš	12
Challenges of Collaborative Mobile Manipulation for Industrial Applications / Brandstötter Mathias, Dieber Bernhard, Lucchi Matteo, Mühlbacher-Karrer Stephan, Pichler Horst	16
Resilience in mobile manipulation / Haspl Thomas, Dieber Bernhard, Breiling Benjamin	17
Visual Feedback and Learning for Optimal Velocity of Robotic Visual Quality Inspection / Gams Andrej, Reberšek Simon, Ude Aleš	18
Intuitive Hand-Guidance of a Mobile Manipulator / Weyrer Matthias, Hofbaur Michael	22
Learning Robotic Handwriting with Convolutional Image-to-Motion Encoder-Decoder Networks / Ridge Barry, Pahič Rok	23
Indeks avtorjev / Author index	27

Zbornik 22. mednarodne multikonference
INFORMACIJSKA DRUŽBA – IS 2019
Zvezek G

Proceedings of the 22nd International Multiconference
INFORMATION SOCIETY – IS 2019
Volume G

Robotika
Robotics

Uredila / Edited by

Andrej Gams, Aleš Ude

<http://is.ijs.si>

11. oktober 2019 / 11 October 2019
Ljubljana, Slovenia

PREDGOVOR

Pod okriljem multikonference »Informacijska družba« po letu premora 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, ki pa se z razvojem znanosti tudi spreminjajo. Niso več nemi, neodzivni mehanizmi v kletkah temveč sodelavci, zaenkrat v industriji, kmalu pa že doma. Tako sodelovanje ima svoje varnostne zahteve, ki so postale pomemben del moderne robotike. Hkrati s razvojem robotike v industrijskih okoljih, se razvija tudi robotika povsod drugod. Ne presenečajo 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 priznanega in Instituta za robotika in mehatroniko z JOANNEUM Research Inštituta v Celovcu. 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 continuing its biannual tradition, is 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. However, with advances of science, industrial processes and the role of robots are also changing. Robots are not anymore mute, unresponsive mechanisms in cages, but coworkers. Thus far in the industry, but sooner rather than later, they will take this role in our homes as well. Such collaborative robotics brings about also its own demands for safety, which are becoming an important topic of modern robotics. Together with the change of robotic role in industrial processes, robotics is changing everywhere. 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 Institute for Robotics and Mechatronics from JOANNEUM Research, Klagenfurt, Austria. 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

Autonomous learning of assembly policy

Mihael Simonič
Jožef Stefan Institute
Ljubljana, Slovenia
mihael.simonic@ijs.si

Aleš Ude
Jožef Stefan Institute
Ljubljana, Slovenia
ales.ude@ijs.si

Bojan Nemec
Jožef Stefan Institute
Ljubljana, Slovenia
bojan.nemec@ijs.si

ABSTRACT

In the paper, we propose to learn an assembly task from the corresponding disassembly. Autonomous learning of disassembly can be easier than learning of the corresponding assembly task, because the admissible set of motions during disassembly is initially fully constrained by the environment. During the disassembly the robot exploits its compliance in order to detect admissible motions and takes appropriate decisions when multiple options exist. Learning of the disassembly was realized using hierarchical reinforcement learning. The disassembly policy is then used to derive the corresponding assembly policy. The proposed approach was experimentally validated on the case of light-bulb assembly.

KEYWORDS

reinforcement learning, robot learning, autonomous assembly

1 INTRODUCTION

Developing robust assembly skills is one of the main challenges in contemporary robotics. Assembly skills are needed not only in production plants, but will also be important for the future generation of home and service robots. For fast deployment of such tasks, new user-friendly tools for programming robot operations are needed. Ideally, a robot would be able to derive assembly policy autonomously.

Autonomous policy learning, is usually accomplished by utilizing reinforcement learning. Starting from an existing parameterized policy, a robot tries to adapt to a new situation by randomly changing task parameters and find out how to modify the policy to maximize the reward function [9, 12]. However, the main challenge is huge search space which characterizes an assembly policy. For that reason, there were very few successful attempts of completely autonomous learning of assembly tasks in robotics [4, 7]. Existing techniques for reducing the search space of reinforcement learning usually assume prior information about process, either in an explicit form or inherited from previous experiments and therefore still rely on skilled robot operators that guide the robot through the learning process [5].

In our previous research, we proposed an alternative approach to autonomous policy learning, which unifies compliant motion control and reinforcement learning. Tasks that involve interaction with the environment are traditionally considered as extremely hard to learn due to the unknown and possibly changing environment. On

the other hand, interacting with the environment can be advantageous to accelerate the learning process. Namely, if appropriately addressed, learning of physically constrained tasks is more efficient than the learning of tasks, where a robot can move completely freely in space. The reason is that the environment limits the admissible movement directions. Consequently, the number of parameters that need to be learned can be greatly reduced. To implement this type of learning, we need to make use of the natural robot motion along with the constraints imposed by the environment. Compliant robot control provides a suitable framework for implementing such a strategy. This concept has been already successfully applied to the learning of tasks such as autonomous learning for doors and drawers opening [8].

In this paper, we present how the above-described methodology can be extended to autonomous learning of assembly operations. The main idea is that robot first learns the reverse action – disassembly of an object. In an assembled object, the set of possible motions is constrained, and typically only a single motion or operation is possible. During the disassembly, the motion becomes less and less constrained until the part is completely disassembled and the environment no more constrains motion of individual parts. The situation is opposite during the assembly. The initially virtually unlimited set of possible motions becomes more and more constrained as the assembly process advances. Given no previous knowledge about the task, learning of disassembly is therefore more straightforward than learning of the assembly task. Imagine generic peg-in-hole task: by removing a peg from a hole, we also learn the exact pose of the hole, whereas we would first have to guess where the hole is if we are to insert the peg into the hole without any prior knowledge.

Similarly, we transfer the knowledge obtained during disassembly to the corresponding assembly process. We assume that the initial assembly policy can be obtained by reverse execution of the learned disassembly policy. This is possible because in most cases assembly and disassembly are mutually reversible operations. Common assembly tasks such as putting/placing, peg-in-hole, or screwing are directly reversible [6]. Tasks that result in structural deformations or require external equipment (e.g. riveting pistol and rivets) are not directly reversible, but can be omitted for the purposes of disassembly learning and manually added to the final assembly policy.

This paper is structured as follows. We first introduce our algorithm for hierarchical reinforcement learning on the example of maze learning in Section 2. Then we present the underlying intelligent controller in Section 3. In section 4, we present our methodology to learn assembly policy from disassembly policy, along with experiential verification of the proposed framework in Section 5. We conclude with a short summary.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IS2019, October 7–11, 2019, Ljubljana, Slovenia, Europe

© 2019 Copyright held by the owner/author(s).

2 HIERARCHICAL REINFORCEMENT LEARNING

In the reinforcement learning (RL) literature, maze learning has been traditionally used as a benchmark for validating learning algorithms. Maze learning also bears a lot of similarities with disassembly process, where the robot should come from an initially fully contained state into the final unconstrained state. Within a maze, the agent mostly follows the corridors and only has to take decisions in the crossings.

Traditional approaches rely on discrete state-space with predefined set of actions as illustrated in Figure 1.

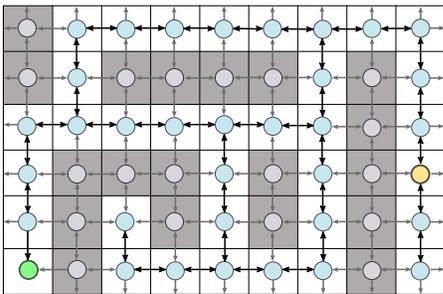


Figure 1: An example of maze with 9×11 cells. White cells represent corridors where the agent can move, whereas gray cells are walls. The state space for maze learning is represented with a graph. In each state (represented as node), the agent can choose from a fixed set of possible actions (represented with edges): relative left, right, up and down movement. The agent starts in the yellow node and should learn to exit maze (arrive at green node).

In contemporary robotics we need continuous policies. Within the traditional RL framework, an approximation of continuous policy can be achieved by increasing the number of states and actions, which substantively decreases the performance of the learning.

Considering the example in Figure 1, we can notice that in the discretization of the maze many of the states are redundant and the robot can not access them (wall cells). Following the corridor, the agent eventually arrives either at a crossing, in a dead-end or to the target. This suggests, that also the states between two crossings and between a crossing and a dead-end or the target can be left out.

Therefore, we propose to dynamically assign states rather than allocate them in advance. A suitable framework to achieve this is hierarchical reinforcement learning, where we combine RL with control algorithm as shown in Figure 2.

The upper hierarchical level is classical RL algorithm, where the states are discovered online by the lower hierarchical level. The later consists of an intelligent compliant controller, which autonomously moves within the environment constraints and detects where multiple movement options exist. The states for the upper level are only assigned when multiple options are possible. There are two main benefits of using such approach:

- The generated policies are inherently continuous.
- The number of states is greatly reduced.

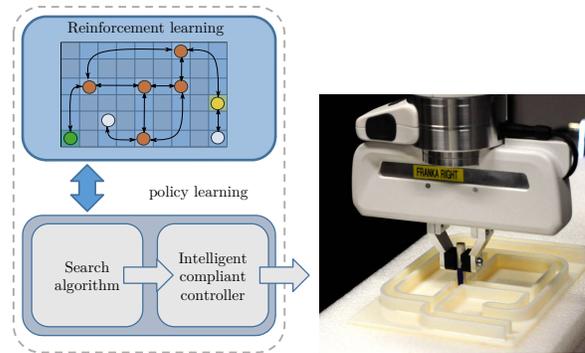


Figure 2: Block scheme of the proposed hierarchical policy learning algorithm. The upper level is RL of the policy, where the states and actions are represented with a directed graph. The lower level is an intelligent controller, consisting of a search algorithm and a Cartesian impedance controller.

The states and actions of hierarchical RL can be also represented with nodes and edges, respectively, of a directed graph as shown in the upper blue box in Figure 2.

3 INTELLIGENT COMPLIANT CONTROLLER

The lower level of the hierarchical learning utilizes a compliant control framework. As the robot moves along the boundaries, the controller searches for possible alternative movement directions.

In general, the physical constraints of the system are not known in advance. To find a feasible initial motion direction, the controller keeps applying force in random directions until this results in a movement. We then use operational space compliant controller to continue the motion in the initiated direction. The control parameters make the robot more compliant in directions orthogonal to the movement direction.

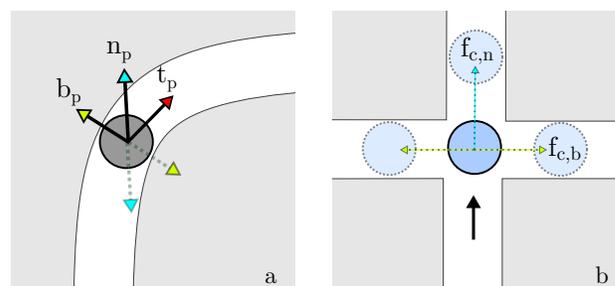


Figure 3: Searching path and possible states in restricted environment. The left part (a) shows Frenet-Serret frame attached to the end effector in the labyrinth. The right part (b) shows an instance, when the controller discovers a new state for reinforcement learning. Both parts show how search forces are applied in the normal and binormal direction.

We specify these directions using Frenet-Serret frames along the resulting motion trajectory [10] as illustrated in Figure 3 a. The Frenet-Serret frame can be expressed $R_p = [t_p \ n_p \ b_p]$ with

- the unit vector $t_p = \frac{\dot{p}}{\|\dot{p}\|}$ tangent to the curve, pointing in the direction of motion,
- the normal unit vector $n_p = \frac{\dot{p} \times \ddot{p}}{\|\dot{p} \times \ddot{p}\|} \times t_p$, and
- the binormal unit vector $b_p = n_p \times t_p$

where $p \in \mathbb{R}^3$ are the measured positions of the robot end-effector.

In order to follow environmental constraints, we exploit robot's compliance. We modified a passivity-based variant of impedance control for manipulators with flexible joints [2] by allowing to set the compliance along the operational space trajectory expressed using Frenet-Serret frame rather than global frame. The task command input $\lambda_c^v = [\ddot{p}_c^T, \dot{\omega}_c^T]^T$ is then given by:

$$\ddot{p}_c = -R_p D_p R_p^T \dot{p} + R_p K_p R_p^T e_p, \quad (1)$$

$$\dot{\omega}_c = -R_o D_o R_o^T \omega + R_o K_o R_o^T e_q, \quad (2)$$

where e_p and e_o are position and orientation tracking errors; K_p and $K_o \in \mathbb{R}^{3 \times 3}$ are the diagonal matrices, which define the positional and rotational stiffness in the Frenet-Serret and global frames, respectively. Likewise, D_p and $D_o \in \mathbb{R}^{3 \times 3}$ are diagonal damping matrices, which are set to $D = 2\sqrt{K}$ for critically damped system. For other parameters, please see [1].

By applying high positional gain in the direction of movement and low gains in the orthogonal direction, the robot can autonomously move along the environmental boundaries. However, following the constraints alone can not discover new states for the upper RL level. For this, small test forces are applied in the positive and negative directions of the normal and bi-normal (see Figure 3). All test forces are applied in each test position, which are placed in short intervals along the entire trajectory. If the robot moves above some predefined positional displacement threshold as a result of applying this forces in multiple directions in the same test position, the controller has found a new state (see Figure 3 b). In the new state each action corresponds to applying the specific force, which results in a movement in one of the admissible directions. The controller waits for the decision of RL algorithm, which action to take.

We assume that motion can be stopped only due to the task constraints. If the motion is interrupted, the controller searches for a new feasible motion by applying a random force in a random direction in the same manner as at the beginning. Following this strategy, the robot eventually generates a continuous policy.

4 ASSEMBLY LEARNING BY DISASSEMBLY

We can apply the same algorithm as for maze learning to disassembly operations. Key stages of disassembly and their analogies in the graph representation and hierarchical reinforcement learning are summarized in the Table 1.

A positive reward is given only when the robot has disassembled the object, i.e the target state. Negative reward is assigned when the robot arrived in a state where the motion could not be continued.

When the robot explores state s_k , the action-value function $Q(s_k, a_k)$ is updated according to the SARSA algorithm [11]:

$$Q(s_k, a_k) \leftarrow Q(s_k, a_k) + \alpha(r_k + \gamma Q(s_{k+1}, a_{k+1}) - Q(s_k, a_k)), \quad (3)$$

where s_k is the label of the k -th state, a_k is the label of the action taken in s_k , r_k is the reward obtained in state s_k , $0 < \alpha < 1$ is the learning gain and $0 < \gamma < 1$ is the discount factor, which gives recent rewards higher importance. The optimal policy can be obtained by applying ϵ -greedy strategy in the form

$$\pi(s) = \begin{cases} \arg \max_a Q(s, a), & \text{with probability } 1 - \epsilon, \\ \text{random action}, & \text{with probability } \epsilon, \end{cases} \quad (4)$$

where parameter ϵ is the ratio between the exploration and exploitation [12].

Using the hierarchical reinforcement learning, the robot not only learns the disassembly policy, but identifies all crucial stages for the corresponding assembly process.

We assume that assembly and disassembly are mutually reversible operations, therefore we obtain initial assembly policy by merely reversing the disassembly policy. However, even if the operation is reversible small deviations in part geometry, grasping, material, etc. can result in failure. To account for this, we have to apply appropriate control together with the exception strategies, which mimic human behavior during the assembly.

We set high gains in all spatial directions until the parts to be assembled are in contact. This assures precise path tracking during the approach motion in assembly. When the parts are in contact, we use the same compliance settings as during disassembly.

During the assembly, we measure contact forces and torques and compare them with the measured forces and torques during disassembly. Note that the forces/torques during assembly have the opposite sign in relation to those measured at disassembly. If the values are still notably different, we slow down the motion and if the forces/torques are still increasing, we carry out a trajectory in the

Table 1: Key stages of disassembly and their analogies in hierarchical reinforcement learning and graph representation

Observation	Lower level	Upper level	Graph
Fully assembled product.	Controller tries to move in different directions and thereby determines admissible directions.	Start state	Yellow node
Partially disassembled product.	Controller follows the environmental constraints and moves in the only admissible direction.	Action	Edge
Multiple options to continue disassembly.	Controller tries to move in different directions and thereby determines admissible directions.	State	Orange node
Disassembly cannot be continued in the same direction.	Goes in reverse direction.	Penalty state	White node
Fully disassembled product.	Controller can freely move.	Target state	Green node

opposite direction for some time and then try again, as suggested in [6].

For improving the obtained policies many different methods exists. We apply iterative learning control, which has proven useful for on-line adaptation of force profiles in manipulation tasks [1].

5 EXPERIMENTAL VERIFICATION

We experimentally verified the proposed disassembly learning on a Franka Emika Panda robot. The control algorithm was implemented as a `ros_control` plugin in C++ using `libfranka`[3], while the learning algorithm was implemented in Matlab as a ROS node.

We verified the proposed approach using a R5W car bulb and corresponding plastic casing, used to fix the bulb above the registration plates. The R5W bulb is mounted into the plastic casing using bayonet mechanism as shown in Figure 4.

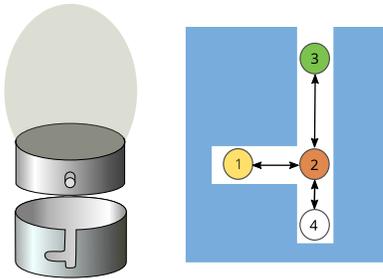


Figure 4: On the left illustration of a bayonet bulb with the corresponding casing is shown. Bayonet mechanism consist of radial pins, and a matching slot and spring to keep the two parts locked together. On the right, a projection of the slot in the casing to the plane is shown along with states than can be discovered by the controller. In disassembly task in order to release the lock, the robot first has to rotate the bulb across the horizontal part of the slot and then the pin slides into the vertical part of the slot. By lifting it upwards, the robot eventually learns to remove the bulb.

This example shows why disassembly can be easier than the assembly. In disassembly, the robot starts in state 1, and the only decision it has to make is in the state 2 to arrive in the state 3. In assembly, however, it has first to learn the proper pose of the state 3 and then search for the state 2.

The robot learns to remove the bulb from the casing as shown in Figure 5.

Applying the procedure described in Section 4, the robot successfully learns the assembly operation - bulb insertion.

6 CONCLUSIONS

Physical constrains can be used to structure and reduce search space for reinforcement learning. During the disassembly the motion of object parts is more constrained. As a consequence, learning of disassembly can be easier than learning of assembly.

Hierarchical reinforcement learning, consisting of high level decision making and intelligent compliant controller, has proven to be an efficient framework for learning in the constrained environments, such as disassembly processes. The controller exploits its

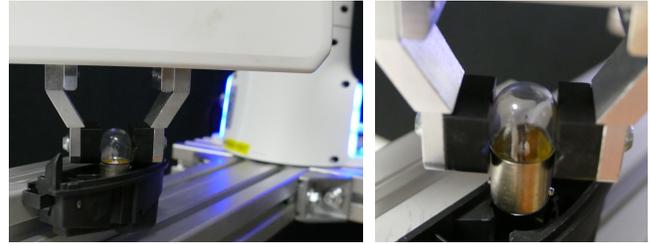


Figure 5: On the left the bulb is mounted in the casing. On the right the bulb is removed from the casing revealing its two radial pins.

compliance in order to detect admissible motions. When motion in multiple directions is possible, decisions are taken at the upper hierarchical level.

The proposed approach was experimentally validated on the case of light-bulb insertion. During the disassembly (bulb removal from the casing), all crucial stages for the corresponding assembly process (bulb insertion) can be learned autonomously and simplify the assembly learning.

Our future research will focus on evaluation of the proposed method for objects, composed of multiple parts.

ACKNOWLEDGMENTS

The research leading to these results has received funding by the EU Horizon 2020 Research and Innovation Programme under grant agreement No 820767, project CoLLaboratE.

REFERENCES

- [1] Fares J. Abu-Dakka, Bojan Nemeč, Jimmy A. Jørgensen, Thiusius R. Savarimuthu, Norbert Krüger, and Aleš Ude. 2015. Adaptation of manipulation skills in physical contact with the environment to reference force profiles. *Autonomous Robots* 39, 2 (2015), 199–217.
- [2] A. Albu-Schaffer, C. Ott, and G. Hirzinger. 2007. A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. *The International Journal of Robotics Research* 26, 1 (2007), 23–39.
- [3] Franka Emika. 2019. `libfranka`: C++ library for Franka Emika research robots. <https://github.com/frankaemika/libfranka>.
- [4] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana. 2017. Deep reinforcement learning for high precision assembly tasks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 819–825.
- [5] J. Kober, J. a. Bagnell, and J. Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (aug 2013), 1238–1274.
- [6] J. S. Laursen, L.-P. Ellekilde, and U. P. Schultz. 2018. Modelling reversible execution of robotic assembly. *Robotica* 36, 5 (2018), 625–654.
- [7] Sergey Levine, Nolan Wagener, and Pieter Abbeel. 2015. Learning Contact-Rich Manipulation Skills with Guided Policy Search. *International Conference on Robotics and Automation (2015)*, 156–163. arXiv:1501.05611
- [8] B. Nemeč, L. Zlajpah, and A. Ude. 2017. Door opening by joining reinforcement learning and intelligent control. In *18th International Conference on Advanced Robotics (ICAR)*. 222–228.
- [9] Jan Peters. 2010. *Policy Search for Motor Primitives in Robotics*. Technical Report.
- [10] R. Ravani and A. Meghdari. 2006. Velocity distribution profile for robot arm motion using rational Frenet-Serret curves. *Informatica* 17, 1 (2006), 69–84.
- [11] Gavin A Rummery and Mahesan Niranjan. 1994. *On-line Q-learning using connectionist systems*. Vol. 37. University of Cambridge, Department of Engineering Cambridge, England.
- [12] Richard S. Sutton and Andrew G. Barto. 2015. *Reinforcement Learning: An Introduction, Second edition*. The MIT Press, Cambridge, London.

Grasp Detection for Human-to-Robot Object Handover

Lucas Wohlhart
JOANNEUM RESEARCH ROBOTICS
Lakeside B08a, EG
9020 Klagenfurt am Wörthersee
lucas@wohlhart.at

ABSTRACT

This project presents an attempt to apply current state-of-the-art methods for grasp pose estimation to human-to-robot handover scenarios. The implemented method shall enable a robotic mobile manipulator to perform antipodal grasps on previously unknown objects presented by a human collaborator.

1. INTRODUCTION

Grasping is to be considered one of the fundamental object manipulation tasks a robot has to perform. In a **human-robot collaboration** scenario with a human giver handing over an object to a robot receiver the perception task is to determine the desired object transfer point and a corresponding grasp pose. This has proven to be challenging especially when facing unknown objects in unstructured environments. Driven by applications in fields such as warehouse automation or flexible manufacturing, recent advances in object agnostic robotic bin picking, mainly inspired by vision-based deep learning techniques, suggest that currently proposed methods are increasingly capable of solving these **grasp synthesis** tasks.

Mahler et al. [1] trained a neural network, dubbed grasp quality CNN (GQ-CNN), to learn the evaluation of a grasp success probability. The model is trained on the DexNet-2.0 dataset; an extensive collection of synthesized RGB-D images annotated with corresponding grasp configurations. By iteratively ranking and resampling grasp candidates this method has shown to yield good proposals for unknown real world objects. Morrison et al. [2] propose a fully convolutional generative grasp CNN (GG-CNN) estimating individual maps for grasp quality, gripper angle and gripper width from a given 2 image. The resulting best grasp is determined by choosing the gripper configuration corresponding to the highest success probability encountered in the grasp quality map.

2. METHOD

Our method builds on the idea of estimating grasp configuration maps as in GG-CNN and extends the approach by adding a semantic segmentation layer to enforce scene understanding. This acts as guidance to focus on the region of interest for the object handover task and avoid estimating grasps that would collide with the hand of the human collaborator. The proposed fully convolutional neural network architecture is based on a U-Net inspired structure featuring encoder and decoder each comprised of four residual network

blocks connected by an atrous spatial pyramid pooling layer to foster scale invariance. At the input stage, the network is fed a depth map acquired by an RGB-D sensor. The multi-headed output consists of a pixelwise semantic segmentation classifying as background, hand or object, a grasp center point quality map, a grasp angle map and a gripper opening width map. To obtain training data we extend the pipeline of DexNet 2.0 by combining it with the hand pose estimation data synthesis approach of Riegler et al. [3]. This enables us to render depth images and segmentation masks and to annotate corresponding grasp rectangles as introduced by Jiang et al.[4] for scenes in which a human presents an object to hand over in various poses and viewpoints.

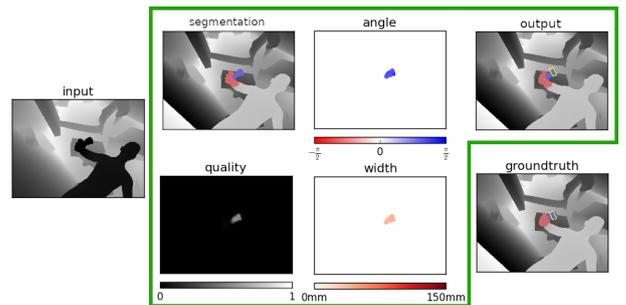


Figure 1: Left: input depth image. Green box: intermediate segmentation and grasp maps, resulting output estimated grasp configuration. Bottom-right: ground truth segmentation and grasp configuration

We are currently constructing a data acquisition pipeline to capture real world ground truth using RGB-D sensors, to bridge the sim-to-real gap due to noisy sensors by fine-tuning the trained model on such data.

3. REFERENCES

- [1] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics.
- [2] D. Morrison, P. Corke, and J. Leitner. Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach.
- [3] G. Riegler, D. Ferstl, M. Rütger, and H. Bischof. A framework for articulated hand pose estimation and evaluation.
- [4] Yun Jiang, S. Moseson, and A. Saxena. Efficient grasping from RGBD images: Learning using a new rectangle representation.

Autonomous adaptation to changes in production demands with a reconfigurable robot workcell

Timotej Gašpar
Miha Deniša
Primož Radanovič
Aleš Ude
timotej.gaspar@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

ABSTRACT

Current challenges in automation represent automating low-batch production processes where changes in the production parameters happen frequently. These type of production are often happening in Small and Medium-sized Enterprises, which have many time been dismissed as potential end user of automation technologies. This was mainly due to the high costs of setup, both in terms of the costs of the equipment and time required to set it up. In this paper we present a new type of reconfigurable robot workcell for fast set-up of automated assembly processes for SMEs. By developing passive reconfigurable elements and integrating intuitive programming by demonstration methodologies we were able to reduce the costs and set-up times for the automation of few-of-a-kind manufacturing processes without losing the flexibility of the system to cope with changes in market demands.

KEYWORDS

robotics, reconfigurability, ROS, assembly

1 INTRODUCTION

The trend of incorporation robots into manufacturing processes is on the rise. While high cost of process automation does not represent a significant challenge for large enterprises, *Smaller and Medium-sized Enterprises* (SME) might not undertake such an investments. Beside the price of the robots and the necessary accompanying hardware for automation, the cost of the time spend on the integration of robotic systems can also be high. Another hurdle for automatization of processes in SMEs is the need for quick adaptation to ever changing market demands. The paradigm of Reconfigurable Manufacturing Systems (RMS) [6] addresses the efficient and quick adaptation of the production process. Although a RMS can have a more complex design and achieve a lower throughput as classic automation approaches, they proved to be more applicable in processes with the need for often changes [10]. But in order to make RMS affordable for SMEs, a high investments cost of incorporating them in the manufacturing process needs to be avoided [3].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IS2019, October 7–11, 2019, Ljubljana, Slovenia, Europe
© 2019 Copyright held by the owner/author(s).

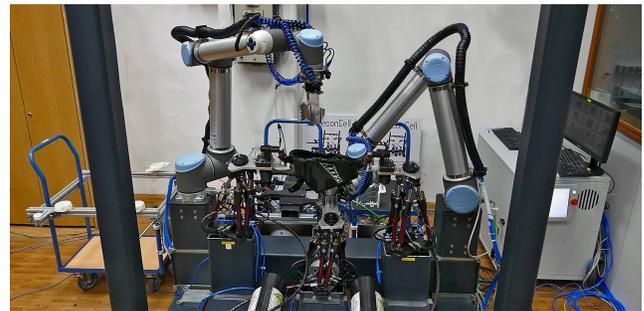


Figure 1: The proposed reconfigurable robot workcell executing an example assembly process.

The goal of the presented system is to offer a reconfigurable robot workcell in line with the RMS paradigm. The workcell must be appropriate for SMEs, where low-volume high-diversity production often takes place. The proposed systems combines a reconfigurable ROS-based software architecture and novel hardware elements that offer cost efficient solutions to reconfigurability. In addition, programming by demonstration methods for teaching of robots assembly skills are included in order to reduce the setup time.

While novel approaches in hardware design for reconfigurable robot workcells are presented in section 2, section 3 describes the software architecture of the cell. Section 4 presents technologies for fast set-up times and intuitive robot programming. Concluding remarks and implementation results are given in the last section.

2 RECONFIGURABLE HARDWARE

While designing the reconfigurable robot workcell in line with the RMS paradigm several aspects need to be taken into consideration: the desired physical properties (size, stiffness, robot workspace, etc.); available factory space; the integration of the workcell into the establish production process without too many significant changes; and the ability of the cell to quickly adapt to changing demands in the process. To ensure the workcell's ability for reconfiguration and adaptation in an affordable way, we introduced several passive reconfigurable hardware component as an alternative to off-the-shelf solutions.

The **reconfigurable frame** of the workcell connects the robot to peripheral modules. Several requirements need to be taken into account while designing the frame. While the cell's stiffness is paramount, as even small frame deformations can result in large positioning errors, the structure must be easily adaptable to ensure the needed reconfiguration. The affordability of such a solution should also be taken into account. To ensure the stiffness of the frame structure can be comparable with welded joints and at the same time enable simple assembly and modifications, rectangular steel beams in combination BoxJoint connectors [7] were selected.

Reconfigurable robot tools were also introduced. A mounted tool exchange system at the robot's end effector enables a vast array of assembly operations by un/equipping tools needed for various steps in the process. Besides ensuring a stiff coupling between the robot and the tool, the exchange systems provides electrical power, Ethernet connectivity, and pressurised air to the tool.

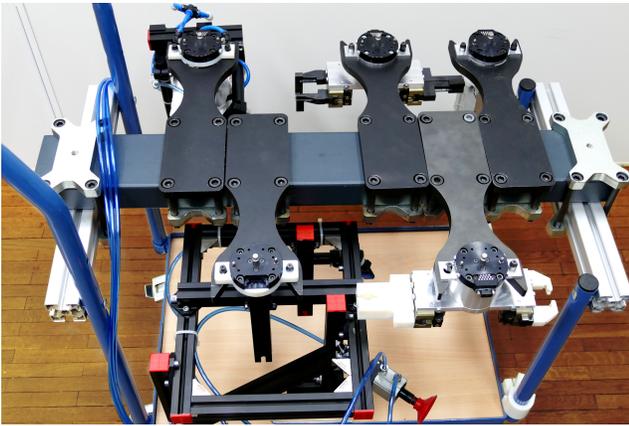


Figure 2: Various robot tools mounted on a rack. The robot can attach the one needed for the current task.

Special “**Plug & Produce**” (PnP) connectors were developed to ensure the connectivity to peripheral modules. These peripheral modules are crucial in a reconfigurable environment, as they enhance the cell with various functionalities. Used modules can include various fixtures, material flow management, tool storage, etc. These modules need to be introduced or removed from the workcell as quick as possible with as little disturbance to the process as possible. The design of the PnP connectors provides a highly repeatable, stiff, and quick mechanical coupling of the modules to the cell. Beside a mechanical coupling, connectors enable the transfer of data, pressurised air, and electrical power. This enables the peripheral modules to be self sufficient and connect to the overall structure of the cell as quickly as possible. While PnP connectors allow us to introduce new modules into the workcell such modules often need to be introduced manually, and can not be regarded as fully autonomous.

A concept of **passive reconfigurable elements** introduces needed reconfiguration into the cell, while reducing the cost of the elements. In contrast to standard off-the-shelf solutions, which often include active components, these passive reconfigurable elements do not contain any actuators or sensing equipment. By removing these

components the price tag is lowered. To compensate for the missing sensors and actuators in these passive elements, robot is used in the reconfiguration step. A number of passive hardware components were used in various assembly operations. One example of a passive reconfigurable element is a passive rotary table (depicted in Fig. 3). By rotating the table, the workpiece on the table is oriented in the desired way. This is achieved by releasing the table's brakes, re-orienting it by the robot arm, and engaging the brakes as the desired orientation is reached. The last orientation of the table is stored by storing the robot's position.

3 RECONFIGURABLE SOFTWARE

Providing connectivity with respect to the data flow is another paramount issue for a proper workcell. Peripheral modules should be connected to the workcell and between each other, in order to receive and broadcast data and instructions. The data should be parsable by all software components within the system. To ensure the software modularity and connectivity, the proposed software architecture is ROS-based. The software system architecture of the workcell is depicted in Figure 5.

A **robot workcell ROS backbone** was implemented to ensure the needed connectivity. Just the data flow between all the modules is not enough to achieve the desired modularity of the system. The data should be structured in a way that is parsable by all the modules in the workcell. The suitable framework is offered by the Robot Operating System (ROS) [8], which enables the development of software components that need to share data over the common network. In addition it allows monitoring and controlling the complete workcell.

ROS-based modules ensure that they are all connected within the workcell through the ROS network. All modules are equipped with the computational hardware that enables running ROS *nodes*. This means each module's data and functionalities are available through the workcell ROS network. They are denoted as Micro computer in Fig. 5. A top-level task scheduling software can control all modules as soon as they are plugged into the cell. They are connected to the cell using the described hardware components (PnP connectors or tool exchange systems).

Low level real-time robot control is another crucial part of the proposed reconfigurable workcell. To follow the previously described paradigms of seamless integration of all the hardware components in the workcell, robots should be treated as a ROS enabled module. While industrial robots are equipped with a control box that provides real-time control, most of them do not offer support for running ROS nodes and in turn are not able to communicate over the ROS network. A special communication layer that connects the robot module to the rest of the ROS network was implemented. In order to not make the workcell robot-specific an abstraction layer that supports different types of robots was introduced. It enable programming of new strategies through a suitable control interface and various trajectory and feedback control strategies. Again, independently of the selected robot. This abstraction layer enhances the overall modularity of the cell.

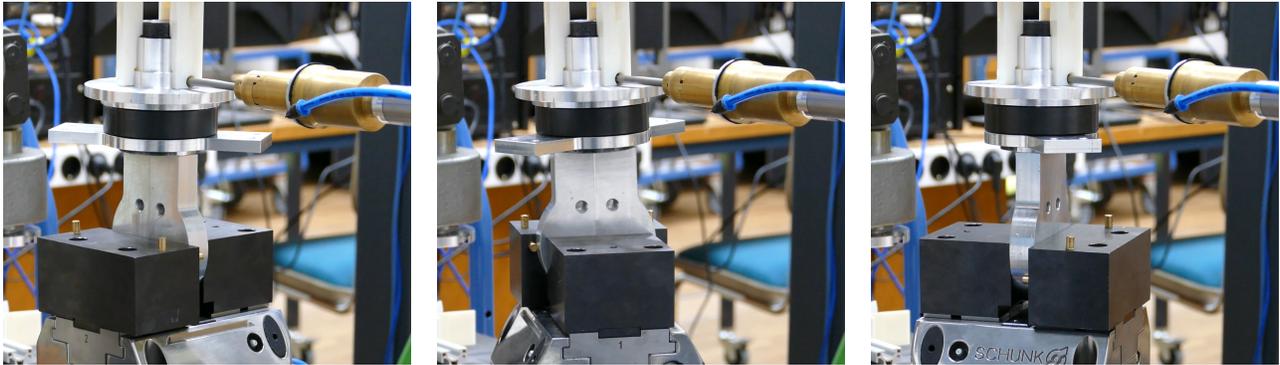


Figure 3: An example use of the passive rotary table. As can be seen in the figure, the table is being used to fasten screws on 3 different sides of a workpiece. As it would be impossible to reach the object on all three sides with the screwdriver, due to the kinematic restrictions, the table needs to be used.

4 ASSEMBLY SEQUENCES

During a set-up of a classic assembly workcell, a significant portion of the time is dedicated to determining, writing and compiling of the assembly sequence. In order to reduce the set-up time and enable short reconfiguration times between different assembly processes, this should be done as fast as possible. In this section we present a set of technologies that facilitate and accelerate the programming of robot workcell assembly operations.

Learning of **assembly skills through programming by demonstration** (PbD) enables defining the robot motions for a complete assembly process in an intuitive and faster way for non-expert users. PbD provides an approach to define these motions in a natural way and avoids coding complex programs in a robot-oriented programming language [1, 2, 4]. The two present PbD approaches are kinesthetic teaching and remote guidance.

With kinesthetic guidance the user moves the robot by physically guiding through its workspace and thus showing the desired movement. This approach is commonly used in collaborative robotics as it is effective to use with robots which have torque-controlled actuators [5]. The quality of the dynamic model and torque sensors greatly impact the ease of guidance and the needed physical effort. This in turn affects the quality of the shown movement and the smoothness of the demonstrated path.

While useful, the drawbacks of non-perfect kinesthetic guidance represent an inconvenience when working towards methods to shorten times of robot programming. As a result, a large amount of time can be spent to achieve the desired movement and/or configuration of the robot. To mitigate these drawbacks a remote control interface was developed and integrated in the workcell. A displacement of the analogue sticks of a consumer grade joystick was mapped to the Cartesian space velocities. This allows the user to control the robot in a smooth way and can mitigate the drawback of kinesthetic guidance when needed.

A **database of assembly skills** acquired during PbD should be accessible throughout the entire software framework of the workcell. In order to handle storing and loading of the learned skills, MongoDB database was integrated into our system. Whenever a new skill is learned, a new named entry is created in the database.

An assembly sequence can then read the desired database entry from the database and move the robots accordingly. If we wish to update a certain skill, we can simply overwrite the entry with a newly modified skill. In this way, we avoid changing the top-level assembly sequence program.



Figure 4: A consumer grade joystick interface that we used to perform precise motions of the robot in Cartesian space.

State machine assembler is a crucial part of any workcell. An engine for state machine code generation was developed to further accelerate the programming process of the workcell assembly sequence. While there are numerous ROS-based packages aimed at facilitating the high-level task programming by using state machines, defining complex robot behaviours with these tools can be complex. It requires a programmer to dedicate his attention to the structure of the state machine, the basic code, and the programming language syntax. To expedite and enhance this process, a method for code generation, a meta-scripting and templating method was developed to speed up the process. The details on this are omitted in this paper and the reader is referred to our previous work on this topic [9].

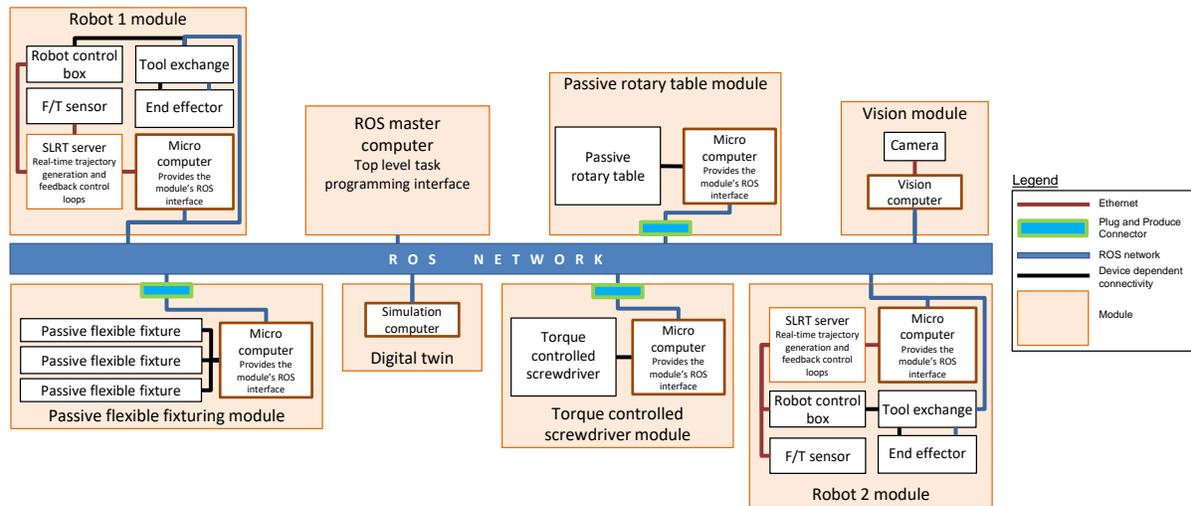


Figure 5: Software architecture for the reconfigurable robot workcell with various software and hardware modules.

5 CONCLUSION

In this paper we present a new type of robot workcell which is highly reconfigurable with innovative hardware concepts and components with a ROS-based software backbone. Throughout the work that lead to the presented results we focused not only towards providing methods for autonomous reconfiguration of the cell in order to adapt to production changes, but also to shorten set-up times by implementing various programming by demonstration technologies. In order to show the industrial relevance of our work we evaluated the proposed paradigms, the underlying technology and the overall quality of the cell through the implementation of various use-cases. The use-cases were provided by SMEs from different fields of industry and our task was to automate part of the production line that is currently done either manually or does not possess the desired flexibility. These use-cases range from the (1) assembly of automotive headlights, (2) the assembly of linear drives, (3) the assembly of a robotic gripper, (4) assembly of airport runway lights and finally the (5) assembly of printed circuit boards (PCBs). The successful implementations provided us with the overall proof that the developed solution are of interest in the industry. We were also able to acquire the first reference key performance indicators, e. g. cycle time, reconfiguration time, setup-time, etc. Throughout the implementation of the various use-cases some of the key equipment stayed the same (i.e. robots, tool rack, etc.), however other parts of the cell were reconfigured according to the requirements of each experiment. Some application-specific periphery modules were either added or removed.

ACKNOWLEDGEMENTS

This work has received funding from the EU's Horizon 2020 IA ReconCell (GA no. 680431) and the the EU's Horizon 2020 RIA AUTOWARE (GA no. 723909).

REFERENCES

- [1] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57, 5 (2009), 469–483.
- [2] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. 2008. Robot programming by demonstration. In *Springer handbook of robotics*. Springer, 1371–1394.
- [3] Thomas Dietz, Ulrich Schneider, Marc Barho, Susanne Oberer-Treitz, Manuel Drust, Rebecca Hollmann, and Martin HÄdgle. 2012. Programming System for Efficient Use of Industrial Robots for Deburring in SME Environments. In *ROBOTIK 2012; 7th German Conference on Robotics* (2012). VDE, 1–6.
- [4] Rüdiger Dillmann. 2004. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems* 47, 2-3 (2004), 109–116.
- [5] M. Hersch, F. Guenter, S. Calinon, and A. Billard. 2008. Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations. *IEEE Transactions on Robotics* 24, 6 (2008), 1463–1467.
- [6] Yoram Koren, Uwe Heisel, Francesco Jovane, Tosliiichi Moriawaki, Guenter Pritschow, A. Galip Ulsoy, and Hendrik M.J. Van BrÄijssel. 1999. Reconfigurable Manufacturing Systems. *CIRP Annals - Manuf. Technol.* 48, 2 (1999), 527–540.
- [7] Alison Millar and Henrik Kihlman. 2009. *Reconfigurable flexible tooling for aerospace wing assembly*. Technical Report. SAE Technical Paper.
- [8] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. 2009. ROS: An open-source Robot Operating System. In *ICRA workshop on open source software*. Kobe, Japan, 5.
- [9] Barry Ridge, Timotej Gaspar, and Ales Ude. 2017. Rapid state machine assembly for modular robot control using meta-scripting, templating and code generation. In *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. Birmingham, UK, 661–668.
- [10] G. Zhang, R. Liu, L. Gong, and Q. Huang. 2006. An Analytical Comparison on Cost and Performance among DMS, AMS, FMS and RMS. In *Reconfigurable Manufacturing Systems and Transformable Factories*, Anatoli I. Dashchenko (Ed.). Springer Berlin Heidelberg, 659–673.

Challenges of Collaborative Mobile Manipulation for Industrial Applications

Mathias Brandstötter, Bernhard Dieber, Matteo Lucchi,
Stephan Mühlbacher-Karrer, and Horst Pichler

Institute for Robotics and Mechatronics, JOANNEUM RESEARCH, Austria

Modern manufacturing processes require high flexibility and human-machine collaboration to cope with increasing product variability, shorter production life cycles, and higher quality demands. Collaborative mobile manipulators (CMMs) meet these requirements. They combine the freedom of movement, higher payload, and speed of mobile robots with the universal applicability, positioning accuracy and repeatability of manipulator arms, equipped with collaborative features, like operator safety and sensitivity, in order to operate near and with humans. Despite recent innovations the industrial application of CMMs did not progress as expected. The main challenges for collaborative mobile manipulation can be found in the following areas:

Sensors and Perception Mobile collaborative applications require fast, robust, and highly accurate localization and detection of objects in the environment, ideally in a range between sub-millimeter to several meters. For safe human-robot interaction the accurate detection and localization of humans and estimation of applied forces for precise, safe and collision-free movement is paramount. As CMM movements are only limited by the environment they need a panorama view. This requires multiple sensors of different types, mounted to maximize the field of view while minimizing self-occlusion, multi-sensor calibration, and fast and reliable algorithms for data fusion and interpretation. Occlusion, due to CMM and operator movement, produces blind spots which decreases safety and productivity. Current research efforts aim towards better machine learning-based perception methods, novel multi-sensor modalities, and entirely new sensor systems; for instance capacitive surrounding-sensing "artificial skins" will result in more sensitive and safer manipulation and contactless interaction with humans.

Control One particular concern is to maintain the stability of the mobile platform while performing manipulation tasks with the arm. Inherent kinematic redundancy of the combined system is another problem. The degrees of freedom associated to the combination of a mobile platform with a manipulator are different in their dynamical and kinematic properties and have fundamentally different effects on the performance of the overall system. Redundancy resolution has a task of utilizing degrees of freedom in such a way that each subsystem is optimally exploited in terms of joint limit avoidance, stability, energy consumption, fault recovery, obstacle avoidance, and so on.

Task Scheduling and Allocation Collaborative applications require multi-agent planning with temporal and spatial

constraints to schedule and allocate tasks. The complexity of the combined optimization problem and the unforeseeable agent behavior impedes fast reactions that are required in dynamic situations. Furthermore, most planning techniques treat humans rather as obstacles, a source of uncertainty, than cooperating entities. Models and fast methods are required to capture uncertainties and skills for both human and robotic agents and to take them into consideration when tasks are allocated.

Human-machine Interface and Programming CMMs present a natural step toward flexible universal robot systems. Non-intuitive kinematic properties make conventional programming tedious and require some form of teaching. For complicated tasks which include simultaneous movement of the whole robot some form of semi-autonomous control has to be present and reprogramming is then reparametrization through interaction. Well-accepted contact-based human-machine interfaces are unsuitable for interaction with a moving system when the robot is either too far away or the position is changing. Even when use-cases are stationary, the geometrical properties of the system force interfaces to be placed on the platform within the manipulator workspace which is in many cases highly inconvenient.

Safety Currently no safety standard addresses mobile manipulation directly. Usually several more general ISO-standards are taken into account when designing robot application that are safe for humans, the environment, and the robot. Critical points during safety assessment are often safety-certified fail-safe hardware and software, and the risks of battery operation, locomotion in large environments, and autonomy of the device. Especially the uncertainty introduced by humans combined with robot autonomy poses a big problem, as robot safety usually tolerates only marginal uncertainties. This results in severe limitations, most notably speed and force reduction, which lessen the applicability and productivity of CMMS considerably.

Security As industrial settings become more and more digitized and connected, also the threat level in terms of cyber-attacks dramatically increases. From a systems engineering point of view, a mobile manipulator is a complex system of systems. In such systems, the risk of uncovered security issues increases along with the growth of complexity. Considering sensitive mobile manipulators, outside influence may lead to harm to humans if (software-defined) safety functions are exposed to attacks.

Resilience in mobile manipulation

Thomas Haspl, Bernhard Dieber, Benjamin Breiling*
firstname.lastname@joanneum.at

ABSTRACT

Safety and Security in robotics have long been known to go together hand in hand in order to make robots safe around humans. In modern, intelligent robots however, where software is a dominating part, the quality and reliability of software is a key issue.

To gain most from the increased potential of robots, adequate software architectures must be developed to handle their complexity. In this abstract, we sketch our ideas and work towards combining software architectures with robot security to work towards highly capable, secure robots.

KEYWORDS

software, security, mobile manipulation

1 RESILIENT SOFTWARE FOR COMPLEX ROBOTS

Security in robotics has gained some attention in the recent years. It has been shown that the most popular framework, ROS, has severe deficiencies in terms of security [1] resulting in easy-to-hack robots [2]. However, software engineering methods in robotics are still lacking the proper attention. We argue that for safety and security of robots, high-quality software is key. We present our work in software architectures and their security and hint towards later research directions.

1.1 Software architecture for mobile manipulation

In [3], we have shown an architecture for our CHIMERA mobile manipulator. This architecture separates the software into hardware, abstraction and application layers and defines clear interfaces between each. The driver layer can be exchanged to enable the reuse of business software on multiple robot platforms. Further, it defines a dedicated space where system integrators can enhance the core firmware with drivers and additional functions.

1.2 Security architecture for mobile manipulation

The architecture described above needs security measures integrated in order to protect the robot from outside attacks. Obviously, network and operating system security are required measures. However, we are convinced, that a multi-level approach is required, where multiple layers of security are implemented. Our secure architecture is shown in fig 1.

*All authors contributed equally to this research.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IS2019, October 7–11, 2019, Ljubljana, Slovenia, Europe

© 2019 Copyright held by the owner/author(s).

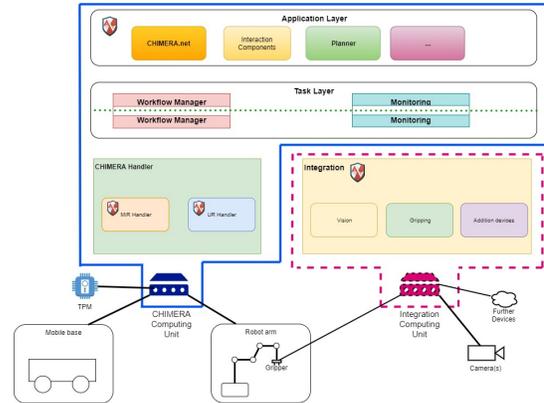


Figure 1: Software architecture for mobile manipulators.

In our approach to securing the software architecture, we heavily rely on isolation. We use two dedicated computing units where each one has different security levels. The CHIMERA computing unit contains the core business software and drivers for mobile base and arm. The Integration computing unit contains code and device drivers developed by system integrators. This separation ensures, that the integrator cannot compromise the security of the core system. In addition, individual layers of the architecture are isolated in separate docker containers with well-defined security boundaries.

2 RESEARCH DIRECTIONS

As part of this ongoing work, we want to establish Software as the third "S" of great robots besides Safety and Security. We see all three topics tightly integrated and required to make future robots productive companions in- and outside of industry. We will work on methods for better robot software that also enables developers to better test their software and easily employ security measures.

REFERENCES

- [1] Bernhard Dieber, Benjamin Breiling, Sebastian Taurer, Severin Kacianka, Stefan Rass, and Peter Schartner. 2017. Security for the Robot Operating System. *Robotics and Autonomous Systems* 98 (2017), 192–203.
- [2] Bernhard Dieber, Ruffin White, Sebastian Taurer, Benjamin Breiling, Gianluca Caiazza, Henrik Christensen, and Agostino Cortesi. 2020. *Penetration Testing ROS*. Springer International Publishing, Cham, 183–225. https://doi.org/10.1007/978-3-030-20190-6_8
- [3] Thomas Haspl, Benjamin Breiling, Bernhard Dieber, Marc Pichler, and Guido Breitenhuber. 2019. Flexible industrial mobile manipulation: a software perspective. In *Proceedings of the OAGM & ARW Joint Workshop 2019*. <https://doi.org/10.3217/978-3-85125-663-5-10>

Visual Feedback and Learning for Optimal Velocity of Robotic Visual Quality Inspection

Andrej Gams
Jozef Stefan Institute
Ljubljana, Slovenia
andrej.gams@ijs.si

Simon Reberšek
Jozef Stefan Institute
Ljubljana, Slovenia
simon.rebersek@ijs.si

Aleš Ude
Jozef Stefan Institute
Ljubljana, Slovenia
ales.ude@ijs.si

ABSTRACT

Robotic learning can effectively be applied for industrial applications. In this paper we show one such example, with a learning algorithm applied to reach the optimal velocity of robotic motion for visual quality inspection. If such learning is performed before the start of the production, even if it takes a lot of repetitions, it can achieve faster cycle times and thus greater productivity. The described approach is general and can be used with different types of learning and feedback signals. In the paper we analyze the appropriate feedback signal and show the results of learning for a standard area-scan camera.

KEYWORDS

robotic learning, visual feedback, focus measure, robotic quality control

1 INTRODUCTION

Many operations are performed with autonomous robots in factories, and many more are expected in the factories of the future. Often, visual feedback is used to provide the trajectory of the robot. [16]. However, various vision techniques, such as time of flight, structured light, laser triangulation, RGB cameras, stereo vision, etc. are used for quality control processes in the industry [4, 9]. Quality control can take different modes. For example, discrete checking of an object from a few viewpoints and comparing the acquired images to predefined templates [11]. Another option is to continuously acquire images with either moving the in-hand camera, or moving the object in front of the camera. A plethora of advanced methods for image processing for quality inspection have been proposed, including deep learning methods [17].

For effective vision-based operations, the machine vision hardware needs to be properly set-up and tuned. In large-scale automated production, it is typically set-up once, and then it remains in the same configuration throughout its life cycle. Consequently, machine vision hardware is often designed in a way that some adjustments can only be carried out manually. Many lenses thus have a fixed focal length and manual adjustment of the iris and focus [1]. However, even if the vision-hardware is set up only once, the process still constitutes a tedious and demanding task. For example, in continuous visual inspection, e. g., for visual inspection of weld



Figure 1: The robot cell composed of a UR-10 robot, a Basler acA1300-60gm area scan camera, a dedicated light source (not shown) and the dummy flat object at a calibrated distance from the robot.

seams [18], requires the robot to follow the seam with the camera at the end-effector. The image has to be sharp in all the positions and at all velocities. Thus, for such continuous visual quality control, the operator has to define the correct robot path, but also the correct speed, because too fast motion in front of the camera might result in a blurry image.

The demands of the industry typically culminate in having to move as fast as possible in order to reach high cycle times. [19]. Thus, when programming robot motion for quality control, the path can be properly configured by exporting the object CAD data and appropriate robot-to-object calibration, but the speed of robot motion is typically left to the operator, who spends a considerable amount of time hand-tuning it. However, this tuning could be left to an autonomous learning algorithm with proper feedback. In this paper we briefly analyze possible visual filters for appropriate feedback, and demonstrate how hand-tuning can be automated by employing learning algorithms.

1.1 Problem Statement

We investigate learning of motion speed for continuous visual quality inspection of products with a robot using an in-hand camera. The system should:

- follow a predefined path,

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IS2019, October 7–11, 2019, Ljubljana, Slovenia, Europe
© 2019 Copyright held by the owner/author(s).

- autonomously optimize motion velocity using learning, so that
- the velocity of motion does not introduce blurring, i. e., a reduced focus measure.

We also analyze appropriate visual feedback filters to determine image sharpness.

The following assumptions hold. i) A CAD system provides an accurate robot path (trajectory) consisting of required positions and orientations; ii) proper robot-object calibration can be achieved; iii) The system operates under constant lighting and camera conditions.

To achieve seamless velocity modulation, we applied Dynamic movement primitives (DMPs) developed by Ijspeert et al. [10]. We used a variant of DMPs called Cartesian Space Dynamic Movement Primitives [21] for the trajectory encoding. Other trajectory encoding approaches could easily be applied, for example Gaussian Mixture Models [3]. For the learning we applied Iterative Learning Control (ILC) [2, 6]. Again, other methods, such as reinforcement learning [5, 13] could be applied.

2 FOCUS MEASURE

Visual quality inspection requires sharp, focused images. Only a few industrial camera/lenses on the market provide autofocus, with little information about how focus is determined in these cameras [1].

We first used robot-driven autofocus as described in [1] to set our fixed-focus camera at the right distance from the object for inspection. To do this we used squared horizontal gradient focus measure, as suggested by [1]. This focus measure has a distinct bell-shape characteristics, with the best focus achieved at the peak. The robot moves the camera perpendicularly to the object of inspection, away and towards the object. After detecting the peak value (the focus measure begins to decrease), the robot reverses its motion and travels in the other direction at a slower speed, again until crossing the peak value. These movements are repeated until the accurate position resulting in peak focus measure ϕ is obtained. Details of this method and results showing that the achieved focus measure is higher than the one achieved by manually positioning the camera, are presented in [1].

Using this approach we can set the camera into focus for one point, for example above the starting point of the path of inspection. We assume that the desired inspection path has been extracted from a CAD model of the inspected object. To obtain the reference values $\phi(t)$ for all points on the inspection trajectory, the robot moves along the desired inspection path. However, the question is whether speed has an effect on the focus measure, and furthermore, out of many focus measures that exist, which will be most effected by the speed.

Focus measures are based on different orders of differentiation (first or second), image histogram, correlation and data compression [14]. Methods employing first-order gradients use different operators, such as squared gradient, Sobel (horizontal, vertical, combined), Laplacian, Scharr, and others. We tested several possible focus measures. We moved the robot with an in-hand camera over a dummy object at three different velocities, completing the motion in 3s, 20s, and 60s. Figure 2 shows the relative focus measure as a function of normalized time (phase), going from 0 to 1, for different

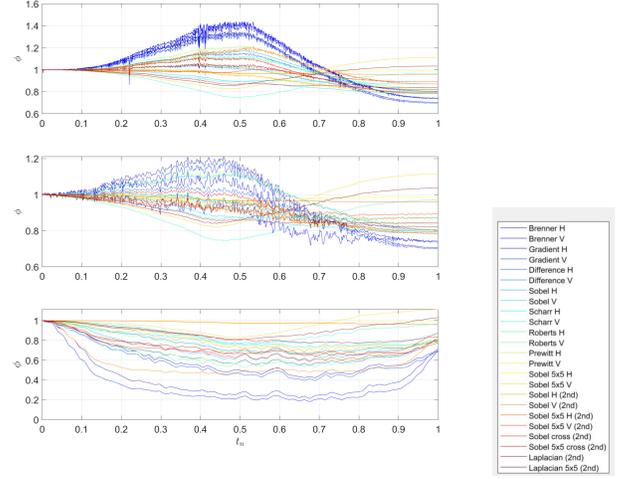


Figure 2: Different focus measures at different speeds of robot motion (top: 60s, middle: 20s, bottom: 3s), for normalized time. The measures were normalized to the initial value.

measures. The label states the measures used. The feedback focus measures were normalized to the initial value. As we can see, a higher velocity indeed decreases the focus measure, and the effect is different for different focus measures.

Figure 3 shows filtered values of relative difference between slow and fast motion for the top 10 focus measures. We can see that squared horizontal gradient focus measure is the most reactive to change of velocity. Is is provided by

$$\phi = \sum_{x=0}^{M-1} \sum_{y=0}^{N-2} (I(x, y+1) - I(x, y))^2. \quad (1)$$

Here the image is sized $M \times N$, with $I(x, y)$ the intensity values at pixels (x, y) .

The values would be the same for the vertical gradient if the camera were rotated 90° . The vertical horizontal gradient is calculated by

$$\phi = \sum_{x=0}^{M-2} \sum_{y=0}^{N-1} (I(x+1, y) - I(x, y))^2. \quad (2)$$

Brenner vertical and horizontal filters provide similar values. They are defined by

$$\phi = \sum_{x=0}^{M-1} \sum_{y=0}^{N-3} (I(x, y+2) - I(x, y))^2. \quad (3)$$

for the horizontal and

$$\phi = \sum_{x=0}^{M-3} \sum_{y=0}^{N-1} (I(x+2, y) - I(x, y))^2. \quad (4)$$

for the vertical filter.

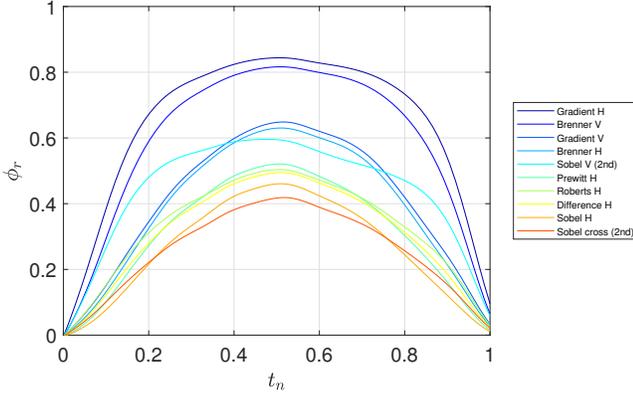


Figure 3: Filtered relative change of feedback values for 10 different focus measures.

3 TRAJECTORY ENCODING

In this paper we used the original formulation of Cartesian DMPs from [21], expanded with temporal scaling, as originally proposed for standard DMPs in [15].

The following parameters compose a CDMP: weights \mathbf{w}_k^p , $\mathbf{w}_k^o \in \mathbb{R}^3$, $k = 1, \dots, N$, which represent the position and orientation parts of the trajectory, respectively; trajectory duration τ and the final desired, goal position \mathbf{g}^p and orientation \mathbf{g}^o of the robot. Variable N sets the number of radial basis functions that are used to encode the trajectory. The orientation is in CDMP represented by a unit quaternion. In this paper we only consider the positions.

$$v(s)\tau\dot{\mathbf{z}} = \alpha_z(\beta_z(\mathbf{g}^p - \mathbf{p}) - \mathbf{z}) + \mathbf{f}_p(s), \quad (5)$$

$$v(s)\tau\dot{\mathbf{p}} = \mathbf{z}, \quad (6)$$

Variable $v(s)$, as a function of the phase, provides temporal scaling. Parameter \mathbf{z} , denotes the scaled linear velocity ($\mathbf{z} = \tau\dot{\mathbf{p}}$). The nonlinear parts, termed also forcing terms, \mathbf{f}_p and are defined as

$$\mathbf{f}_p(s) = \mathbf{D}_p \frac{\sum_{k=1}^N \mathbf{w}_k^p \Psi_k(s)}{\sum_{k=1}^N \Psi_k(s)}, \quad (7)$$

Forcing terms contain parameters $\mathbf{w}_k^p \in \mathbb{R}^3$. They have to be learned, for example directly from an input Cartesian trajectory $\{\mathbf{p}_j, \dot{\mathbf{p}}_j, \ddot{\mathbf{p}}_j, t_j\}_{j=1}^T$. The scaling matrix $\mathbf{D}_p \in \mathbb{R}^{3 \times 3}$ can be set to $\mathbf{D}_p = \mathbf{I}$. Other possibilities are described in [21]. The nonlinear forcing terms are defined as a linear combination of radial basis functions Ψ_k

$$\Psi_k(x) = \exp\left(-h_k(x - c_k)^2\right). \quad (8)$$

Here c_k are the centers and h_k the widths of the radial basis functions. The distribution of weights can be, as in [20], $c_k = \exp\left(-\alpha_x \frac{k-1}{N-1}\right)$, $h_k = \frac{1}{(c_{k+1} - c_k)^2}$, $h_N = h_{N-1}$, $k = 1, \dots, N$. The time constant τ is set to the desired duration of the trajectory, i. e. $\tau = t_T - t_1$. The goal position is usually set to the final position on the desired trajectory, i. e. $\mathbf{g}^p = \mathbf{p}_{t_T}$. Detailed CDMP description and auxiliary math are explained in [21].

Temporal scaling $v(s)$ provides a trajectory that defines a speed profile of the motion. It is composed of a weighted combination of kernel functions

$$v(s) = \frac{\sum_{k=1}^R \mathbf{w}_k^v \Psi_k(s)}{\sum_{k=1}^R \Psi_k(s)}. \quad (9)$$

Here R defines the number of kernel functions, given in (8), for temporal scaling. For simplicity, this number can be the same as N in (7). The weights \mathbf{w}_k^v need to be learned in the same manner as the weights for position trajectories.

4 IMPROVING SPEED OF QUALITY CONTROL WITH LEARNING

Focus measure is repeatable, and there is a clear difference in ϕ for different motion speeds, as evident from Fig. 2. Therefore, we can use ϕ as the feedback for learning.

The goal of learning here is to achieve a fastest possible velocity profile, where there will be only little or even no degradation of the focus measure. Thus, the motion will be executed as fast as possible, and the sharpness of the image, used for quality inspection, will not degrade.

It should be noted that with the chosen parametric speed profile representation, different means of learning open up, as was shown in [5], or in [12]. In this paper we have chosen one of the variations of iterative learning control. The advantage of using a learning control method is that it requires very few iterations to improve results. However, such methods never truly converge, but only asymptotically approach the target value [2].

The chosen learning algorithm for learning was previously applied for coaching of robot motion through human intervention [7]. A short recap is provided for completeness of the paper. Its basis is learning of weights of CDMPs, but in this case it is used for the learning of the weights of the velocity profile v . The weights of the velocity profile \mathbf{w}^v are iteratively updated (for 1DOF) with

$$\mathbf{w}_{i,j+1}^v = \mathbf{w}_{i,j}^v + \Gamma_{i,j+1} P_{i,j+1} r e_j \quad (10)$$

$$P_{i,j+1} = \frac{1}{\lambda} \left(P_{i,j} - \frac{P_{i,j}^2 r^2}{\frac{\lambda}{\Gamma_i} + P_{i,j} r^2} \right) \quad (11)$$

$$e_j = f_{\text{target},j} - \mathbf{w}_{i,j}^v r. \quad (12)$$

Here $j+1$ stands for the next time sample and i for the selected weight. P_i , is the inverse covariance of w_i , r is the amplitude gain. To apply this algorithm for modifying the speed profile based on the focus measure ϕ , we replace (12) with

$$e_j = k * (\phi_{\text{slow}} - \phi_{\text{fast}}). \quad (13)$$

here k is a positive constant gain. The whole algorithm is described in procedure of Fig. 4. The learning takes place until a predefined threshold of e_j is reached. This threshold can be determined empirically.

Instead of learning directly on the weights, one can also simply generate the velocity profile from the weights and add to it a scaled e_j ,

$$v_{l+1}(t) = v_l(t) + k e_j(t), \quad (14)$$

where the gain k is set empirically and l stands for iteration. The resulting $v_{l+1}(t)$ is then again encoded into weights, for example

procedure LearnProfile

```

record  $\phi$  for slow (practically static) motion;
record  $\phi$  for fast motion with  $w_i^v = \text{const}$ ;
while  $\phi_{\text{latest}} > \text{threshold}$ 
  execute motion with current  $w^v$ 
  calculate new error of  $\phi$  with  $\phi - \phi_{\text{latest}}$ 
  update  $w^v$  using (10), (11) and (13)
end

```

Figure 4: Procedure for learning the velocity profile using the squared gradient focus measure.

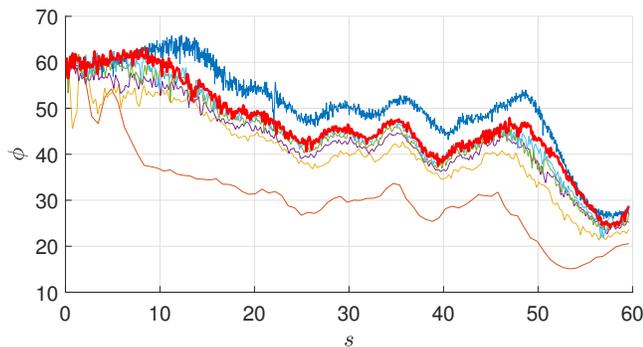


Figure 5: Results of velocity learning for a dummy flat object. The top lines shows absolute ϕ for slow 60s motion. The bottom line shows ϕ for fast, 3s motion. ϕ over iterations is shown between, with the final, red line almost the reference, but at 19.12s.

iteratively using (10) – (12), or with a batch conversion, as shown in [10].

Figure 5 shows the results of the algorithm, applied on a dummy object, using the algorithm described in Fig. 4 and squared horizontal gradient focus measure. Results on a curved object were reported in [8].

5 CONCLUSION

Learning algorithms have tremendous potential to improve the productivity of industrial processes today, not only in the future. The results show that autonomous learning algorithms can improve the performance of the robot, and that such algorithms can be effectively applied optimizing production processes. Thus, they can relieve and help operators/engineers. Fine-tuning and calibration of the processes is a tedious, long process, requiring a lot of effort. Time and money can be saved both in the set-up as well as in the improved productivity.

ACKNOWLEDGMENTS

This research has been funded in part by the GOSTOP programme C3330-16-529000, co-financed by Slovenia and EU under ERDF, and by the EU’s Horizon 2020 IA QU4LITY (GA no. 825030).

REFERENCES

- [1] Robert Bevec, Timotej Gašpar, and Aleš Ude. 2019. Robot-Driven Autofocus Control Mechanism for an In-hand Fixed Focus Camera. In *Advances in Service and Industrial Robotics*, Nikos A. Aspragathos, Panagiotis N. Koustoumparis, and Vassilis C. Moulaniotis (Eds.). Springer International Publishing, Cham, 551–559.
- [2] D. A. Bristol, M. Tharayil, and A. G. Alleyne. 2006. A survey of iterative learning control. *IEEE Ctrl. Sys. M.* 26, 3 (2006), 96–114. <https://doi.org/10.1109/MCS.2006.1636313>
- [3] S. Calinon. 2015. Robot learning with task-parameterized generative models. In *Proc. Intl Symp. on Robotics Research (ISRR)*.
- [4] Che-Seung Cho, Byeong-Mook Chung, and Moo-Jin Park. 2005. Development of real-time vision-based fabric inspection system. *IEEE Transactions on Industrial Electronics* 52, 4 (Aug 2005), 1073–1079. <https://doi.org/10.1109/TIE.2005.851648>
- [5] M. P. Deisenroth, G. Neumann, and J. Peters. 2013. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics* 2, 1-2 (2013), 1–142.
- [6] A. Gams, B. Nemeč, A. J. Ijspeert, and A. Ude. 2014. Coupling Movement Primitives: Interaction With the Environment and Bimanual Tasks. *IEEE Transactions on Robotics* 30, 4 (Aug 2014), 816–830. <https://doi.org/10.1109/TRO.2014.2304775>
- [7] Andrej Gams, Tadej Petrič, Martin Do, Bojan Nemeč, Jun Morimoto, Tamim Asfour, and Aleš Ude. 2016. Adaptation and coaching of periodic motion primitives through physical and visual interaction. *Robotics and Autonomous Systems* 75 (2016), 340 – 351. <https://doi.org/10.1016/j.robot.2015.09.011>
- [8] A. Gams, S. Reberšek, B. Nemeč, J. Škrabar, R. Krhlikar, J. Skvarč, and A. Ude. 2019. Robotic Learning for Increased Productivity: Autonomously Improving Speed of Robotic Visual Quality Inspection. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. 1275–1281. <https://doi.org/10.1109/COASE.2019.8842851>
- [9] Niko Herakovic. 2010. Robot Vision in Industrial Assembly and Quality Control Processes. In *Robot Vision*. IntechOpen, Rijeka, Chapter 26. <https://doi.org/10.5772/9285>
- [10] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal. 2013. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Computation* 25, 2 (2013), 328–373.
- [11] Tetyana Ivanovska, Simon Reich, Robert Bevec, Ziga Gosar, Minija Tamosiunaite, Ales Ude, and Florentin Wörgötter. 2018. Visual Inspection and Error Detection in a Reconfigurable Robot Workcell: An Automotive Light Assembly Example. In *VISIGRAPP*.
- [12] Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274. <https://doi.org/10.1177/0278364913495721>
- [13] J. Kober and J. Peters. 2011. Policy Search for Motor Primitives in Robotics. *Machine Learning (MLJ)* 1-2 (2011), 171–203.
- [14] H. Mir, P. Xu, and P. van Beek. 2014. An extensive empirical evaluation of focus measures for digital photography. In *Digital Photography X (procspie)*, Vol. 9023. 90230I. <https://doi.org/10.1117/12.2042350>
- [15] B. Nemeč, A. Gams, and A. Ude. 2013. Velocity adaptation for self-improvement of skills learned from user demonstrations. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. 423–428. <https://doi.org/10.1109/HUMANOIDS.2013.7030009>
- [16] Luis Pérez, Íñigo Rodríguez, Nuria Rodríguez, Rubén Usamentiaga, and Daniel F. García. 2016. Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. *Sensors* 16 3 (2016).
- [17] D. Racki, D. Tomazevic, and D. Skocaj. 2018. A Compact Convolutional Neural Network for Textured Surface Anomaly Detection. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 1331–1339. <https://doi.org/10.1109/WACV.2018.00150>
- [18] D. Schreiber, L. Cambrini, J. Biber, and B. Sardy. 2009. Online visual quality inspection for weld seams. *The International Journal of Advanced Manufacturing Technology* 42, 5 (01 May 2009), 497–504. <https://doi.org/10.1007/s00170-008-1605-3>
- [19] O. Semeniuta, S. Dransfeld, and P. Falkman. 2016. Vision-based robotic system for picking and inspection of small automotive components. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. 549–554. <https://doi.org/10.1109/COASE.2016.7743452>
- [20] A. Ude, A. Gams, T. Asfour, and J. Morimoto. 2010. Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives. *IEEE Transactions on Robotics* 26, 5 (Oct 2010), 800–815. <https://doi.org/10.1109/TRO.2010.2065430>
- [21] A. Ude, B. Nemeč, T. Petrič, and J. Morimoto. 2014. Orientation in Cartesian space dynamic movement primitives. In *IEEE Int. Conference on Robotics and Automation (ICRA)*. 2997–3004. <https://doi.org/10.1109/ICRA.2014.6907291>

Intuitive Hand-Guidance of a Mobile Manipulator

Matthias Weyrer
Joanneum Research ROBOTICS
Lakeside B08a
9020 Klagenfurt am Wörthersee, Austria
matthias.weyrer@joanneum.at

ABSTRACT

Mobile manipulators are robot systems which enable the capabilities of logistics and manipulation tasks. Thus, they potentially close unconsidered gaps regarding flexibility in modern production lines. We address the problem of developing an easy-to-use interface for intuitive robot programming. This interface implements a whole-body compliance control to allow for hand-guidance.

1. INTRODUCTION

Producing small lot-sizes or highly customized products require enhanced flexibility within the manufacturing processes. This raises the need for flexible and easily adaptable robotic systems. While conventional automated production lines are usually prepared and programmed by external experts, modern applications require frequent adaption or reprogramming. To enable this directly for workers, without explicit programming skills but high domain knowledge, an intuitive interface is needed. One well-known technique is kinesthetic programming by demonstration, where a compliant robot can be hand-guided into desired configurations. While the compliance control for serial manipulators has been well investigated, the whole-body compliance for a mobile manipulator, consisting of a serial manipulator on top of a mobile base, has gained little attention yet. Leboutet et al. [1] presented a strategy with hierarchical force propagation for a mobile manipulator with omni-directional base. Navarro et al. [2] proposed a system where the motion distribution between the serial manipulator and the mobile base is done with optimization.

2. METHOD

In our previous work [4] we presented a control strategy for whole-body compliance of a mobile manipulator with differential drive. A force/torque sensor is mounted close to the end-effector (EE) to measure the external wrench applied by the user. The robot shows kinematic redundancies regarding the 3D task space since the 6 degrees of freedom of the arm are supplemented with those of the mobile base. Our control structure focuses on resolving these redundancies by implementing three different modes: A *pull-mode*, where the mobile base can be pulled like a steered trailer, which means that the base is rotating and translating towards the EE and haptic feedback is given to the user by means of a virtual spring. In the *ur-mode* only the serial arm moves, and the *push-mode* allows for pushing the mobile base while receiving haptic feedback of a virtual spring. The decision, which mode is used depends on the actual

position of the EE. Two circles in the xy -plane define two borders of cylindrical shapes in the 3D space. If the EE leaves the outer circle *pull-mode* is active, in between the two circles the *ur-mode* is active and inside the inner circle *push-mode* becomes active.

The proposed control structure was successfully validated throughout laboratory experiments, but approaching arm configurations close to singularities proved to be problematic. Since the suggested controller uses end-effector velocities as control inputs, close to a singular configuration, a rather slow end-effector motion may lead to very high joint velocities causing possibly dangerous situations. In [3] we extended our controller to avoid approaching singular arm configurations by providing haptic feedback to the user. We did a detailed analysis of all possible singularities of the UR10 and implemented virtual springs to avoid them.

For future work it is planned to integrate haptic feedback to avoid self collisions. Furthermore, depending on the choice of the radius of the inner circle, the workspace of the serial arm is restricted, since when the inner circle is entered by the EE, *push-mode* is active and a virtual spring will move the EE back outside of the inner circle. We plan to refine the strategy at this point to minimize the volume of the restricted workspace.

3. REFERENCES

- [1] Q. Leboutet, E. Dean-León, and G. Cheng. Tactile-based compliance with hierarchical force propagation for omnidirectional mobile manipulators. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 926–931. IEEE, 2016.
- [2] B. Navarro, A. Cherubini, A. Fonte, G. Poisson, and P. Fraisse. A framework for intuitive collaboration with a mobile manipulator. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6293–6298. IEEE, 2017.
- [3] M. Weyrer, M. Brandstötter, and M. Husty. Singularity avoidance control of a non-holonomic mobile manipulator for intuitive hand guidance. *Robotics*, 8(1):14, 2019.
- [4] M. Weyrer, M. Brandstötter, and D. Mirkovic. Intuitive hand guidance of a force-controlled sensitive mobile manipulator. In *IFTToMM Symposium on Mechanism Design for Robotics*, pages 361–368. Springer, 2018.

Learning Robotic Handwriting with Convolutional Image-to-Motion Encoder-Decoder Networks

Barry Ridge

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

Rok Pahič

Dept. of Automatics, Biocybernetics, and Robotics, Jožef
Stefan Institute
Ljubljana, Slovenia
rok.pahic@ijs.si

ABSTRACT

Learning to recognize and reproduce handwriting is a familiar skill taught to every educated human being but is challenging to teach to a robot given its tight coupling between perception and motion. In this work, we address the specific task of recognizing digits in single images and reproducing the digits in the form of robot end-effector trajectories encoded as dynamic movement primitives (DMPs) used to control the pen strokes. Here we present a convolutional image-to-motion encoder-decoder deep neural network architecture that takes the raw digit images as input and produces the DMP parameters as output, learning a mapping between the two as a latent representation. The architecture is tested on several challenging noisy digit datasets under different training regimes and compared to an architecture without convolutional layers in the image encoder where it is shown to provide robust results for the digit writing task.

KEYWORDS

deep neural networks, dynamic movement primitives

1 INTRODUCTION

Effectively learning to predict action mappings directly from perceptual input is a highly challenging problem in robotics research that has seen a broad variety of approaches attempting to solve it in different settings. The particular setting under consideration in this work is depicted in Fig. 1, in which a robot must learn direct mappings between handwritten characters in input images and the motion trajectories needed to draw them. In previous work we proposed a fully-connected encoder-decoder network architecture [6] that used dynamic movement primitives (DMPs) [5] for movement representation and this proved to be an effective choice both for representation and learning with the neural network and ultimately for control of the robot when drawing the actual digits. The fully-connected architecture, however, was not ideal for image representation.

Here we investigate a different architecture that combines the benefits of convolutional layers for image encoding with those

of a fully-connected encoder-decoder architecture for DMP parameter prediction and image-to-motion representation in a low-dimensional latent space. This combination allows for relatively robust prediction compared to the previously proposed architecture, even when the input images are heavily corrupted by noise. The use of convolutional layers has the added benefit of significantly reducing the number of network parameters and by pre-training these layers on images from a similar image domain, the learning process is further improved.



Figure 1: Writing digits with a robot using image-to-motion encoder-decoder network prediction.

Autoencoders [3], as well as variational autoencoders [4], have been demonstrated to be quite effective when it comes to calculating DMP-based representations of human motion. Since our focus is on learning direct mappings between images and actions, instead of using such autoencoder networks in which the DMP encoding occurs in the latent space, we use an encoder-decoder architecture in which the image is encoded from the input layer, the DMP parameters are predicted at the output layer and the transformation and generalization of the image-to-motion representation occurs in the low-dimensional latent space. Encoder-decoder networks in combination with convolutional layers have proven to be useful in computer vision. A well-known example is SegNet [1], in which pre-trained convolutional layers from a convolutional neural network (CNN) were adapted to form a fully-convolutional encoder-decoder architecture for semantic pixel-wise segmentation.

Usually when CNNs are used for supervised learning of perception-action couplings, they are used in combination with another neural network in two separately trainable parts. In [9], Yang et al. first used a deep convolutional autoencoder for finding camera image features and then in combination with recorded robot angles, formed sequences for the learning task dynamics with a time delay neural network. Pervez et al. [7] used a pre-trained CNN for finding task parameters from input images, while using another fully-connected neural network to learn to generate forcing terms

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IS2019, October 7–11, 2019, Ljubljana, Slovenia, Europe

© 2019 Copyright held by the owner/author(s).

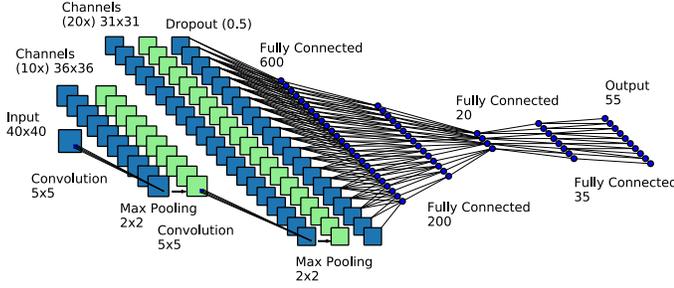


Figure 2: The CIMEDNet architecture.

from the clock signal and task parameters, before combining both networks in an end-to-end training scheme. Both of these two examples produce the next step from the image of the current step while working in online loop, whereas our method, by contrast, uses just single images for generating entire trajectories.

2 CONVOLUTIONAL IMAGE-TO-MOTION ENCODER-DECODER NETWORKS

The structure of the data under consideration in this work is the same as in [6] where the input and output data pairs take the form $\mathbf{D} = \{C_j, \mathbf{M}_j\}_{j=1}^M$ where M is the number of input and output training pairs, $C_j \in \mathbb{R}^{H \times W}$ are the input images of width W and height H , and \mathbf{M}_j the corresponding movements associated with each image, i. e. $\mathbf{M}_j = \{y_{i,j}, t_{i,j}\}_{i=1}^T$. Here $y_{i,j} \in \mathbb{R}^d$ are the vectors describing the movement's degrees of freedom, e. g. Cartesian positions or joint angles, $t_{i,j} \in \mathbb{R}$ the measurement times for the j -th movement, and d is the number of degrees of freedom. However, it should be noted that in this paper, we convert the movements \mathbf{M}_j to DMPs and construct all of the datasets used to train the network models as follows: $\mathbf{D}' = \{C_j, \mathbf{k}_j\}_{j=1}^M$, where \mathbf{k}_j are the DMP parameters calculated for each movement \mathbf{M}_j and are represented as

$$\mathbf{k}_j = \left\{ \{w_k\}_{k=1}^N, \tau, g, y_0 \right\}. \quad (1)$$

The construction of DMPs and the nature of the parameters $\{w_k\}_{k=1}^N$, τ , g and y_0 are explained in detail in the following subsection.

2.1 Motion Representation with DMPs

Letting a time-dependent movement trajectory be denoted as $y(t) \in \mathbb{R}^d$, a DMP specifying this trajectory is given by the following system of differential equations

$$\tau \dot{z} = \alpha_z (\beta_z (g - y) - z) + \text{diag}(g - y_0) \mathbf{F}(x), \quad (2)$$

$$\tau \dot{y} = z, \quad (3)$$

where $y_0 \in \mathbb{R}^d$ is the initial position on the trajectory, $g \in \mathbb{R}^d$ the final position on the trajectory, $\text{diag}(g - y_0) \in \mathbb{R}^{d \times d}$ a diagonal matrix with components of vector $g - y_0$ on the diagonal, $\mathbf{F}(x) \in \mathbb{R}^d$ a nonlinear forcing term, $z \in \mathbb{R}^d$ a scaled velocity of motion, and $x \in \mathbb{R}$ the phase defined by the following equation

$$\tau \dot{x} = -\alpha_x x. \quad (4)$$

The phase x is used instead of time to avoid explicit time dependency. It is fully defined by setting its initial value to $x(0) = 1$. Eq. system (2) – (4) constitutes a *dynamic movement primitive* (DMP).

If the parameters $\tau, \alpha_x, \alpha_z, \beta_z \in \mathbb{R}$ are defined appropriately, e. g. $\tau, \alpha_x > 0$ and $\alpha_z = 4\beta_z > 0$, then the linear part of equation system (2) – (3) becomes critically damped and y, z monotonically converge to a unique attractor point at $y = g, z = 0$. The forcing term $\mathbf{F}(x)$ is usually defined as a linear combination of radial basis functions

$$\mathbf{F}(x) = \frac{\sum_{k=1}^N w_k \Psi_k(x)}{\sum_{k=1}^N \Psi_k(x)} x, \quad (5)$$

$$\Psi_k(x) = \exp\left(-h_k (x - c_k)^2\right), \quad (6)$$

where c_k are the centers of Gaussians distributed along the phase of the trajectory, and h_k their widths. The role of \mathbf{F} is to adapt the dynamics of (2) – (3) to the desired trajectory $y(t)$, thus enabling the system to reproduce any smooth movement from the initial position y_0 to the final configuration g . This can be accomplished by computing the free parameters $w_k \in \mathbb{R}^d$ using regression techniques. See [8] for more details.

α_z, β_z , and α_x are usually constants that do not change between movements. Thus the neural network needs to learn the other parameters of differential equation system (2) – (4) to fully specify a DMP as defined in Equation (1).

2.2 Network Architecture

In our improved architecture, images are encoded via convolutional layers that are pre-trained as part of a basic CNN classifier that was trained on the original MNIST dataset. The input is a $40 \times 40 \times 1$ grayscale pixel image, followed by a convolutional layer with 5×5 kernel size and 10 feature maps, a convolutional layer with 5×5 kernel size and 20 feature maps, a 0.5 dropout layer, a fully-connected layer of size 320, a fully-connected layer of size 50 and the output layer of size 10 matching the number of digits. After training the classifier, the fully-connected layers are removed and the convolutional layers are retained and are used to form the first layers of the encoder in our proposed architecture. These two convolutional layers are followed by two added fully-connected layers with sizes of 600 neurons and 200 neurons, illustrated on the left side of Fig. 2.

Following the bottleneck of the network that forms the latent space representation, a decoder is formed via a number of fully-connected layers that gradually expand the number of units in each layer until the final output layer which has a size set to 55 units in order to match the DMP parameters $\{w_k\}_{k=1}^N, \tau, g$ and y_0 . The layers of the decoder are illustrated on the right side of 2 starting with the bottleneck of size 20, followed by a layer of size 35 and finishing with the output layer. This is the same decoder structure as used [6] and we retain it here as-is, having found it to be effective throughout our experiments for this particular use case. The cost function used to evaluate the output of the network is the same as that of Equation (9) in [6], which is defined for the j -th DMP as follows:

$$E_p(j) = \frac{1}{2} \left(\sum_{k=1}^N \|w_k - w_{k,j}\|^2 + (\tau - \tau_j)^2 + \|g - g_j\|^2 + \|y_0 - y_{0,j}\|^2 \right), \quad (7)$$

where $\{\{w_k\}_{k=1}^N, \tau, g, y_0\}$ denotes the output of the neural network and $\{\{w_{k,j}\}_{k=1}^N, \tau_j, g_j, y_{0,j}\}$ the DMP parameters from the training data $\mathbf{k}_j \in \mathbf{D}'$. For further details on the gradient calculations required for minimizing the cost function via backpropagation we refer the reader to [6].

3 EXPERIMENTS

In our experiments, we trained both the fully-connected image-to-motion encoder-decoder architecture (IMEDNet) and the convolutional architecture (CIMEDNet) on various digit image and motion trajectory datasets. The IMEDNet architecture was the same as described in [6] with fully-connected hidden layer sizes of 1500, 1300, 1000, 600, 200, 20, and 35 neurons, respectively. The CIMEDNet architecture was as described in Section 2.2 and as illustrated in Fig. 2.

In the case of CIMEDNet, we also experimented with either freezing the convolutional layer weights or training the entire network end-to-end. The results for these different training regimes are cataloged in Table 1.

3.1 Datasets

In order to construct \mathbf{D} , we employed the same scheme described in [6] to generate 40×40 images of synthetically written digits and associated two-dimensional artificial writing trajectory movements. Briefly, the synthetic trajectory data was generated using a combination of straight lines and elliptic arcs. These geometric elements were used to generate grayscale digit images and their parameters were varied according to a uniform distribution. The resulting images were processed with a Gaussian filter and some moderate salt-and-pepper noise was added to the foreground pixels. Finally, both the generated trajectories and the resulting images were transformed using affine transformations composed of translation, rotation, scaling, and shearing. These parameters were again taken from a uniform distribution. For the DMP representation of the trajectories, 25 radial-basis functions were selected for every dimension. The weights of these basis functions form together with the common time constant (1 parameter) and the start and the goal values of a planar movement (2×2 parameters), the full set of 55 DMP parameters that represent the motion. Using this procedure, several datasets were generated both with and without similar noise as used in the noisy MNIST (n-MNIST) datasets [2] as follows:

- **s-MNIST**: 2000 pairs of images and trajectories without any added noise were generated for each digit, for a total of 20000 samples that were split in a 70%/15%/15% ratio between training/validation/test data,
- **s-MNIST-AWGN-19.0**: 300 samples per digit/3000 total samples, using additive white gaussian noise with a signal-to-noise ratio of 19.0,
- **s-MNIST-AWGN-9.5**: 300 samples per digit/3000 total samples, using additive white gaussian noise with a signal-to-noise ratio of 9.5,
- **s-MNIST-MB**: 300 samples per digit/3000 total samples, using a motion blur filter emulating a linear motion of the camera of 5 pixels and a 15 degree motion in the counter-clockwise direction,

- **s-MNIST-RC-AWGN**: 300 samples per digit/3000 total samples, using a contrast range scaled down to half as well as additive white gaussian noise with a signal-to-noise ratio of 9.5.

It should be emphasized that in the results that follow, only the s-MNIST dataset was used for training the presented models.

3.2 Results

The main quantitative results are presented in Table 1 while qualitative results for selected samples are presented in Fig. 3. After training on the noiseless s-MNIST dataset each of the models were tested on all five of the noiseless and noisy s-MNIST datasets described in the previous section. The CIMEDNet architecture was trained with two separate training regimes in which the convolutional layer weights were frozen and the models were trained end-to-end respectively. For the quantitative evaluation, dynamic time warping was used to measure the mean pointwise pixel distance between the trajectories generated by the DMPs predicted by the networks from the digit images and the actual digit trajectories.

Table 1: DMP reconstruction statistics. The results are in pixels. The best result for each dataset is highlighted in bold-face.

	IMEDNet (End-to-End)	CIMEDNet (Frozen Conv.)	CIMEDNet (End-to-End)
s-MNIST	0.22 ± 0.08	0.26 ± 0.10	0.19 ± 0.08
s-MNIST-AWGN-19.0	0.56 ± 0.20	0.54 ± 0.20	0.36 ± 0.14
s-MNIST-AWGN-9.5	1.66 ± 0.60	1.48 ± 0.55	1.02 ± 0.45
s-MNIST-MB	0.35 ± 0.15	0.47 ± 0.25	0.36 ± 0.12
s-MNIST-RC-AWGN	2.32 ± 0.77	2.19 ± 0.76	1.93 ± 0.66

As can be seen in Table 1, the CIMEDNet model that is trained end-to-end significantly out-performs the IMEDNet model on both the noiseless s-MNIST dataset and on most of the noisy s-MNIST datasets, apart from the dataset featuring motion blur noise. We reason that this may be due to the fact that motion blur can significantly distort overall object shape and edge profiles and given that convolutional neural networks function the basis of exploiting hierarchies of image filters often heavily represented by edge detectors, this may impact on their effectiveness in such circumstances. The CIMEDNet that was trained with frozen convolutional layers also fared well, beating the IMEDNet model on the same noisy datasets despite not scoring as well on the noiseless dataset. This indicates that the feature detectors in the convolutional layers allow for more robust generalization whereas fully-connected layers are more inclined to overfit.

The qualitative result samples in Fig. 3, are also interesting. Here, original trajectories are shown in blue whereas trajectories calculated by the neural networks are shown in red and samples in matching dataset rows are identical for a fair comparison between each network. Results using the s-MNIST-RC-AWGN dataset are omitted as the noise levels are so pathologically difficult that the qualitative results are comparatively worthless. However, the CIMEDNet

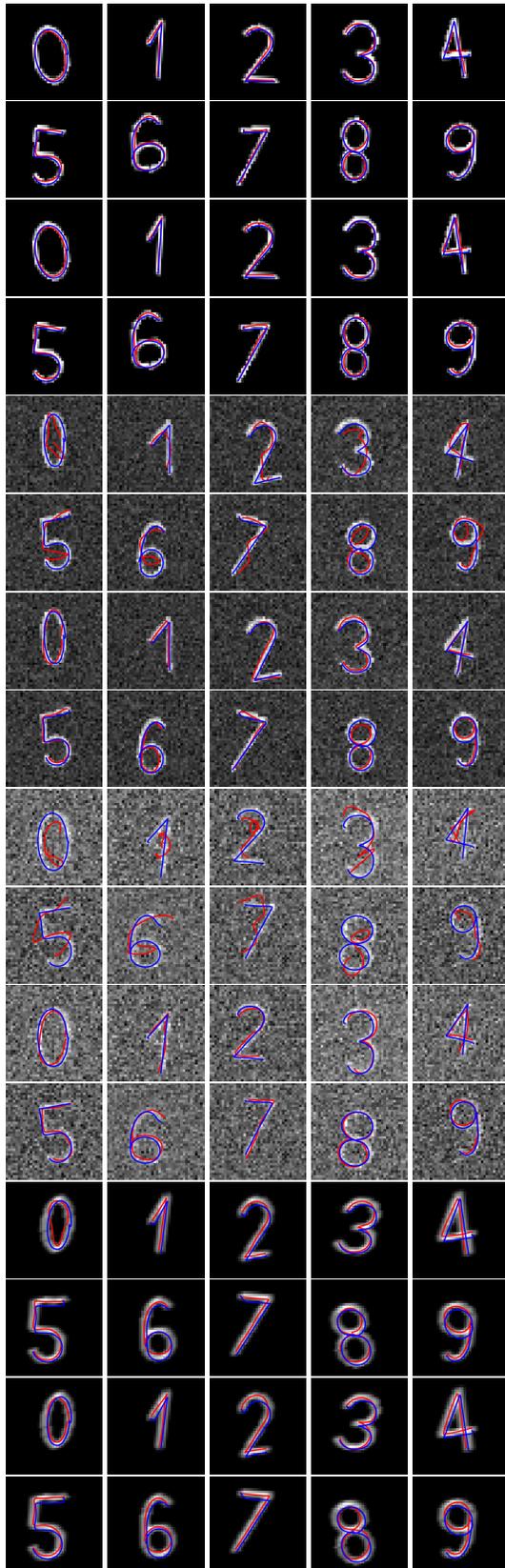


Figure 3: Example results for IMEDNet (rows 1, 2, 5, 6, 9, 10, 13 & 14) & CIMEDNet trained end-to-end (rows 3, 4, 7, 8, 11, 12, 15 & 16). Rows 1-4: s-MNIST, rows 5-8: s-MNIST-AWGN-19.0, rows 9-12: s-MNIST-AWGN-9.5 and rows 13-16: s-MNIST-MB.

model often performs surprisingly well given that it was not trained or fine-tuned on the noisy data. Both models appear to produce highly legible writing trajectories that closely match the actual trajectories in the case of the s-MNIST-MB dataset, but the CIMEDNet model is demonstrably superior to IMEDNet in many cases with the s-MNIST-AWGN-19.0 and s-MNIST-AWGN-9.5 data, producing much more legible results and demonstrating the robustness of the convolutional layers in dealing with even high noise levels.

4 CONCLUSIONS AND FUTURE WORK

We have presented an extended form of an encoder-decoder neural network for image-to-motion prediction that employs convolutional layers in the encoder in order to make the image recognition component more robust to noisy input. We have demonstrated that this architecture outperforms its predecessor on a variety of different kinds of noise. Regarding future work, we intend to further expand the capabilities of this model by incorporating layers from more powerful pre-trained CNN models into the encoder and training the network on more challenging image sets. One challenge here lies in either finding suitable image datasets that include trajectory information in their target outputs or in finding other means of producing images with corresponding motion trajectories, e.g. by gathering both in a robot simulation environment.

ACKNOWLEDGMENTS

This work has received funding from the EU’s Horizon 2020 RIA AUTOWARE (GA no. 723909); the Slovenian Research Agency under GA no. J2-7360; JSPS KAKENHI JP16H06565; NEDO; the Commissioned Research of NICT; the NICT Japan Trust (International research cooperation program); and JST-Mirai Program Grant Number JPMJMI18B8, Japan.

REFERENCES

- [1] V. Badrinarayanan, A. Kendall, and R. Cipolla. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (Dec. 2017), 2481–2495.
- [2] Saikat Basu, Manohar Karki, Sangram Ganguly, Robert DiBiano, Supratik Mukhopadhyay, Shreekanth Gayaka, Rajgopal Kannan, and Ramakrishna Nemani. 2017. Learning Sparse Feature Representations Using Probabilistic Quadrees and Deep Belief Nets. *Neural Processing Letters* 45, 3 (June 2017), 855–867.
- [3] N. Chen, J. Bayer, S. Urban, and P. van der Smagt. 2015. Efficient Movement Representation by Embedding Dynamic Movement Primitives in Deep Autoencoders. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. 434–440.
- [4] N. Chen, M. Karl, and P. van der Smagt. 2016. Dynamic Movement Primitives in Latent Space of Time-Dependent Variational Autoencoders. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. 629–636.
- [5] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. 2013. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural computation* 25, 2 (2013), 328–373.
- [6] Rok Pahič, Andrej Gams, Aleš Ude, and Jun Morimoto. 2018. Deep Encoder-Decoder Networks for Mapping Raw Images to Dynamic Movement Primitives. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia, 5863–5868.
- [7] A. Pervez, Y. Mao, and D. Lee. 2017. Learning Deep Movement Primitives Using Convolutional Neural Networks. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. 191–197.
- [8] A. Ude, A. Gams, T. Asfour, and J. Morimoto. 2010. Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives. *IEEE Transactions on Robotics* 26, 5 (Oct. 2010), 800–815.
- [9] P. C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata. 2017. Repeatable Folding Task by Humanoid Robot Worker Using Deep Learning. *IEEE Robotics and Automation Letters* 2, 2 (April 2017), 397–403.

Indeks avtorjev / Author index

Brandstötter Mathias	16
Breiling Benjamin	17
Deniša Miha	12
Dieber Bernhard	16, 17
Gams Andrej	18
Gašpar Timotej	12
Haspl Thomas	17
Hofbaur Michael	22
Lucchi Matteo	16
Mühlbacher-Karrer Stephan	16
Nemec Bojan	7
Pahič Rok	23
Pichler Horst	16
Radanovič Primož	12
Reberšek Simon	18
Ridge Barry	23
Simonič Mihael	7
Ude Aleš	7, 12, 18
Weyrer Matthias	22
Wohlhart Lucas	11

IS
20
19

Konferenca / Conference

Uredila / Edited by

**Robotika /
Robotics**

Andrej Gams, Aleš Ude