Interreg

EUROPEAN UNION

Grande Région | Großregion

Robotix-Academy

Fonds européen de développement régional | Europäischer Fonds für regionale Entwicklung

Axe prioritaire | Prioritätsachse 4
Compétitivité et attractivité
Wettbewerbsfähigkeit und Attraktivität

Robotix-Academy Conference for Industrial Robotics (RACIR) 2020

Partenaires du projet | Projektpartner:

ZeMA

uni.lu UNIVERSITÉ DU LUXEMBOURG

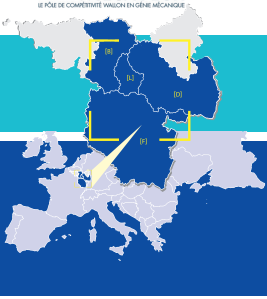
LIÈGE université

UNIVERSITÉ DE LORRAINE

Umwelt-Campus Birkenfeld | HOCHSCHULE TRIER

PÔLE MECATECH LE PÔLE DE COMPÉTITIVITÉ WALLON EN GÉNIE MÉCANIQUE

Rainer Müller, Peter Plapper, Olivier Brüls, Wolfgang Gerke, Gabriel Abba, Bassem Hichri, Ali Kanso (Hrsg.)



www.robotix.academy

Berichte aus der Robotik

**Rainer Müller, Peter Plapper,
Olivier Brüls, Wolfgang Gerke, Gabriel Abba,
Bassem Hichri, Ali Kanso (Hrsg.)**

Robotix-Academy Conference for Industrial Robotics (RACIR) 2020

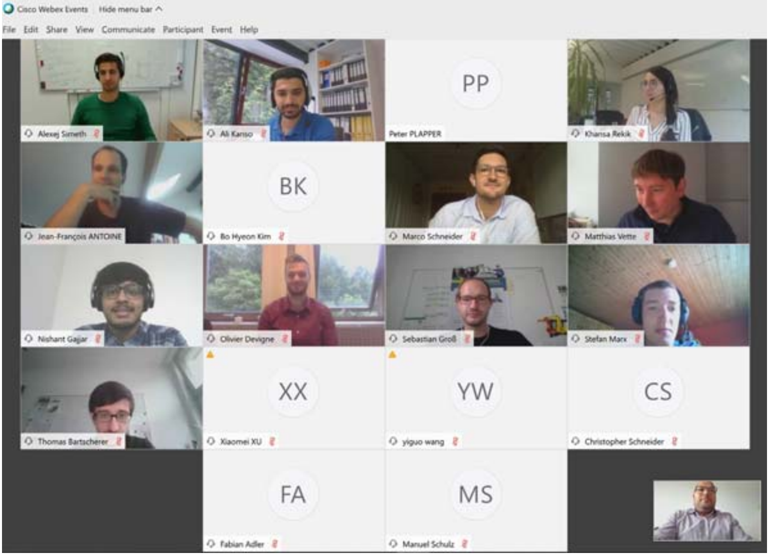
Shaker Verlag
Düren 2020

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Printed in Germany.

ISBN 978-3-8440-7606-6
ISSN 1434-8098

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren
Phone: 0049/2421/99011-0 • Telefax: 0049/2421/99011-9
Internet: www.shaker.de • e-mail: info@shaker.de



Robotix-Academy Conference for Industrial Robotics (RACIR) 2020

Preface:

Robotix-Academy Conference for Industrial Robotics (RACIR) is held at ZeMA, Germany, during July 16, 2020 as a web conference due to the effects of the Covid-19 crisis.

The venue for RACIR 2020 is the ZeMA - Center for Mechatronics and Automation. Founded in 2009, ZeMA sees itself as a development partner with the goal of industrialization and technology transfer of research and development results.

Working closely with institutes and chairs at Saarland University (UdS) and the University of Applied Sciences (htw saar), ZeMA passes on its research results to companies through an actively pursued technology transfer. In addition, ZeMA plays an active role in academic training.

The topics concerned by RACIR are: robot design, robot kinematics/dynamics/control, system integration, sensor/actuator networks, distributed and cloud robotics, bioinspired systems, service robots, robotics in automation, biomedical applications, autonomous vehicles (land, sea, and air), robot perception, manipulation with multifinger hands, micro/nano systems, sensor information, robot vision, multimodal interfaceand human-robot interaction.

Acknowledgements:

The Robotix-Academy partners and the participating students are acknowledged for their contributions and participation to the conference.

The organisation committee and involved persons are also acknowledged for their help and support.

Content

1 **Soft finger modeling using a nonsmooth contact approach** 1
Olivier Devigne, Alejandro Cosimo and Olivier Brüls

2 **Modified Feedback Linearization for Underactuated Cable Robot Control: Case studies** 7
Atal Anil Kumar, Jean-François Antoine, Vianney Papot, Patrick Zattarin and Prof. Dr. Gabriel Abba

3 **A synchronized-HRC pick and place application using an intelligent vision system** 14
Prof. Dr.-Ing. Rainer Müller, Nishant Ketan Gajjar, Ahmad El Masri and Khansa Rekik

4 **Hybrid Workstations: A Simultaneous Planning Method for Economic-oriented Selection between Industrial and Collaborative Robots** 19
Christopher Schneider, Thomas Suchanek, Martina Hutter-Mironovova, Francisco Hernandez, Mohamad Bdiwi and Matthias Putz

5 **Practically Oriented Investigation of Sensitive Robots Regarding the Execution of Force Controlled Applications** 27
Prof. Dr.-Ing. Rainer Müller, Ali Kanso and Marco Schneider

6 **External communication of the robot ABB YuMi via virtual machine with analysis of hybrid position-force control** 35
Wang Yiguo, Meryem Taghbalout, Jean-François Antoine and Gabriel Abba

7 **Mobile Robot Lifting Mechanism Design for Manipulation and transportation task** 41
Bassem Hichri and Peter Plapper

8 **Mobile Robots Target Reaching and Virtual Structure Navigation Based on Limit-Cycle Method** 49
Bassem Hichri and Peter Plapper

9 **Unconventional path planning for a serial kinematics robot with Reinforcement Learning using the example of the wire loop game** 55
Prof. Dr.-Ing. Rainer Müller, Ali Kanso and Stefan Marx

Soft finger modeling using a nonsmooth contact approach

Olivier Devigne
Department of Aerospace and Mechanical Engineering
University of Liège
Liège, Belgium
o.devigne@uliege.be

Alejandro Cosimo
Department of Aerospace and Mechanical Engineering
University of Liège
Liège, Belgium
Centro de Investigación de Métodos Computacionales (CIMEC)
Universidad Nacional del Litoral - CONICET
Santa Fe, Argentina
acosimo@uliege.be

Olivier Brüls
Department of Aerospace and Mechanical Engineering
University of Liège
Liège, Belgium
o.bruls@uliege.be

***Abstract*—In the frame of accurate grasping and safer human-robot interactions, soft robots are an emerging and promising technology. Due to the fact that they do not rely on joints to produce a motion, but on deformation, they have a theoretically infinite number of degrees of freedom. This particularity calls for advanced numerical models to analyze them. Although some modeling tools have already been developed by other research teams, many open questions remain and should be addressed to accurately represent this kind of robots. Our work relies on the development of a research code for analyzing flexible multibody systems. This code is based on a Lie group formalism which is coupled with state-of-the-art nonsmooth algorithms for solving contact interactions. This numerical formulation opens the possibility to later consider more advanced models for describing the flexibility characterizing these problems, such as, for example, geometrically exact beam and shell elements. In this paper, a frictionless soft finger model able to interact with a sphere is introduced as a first prototype intended to test and present our code capabilities.**

***Index Terms*—soft robot, soft gripper, geometrically exact, nonsmooth, contact**

I. Introduction

A new generation of robots has recently made its appearance in the robotics community. Made of so-called “soft” materials, such as plastic or silicone, these robots, in contrast to classical “rigid” robots, achieve their tasks by following a trajectory that is accomplished through a deformation of their structure. These soft robots can be used in various applications such as those requiring a safer human-robot interaction or those involving the manipulation of fragile objects [1]–[3]. Indeed, the soft nature of the material used to build them allows for extended grasping capabilities which is achieved by modulating the shape of the gripper and the contact force exerted on the object. This also ensures a reduction in the severity of impacts in environments involving interactions with humans. Soft robots are also particularly relevant in the frame of surgical robotics [4]. Moreover, they do not depend on classical manufacturing techniques and can be easily replaced at low financial costs, for instance, using 3D printing.

As previously mentioned, soft robots do not rely on joints to move, as it is the case for classical industrial robots, but on a deformation of their own structure. Three main actuation types can be distinguished. The first one is based on a local deformation induced by a linear actuator such as a pneumatic cylinder. The second one relies on pressure or vacuum actuation by inflating or deflating a chamber inside the structure. This actuation type produces a more global action on the whole robot. The third one is the technique which is investigated in this paper, called cable actuation. In this case, a cable is attached to different points of the compliant robot and is pulled so that the structure is locally and/or globally deformed.

Based on the fact that soft robots rely on the deformation of their structure to achieve a task, theoretically speaking, they can be characterized by an infinite number of degrees of freedom. This complexity calls for advanced numerical models for simulation, virtual prototyping or for control design purpose. Many times, the simulation should be executed in real-time [5], which can motivate the drastic simplification of the physics of the problem, as it is the case in [6], where only a sticking friction model is implemented for the gripper. The error on the controlled point between the simulation and the physical demonstrator can be in the order of several millimeters. This value should be put in light with the accuracy of an industrial robot which achieves a precision of some tenths of a millimeter.

A myriad of applications are targeted by soft robots. This paper specifically focuses on finger-like grippers. Although conceptually simple, these "soft fingers" reveal several interesting numerical challenges. Indeed, contact and friction between the gripper and the object (or with the gripper itself) should be considered. Moreover, the inherent flexibility of the gripper must be taken into account. Finally, control parameters can be tuned through experimental identification.

In order to deal with these challenges, a plugin in the open-source software SOFA is developed in [7]. However, they have a constraint on the number of vertices that can be used to approximate the 3D geometry in order to satisfy the real-time requirement. They are planning to adopt reduced order models in order to remove this constraint. Nevertheless, although these models perform well for preliminary analyses, they cannot be used for detailed analyses. In addition, their plugin is not open-source. In the context of the computer animation community, a methodology to solve contact problems based on a nonsmooth formalism is tested on the simulation of a soft robot in [8]. Despite some nice features, bilateral constraints are replaced by compliant models, and the contacts are described using a kind of node-to-surface technology that is known to have many drawbacks for the collision of flexible to flexible bodies [9]. This motivates the development of a more general research code, able to deal with the numerical challenges mentioned above.

The aim of this paper is double. Firstly, we introduce a robust modeling of highly flexible systems using a geometrically exact 3D approach [10]. The representation is based on the nonlinear finite element method where a local frame is defined at each node, and is treated as an element of the special Euclidean group $SE(3)$. The ability to express the equations of motion in $SE(3)$ yields interesting properties, noticeably for flexible elements, because their deformation measure is expressed in a local frame, giving an invariance property to the equations under a superimposed Euclidean transformation [11], [12]. Secondly, this representation is coupled with a more physical contact model relying on a sophisticated nonsmooth solver based on the generalized-α integration scheme [13]. As an illustration of this algorithm capabilities, a frictionless soft finger gripper is presented.

The paper is organized as follows. In Section II, the nonsmooth equations of motion and the time integration method are briefly presented. In Section III, the soft finger is first described as a physical object before introducing the modeling assumptions. Results are shown in Section IV. Finally, Section V gives the conclusions and future work perspectives.

II. Equations of motion and time integration

In order to describe the motion of flexible multibody dynamic systems, such as soft robots, the nonlinear finite element method is adopted [10]. In this setting, the elements comprising a mechanical system are represented by a set of nodal variables q of nodal position and orientation. It must be emphasized that when dealing with large rotations, q does not belong to a linear vector space but to a Lie group [14]. In the current work, we adopt the special Euclidean group $SE(3)$, where the vector $\boldsymbol{x} \in \mathbb{R}^3$ is used to represent the position of a node and the matrix $\boldsymbol{R}$, which belongs to the rotation group $SO(3)$, is used to represent its orientation, both measured with respect to the chosen inertial frame. An element from $SE(3)$ can be assimilated to a local frame attached to the body of the object under analysis. In this context, the translational velocity is represented by $\boldsymbol{u} = \boldsymbol{R}^T \dot{\boldsymbol{x}}$ which is interpreted as the velocity of the reference point $\boldsymbol{x}$ with respect to the inertial frame but resolved on the body-attached frame. The rotational velocity is represented by the vector $\boldsymbol{\Omega} \in \mathbb{R}^3$ in the body-attached frame. By arranging both velocities in the vector $\boldsymbol{v}^T = [\boldsymbol{u}^T \boldsymbol{\Omega}^T]$, the relation between the time derivative of the configuration $\dot{q}$ and $\boldsymbol{v}$ is given by a (nonlinear) kinematic compatibility condition of the form $\dot{q} = h(q, \boldsymbol{v})$.

In any multibody system, it is of great importance to be able to model kinematic joints which describe the restricted relative motion between two bodies. They are stated as equality conditions and, therefore, are classified as bilateral constraints. In order to model contact, we can assume that it develops as an impulsive process in which the velocity changes instantaneously in a discontinuous manner at an impact event. This nonsmooth (or impulsive) description of contact processes introduces non-equality conditions or unilateral constraints to the problem. The contact interaction between two bodies A and B at the points $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ from bodies A and B, respectively, is described by the gap $g = \boldsymbol{n}^T(\boldsymbol{x}_B - \boldsymbol{x}_A)$, where $\boldsymbol{n}$ is the outward unit normal from the surface of body A. At position level, whenever there is contact, the gap is zero and the contact reaction force F_c takes a positive value. On the contrary, when there is no contact, *i.e.* $g > 0$, the contact force vanishes. This complementarity relation is stated by the Signorini contact law as $0 \leq g \perp F_c \geq 0$, which is an abbreviation of the three conditions $g \geq 0$, $F_c \geq 0$ and $gF_c = 0$. At velocity level, a similar relation holds, however, an impact law must be provided in order to relate the pre-impact and post-impact velocities. It is important to note that the velocity field will be not continuous. Therefore, the usual equations of motions expressed in terms of partial differential equations are not valid anymore. In order to solve this issue, the equations of motion are written in terms of differential measures. The interested reader can find out more about this in [15], [16].

Under this setting, the equations of motion for a frictionless multibody system with unilateral and bilateral constraints expressed at velocity level are written in the following form:

$$
\begin{aligned}
\dot{q}^+ &= h(q, \boldsymbol{v}^+) && \text{(1a)}\\
\boldsymbol{M}(q)\,\mathrm{d}\boldsymbol{v} - \boldsymbol{g}_q^T\,\mathrm{d}\boldsymbol{i} &= \boldsymbol{f}(q, \boldsymbol{v}, t)\,\mathrm{d}t && \text{(1b)}\\
-\boldsymbol{g}_q^{\overline{\mathcal{U}}}\,\boldsymbol{v}^+ &= \boldsymbol{0} && \text{(1c)}\\
\text{if } g^j(q) \leq 0 \text{ then } 0 \leq g_q^j\,\boldsymbol{v}^+ &+ e^j\,g_q^j\,\boldsymbol{v}^- \perp \mathrm{d}i^j \geq 0 && \text{(1d)}\\
& \forall\, j \in \mathcal{U}
\end{aligned}
$$

where

- t is the time, and $\mathrm{d}t$ is the corresponding standard Lebesgue measure.

- $q(t)$ is the set of nodal variables, which are absolutely continuous in time.
- $\mathcal{U}$ denotes the set of indices of the unilateral constraints, $\overline{\mathcal{U}}$ is its complementarity set, *i.e.*, the set of bilateral constraints, $\mathcal{C} = \mathcal{U} \cup \overline{\mathcal{U}}$ is the full set of constraints.
- $\boldsymbol{g}$ is the combined set of bilateral and unilateral constraints, and $\boldsymbol{g}_q(q)$ is the corresponding matrix of constraint gradients.
- $\dot{q}^+(t) = \lim_{\tau \to t, \tau > t} \dot{q}(\tau)$ and $\boldsymbol{v}^+(t) = \lim_{\tau \to t, \tau > t} \boldsymbol{v}(\tau)$ are the right limits of the velocity, which are functions of bounded variations. Similarly, $\boldsymbol{v}^-(t) = \lim_{\tau \to t, \tau < t} \boldsymbol{v}(\tau)$ is the left limit of the velocity. If one assumes that the contact point j is in contact, then $g^j(q) \leq 0$ and $\mathrm{d}i^j \geq 0$. In this case, Eq. (1d) expresses that in the event of $\mathrm{d}i^j > 0$, the relative post-impact velocity $g_q^j \boldsymbol{v}^+$ has to change instantaneously by an impulsive process, modeled by enforcing the Newton impact law $g_q^j \boldsymbol{v}^+ = -e^j g_q^j \boldsymbol{v}^-$, where e^j is the coefficient of restitution at the contact point j. In the event in which the contact point j is superfluous (see *e.g.* [17]), $\mathrm{d}i^j = 0$ and $g_q^j \boldsymbol{v}^+$ can take any value satisfying $g_q^j \boldsymbol{v}^+ \geq -e^j g_q^j \boldsymbol{v}^-$. One can observe that in Eq. (1d), it is implicitly assumed that in the case of $g^j(q) > 0$, then $\mathrm{d}i^j = 0$. In what follows, for simplicity, $\boldsymbol{v}(t)$ and $\dot{q}(t)$ will be used to denote $\boldsymbol{v}^+(t)$ and $\dot{q}^+(t)$, respectively.
- $\boldsymbol{f}(q, \boldsymbol{v}, t) = \boldsymbol{f}^{ext}(t) - \boldsymbol{f}^{cin}(q, \boldsymbol{v}) - \boldsymbol{f}^{damp}(q, \boldsymbol{v}) - \boldsymbol{f}^{int}(q)$ collects the external, complementary inertia, damping and internal forces.
- $\boldsymbol{M}(q)$ is the mass matrix which may, in general, depend on the coordinates.
- $\mathrm{d}\boldsymbol{v}$ is the differential measure associated with the velocity $\boldsymbol{v}$, assumed to be of bounded variations.
- $\mathrm{d}\boldsymbol{i}$ is the impulse measure of the unilateral contact reaction and the bilateral constraint forces.

The numerical integration of the equations of motion must be performed with special care, not only because of the nonsmooth nature of the equations but also because of the presence of flexible components. In this context, time integration schemes can be classified into event-driven and time-stepping schemes. The former adapt their time step to the impact events, whilst the others do not. One inherent disadvantage of event-driven methods is that they do not perform well for systems with accumulation points or a large number of contacts. Therefore, this work concentrates on time-stepping methods. One of the most widespread method of this kind is the Moreau–Jean scheme [18]–[20]. Despite its good performance, it is dissipative for problems involving flexibility and suffers from drift issues at position level. In order to avoid these issues, we adopt the nonsmooth generalized-α (NSGA) scheme. It was introduced in [21] for the solution of contact and impact problems in multibody dynamics. This algorithm artificially splits the motion into smooth and nonsmooth (impulsive) components. In order to correctly capture the vibration effects of problems with flexible components, the smooth part is integrated using the second-order accurate generalized-α method, whereas the nonsmooth part is integrated using a first-order scheme. One distinctive feature of this method is that, through the adoption of a similar procedure proposed in [22], the unilateral and bilateral constraints are simultaneously satisfied both at position and velocity levels in order to avoid drift phenomena. The resulting scheme is characterized by a set of three coupled sub-problems: one for the smooth prediction of the motion and two others for the corrections at position and at velocity levels. Quite recently, a decoupled version of the NSGA method was presented in [13]. This specific version is the one adopted in this work. Two noticeable differences with respect to its predecessor [21] should be highlighted. On the one hand, the definition of the splitting is modified in order to ensure a decoupling of the three different subsets of equations that need to be solved at every time step. This feature improves considerably the robustness of the integrator for problems involving nonlinear bilateral constraints and flexible elements. From a practical point of view, the NSGA method involves the following steps, each of them performed in a decoupled manner at each time step of the simulation:

1) Solve for the smooth prediction of the motion: only bilateral constraints are involved.
2) Collision detection: detect which pair of bodies could be in contact and create the corresponding contact elements.
3) Solve for the position correction: both bilateral and unilateral constraints are involved.
4) Solve for the velocity jump: both bilateral and unilateral constraints are involved.

Remark: in order to perform the collision detection phase the corresponding capabilities provided by the Bullet Physics library [23] are used. The rest of the algorithmic steps are implemented within our in-house code.

III. Soft finger demonstrator

In this Section, a soft finger-like gripper is presented. Its aim is to provide a conceptually simple but yet representative model. Indeed, nonsmooth behaviors are dominating during the grasping operation. The physical robot that our numerical model wishes to represent is first introduced. Afterwards, the modeling assumptions that have been made are explained.

A. Physical demonstrator description

In order to highlight nonsmooth behaviors, a soft finger gripper displaying a motion history characterized by many impact events is modeled. A pressure-actuated version of this example is well-known in the soft robotics community under the name PneuNet [24]. However, the finger can also be cable-actuated, as presented in [7]. This last technology is investigated in this paper. An illustration of the system is presented in Fig. 1. When the cable is pulled, the finger will bend thanks to the low rigidity of the interphalangeal spaces. In the case of a complete gripper, three fingers are usually used to grasp objects.

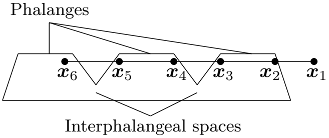
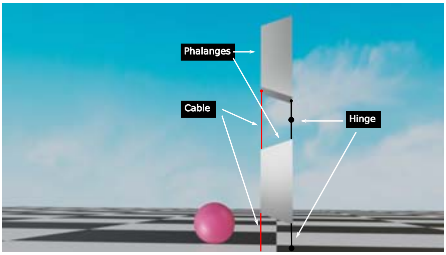
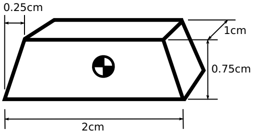


Fig. 1: Cable actuation of a soft finger. The different passing points of the cable through the structure are represented.



(a) Initial configuration of the sphere and finger ($t = 0$ s).



(b) Geometry of the finger phalanges.

Fig. 2: Initial configuration of the finger model and geometry of its phalanges.

B. Numerical model assumptions

In this first virtual prototype, the finger is modeled using the following assumptions. Firstly, although the finger is made of a soft material, it can be observed, as it is the case for a real human finger, that the deformation is highly localized in the interphalangeal space. Thus, as a first approximation, we consider the finger as made of rigid bodies interconnected by hinge joints, allowing it to bend. The implication of this assumption is that the deformation of the phalange is neglected by the model. The torque resulting from the local deformation of the interphalangeal space is modeled by a torsional spring element coupled to the hinge joint.

Secondly, the cable needs to be modeled. Considering that the cable is extensible, the length variation, according to the notation in Fig. 1, can be expressed as

$$\varDelta l = ||\boldsymbol{x}_2 - \boldsymbol{x}_1|| + ||\boldsymbol{x}_3 - \boldsymbol{x}_2|| + \cdots + ||\boldsymbol{x}_n - \boldsymbol{x}_{n-1}|| - l \quad (2)$$

Assuming a linear elastic behavior in the axial direction, the potential energy of the cable writes

$$\sigma = E\varepsilon \quad (3)$$

$$\mathcal{V} = \frac{1}{2}\int_0^l EA\varepsilon^2 ds \quad (4)$$

$$= \frac{1}{2}\frac{EA}{l}\varDelta l^2 \quad (5)$$

with the unstretched length l, the strain $\varepsilon = \frac{\varDelta l}{l}$, which is considered constant over the whole cable, the Young's modulus E and the cross-sectional area A. Under these assumptions, the cable behaves as a spring of stiffness $\frac{EA}{l}$. Because of the assumption of a linear stress-strain constitutive law in Eq. 3, the model is not able to represent slackness. Indeed, a real cable can only withstand traction forces. Therefore, the current version of this model is only valid in traction.

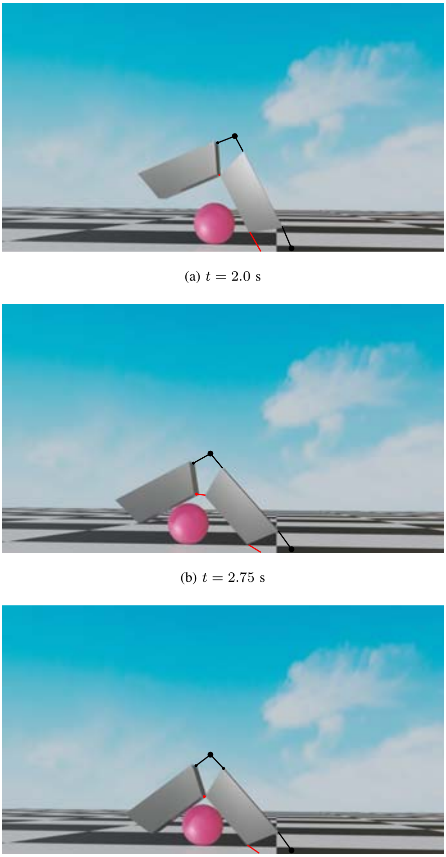
Finally, no friction is considered in this preliminary model. The grasping operation represented by the model is thus incomplete, in the sense that the gripper can get in contact with the object but the contact forces are only represented in the normal direction.

IV. Numerical results

The dynamics of a finger-like gripper comprised of one finger made of two phalanges is studied next. The initial configuration of the problem can be observed in Fig. 2(a) where it can be appreciated that the gripper will interact with a sphere that is free to move on a flat ground.

The aim behind this example consists in showing the capabilities that we have for the modeling of this kind of problems. The parameters of the studied problem are specified as follows. The finger is actuated by a flexible cable with a stiffness constant of 10^5 N/cm which links the two phalanges and the actuation mechanism. This mechanism exerts a force which increases from $(0, 0, 0)$ to $(0, -120, 0)$ N in 2 seconds and it remains constant at that value for the rest of the simulation. The two phalanges are joined by a hinge which, as previously mentioned, also models an internal stiffness and damping, taking as parameters a stiffness constant of 10^{-2} N.cm/rad and a damping coefficient of 1 N.s/rad. One of the phalanges is linked to the ground through a hinge which is characterized by a stiffness constant of 1 N.cm/rad and a damping coefficient of 1 N.s/rad. The acceleration of gravity is neglected. The level of the ground is set at 1 cm in the y-direction. At the initial time step $t = 0$, the center of mass of the sphere of radius $r = 0.5$ cm is located at $(0, 1.5, 1.5)$ cm, and the center of mass of the two phalanges are located at $(0, 2.5, 0)$ cm and $(0, 5.5, 0)$ cm with respect to the inertial frame of reference. The geometry of the phalanges can be observed in Fig. 2(b). The shape and dimensions have been chosen to represent the same type of finger as in [7]. For the impact law, the restitution coefficient is taken as 0.

It should be emphasized, as mentioned above, that we are solving the frictionless problem. Certainly, friction is a really important phenomenon to take into account in this kind of applications. However, one of the aims of this work is to show

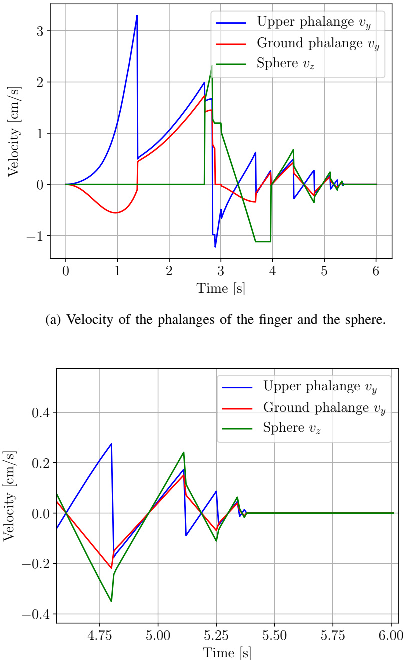


(a) $t = 2.0$ s

(b) $t = 2.75$ s

(c) $t = 3.14$ s

Fig. 3: Results for the soft finger simulation: render of the finger interacting with the sphere at different time steps.



(a) Velocity of the phalanges of the finger and the sphere.

(b) Zoom of the velocity of the phalanges of the finger and the sphere.

Fig. 4: Results of the soft finger simulation: velocities of the phalanges and the sphere.

the current state of our developments, not only in terms of physically-sounded numerical formulations, but also in terms of source code development to solve applications of interest for the robotics industry.

In Figs. 3, snapshots of the time evolution of the obtained solution can be observed. As it can be appreciated, in response to the actuation of the force transmitted through the cable, the gripper adapts its shape until it comes in contact with the sphere. At the end of the simulation, $t = 6$ s, the sphere is totally enclosed between the phalanges of the finger. In Fig. 4(a), the velocity of the phalanges of the finger and the sphere are shown. It is important to observe that the adopted nonsmooth formulation is able to correctly capture all the jumps in the velocity field (*i.e.* all the contacts). The first jump actually represents the contact between the two phalanges. In this case, a relatively large time increment of 0.01 s was used for the simulation, and, actually, higher values could even be used. Consequently, the nonsmooth nature of the adopted numerical scheme is efficient from the point of view of the size of the time increments. In addition, Fig. 4(b) shows that no numerical artifact is observed at the end of the simulation. That is, the velocity is exactly zero when the system is at rest, where other formulations could exhibit oscillations of numerical nature. Moreover, it has been verified that no penetration occurred, so that no drift phenomena are noticed.

V. Conclusions and future work

In this paper, a first prototype of a frictionless soft finger-like gripper was proposed. A Lie group formulation of the equations of motion is coupled with an efficient nonsmooth time integration algorithm in order to account for contacts between the gripper and the object. As a first approximation, a model consisting in two rigid bodies interconnected by a hinge joint is used to represent the finger, whereas a spring element is chosen to model the cable. Our model is able to capture nonsmooth interactions with a rigid sphere, accurately catching the discontinuities in the velocity. However, the deformation of the phalanges is not modeled and slack cables cannot be represented. Additionally, no friction is implemented, so that the grasping operation is incomplete. In the future, we wish to experimentally identify parameters from a real soft finger, implement a working friction model and use more sophisticated flexible elements such as beams or shells, taking advantage of our geometrically exact formulation.

Acknowledgment

This work is partly funded by the Robotix Academy project of the Greater Region.

References

[1] H. Lipson, "Challenges and opportunities for design, simulation, and fabrication of soft robots," *Soft Robotics*, vol. 1, no. 1, pp. 21–27, 2014.

[2] Z. Wang and S. Hirai, "A 3D printed soft gripper integrated with curvature sensor for studying soft grasping," in *2016 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2016, pp. 629–633.

[3] Y. Wei, Y. Chen, T. Ren, Q. Chen, C. Yan, Y. Yang, and Y. Li, "A novel, variable stiffness robotic gripper based on integrated soft actuating and particle jamming," *Soft Robotics*, vol. 3, no. 3, pp. 134–143, 2016.

[4] M. Cianchetti, T. Ranzani, G. Gerboni, T. Nanayakkara, K. Althoefer, P. Dasgupta, and A. Menciassi, "Soft robotics technologies to address shortcomings in today's minimally invasive surgery: the stiff-flop approach," *Soft robotics*, vol. 1, no. 2, pp. 122–131, 2014.

[5] E. Coevoet, A. Escande, and C. Duriez, "Optimization-based inverse model of soft robots with contact handling," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1413–1419, 2017.

[6] ——, "Soft robots locomotion and manipulation control using fem simulation and quadratic programming," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019, pp. 739–745.

[7] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.

[8] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and V. Makoviychuk, "Non-smooth newton methods for deformable multi-body dynamics," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–20, 2019.

[9] F. J. Cavalieri and A. Cardona, "An augmented Lagrangian technique combined with a mortar algorithm for modelling mechanical contact problems," *International Journal for Numerical Methods in Engineering*, vol. 93, no. 4, pp. 420–442, 2013.

[10] M. Géradin and A. Cardona, *Flexible Multibody Dynamics: A Finite Element Approach*. Wiley, 2001.

[11] V. Sonneville, "A geometric local frame approach for flexible multibody systems," Ph.D. dissertation, Université de Liège, Liège, Belgique, 2015.

[12] V. Sonneville, A. Cardona, and O. Brüls, "Geometrically exact beam finite element formulated on the special euclidean group se (3)," *Computer Methods in Applied Mechanics and Engineering*, vol. 268, pp. 451–474, 2014.

[13] A. Cosimo, J. Galvez, F. J. Cavalieri, A. Cardona, and O. Brüls, "A robust nonsmooth generalized-α scheme for flexible systems with impacts," *Multibody System Dynamics*, vol. 48, no. 2, pp. 127–149, 2020.

[14] O. Brüls, A. Cardona, and M. Arnold, "Lie group generalized-α time integration of constrained flexible multibody systems," *Mechanism and Machine Theory*, vol. 48, pp. 121–137, 2012.

[15] B. Brogliato, *Nonsmooth Mechanics*. Springer International Publishing, 2016.

[16] R. I. Leine and N. van de Wouw, Eds., *Stability and Convergence of Mechanical Systems with Unilateral Constraints*. Springer Berlin Heidelberg, 2008.

[17] C. Glocker, *An Introduction to Impacts*. Vienna: Springer Vienna, 2006, pp. 45–101.

[18] M. Jean and J. J. Moreau, "Dynamics in the presence of unilateral contacts and dry friction: A numerical approach," in *Unilateral Problems in Structural Analysis — 2*. Springer Vienna, 1987, pp. 151–196.

[19] J. J. Moreau, "Unilateral contact and dry friction in finite freedom dynamics," in *Nonsmooth Mechanics and Applications*. Springer Vienna, 1988, pp. 1–82.

[20] M. Jean, "The non-smooth contact dynamics method," *Computer Methods in Applied Mechanics and Engineering*, vol. 177, no. 3-4, pp. 235–257, 1999.

[21] O. Brüls, V. Acary, and A. Cardona, "Simultaneous enforcement of constraints at position and velocity levels in the nonsmooth generalized-α scheme," *Computer Methods in Applied Mechanics and Engineering*, vol. 281, pp. 131–161, 2014.

[22] C. W. Gear, B. Leimkuhler, and G. K. Gupta, "Automatic integration of Euler-Lagrange equations with constraints," *Journal of Computational and Applied Mathematics*, vol. 12-13, pp. 77–90, 1985.

[23] E. Coumans *et al.*, "Bullet physics library," *Open source: bullet-physics.org*, vol. 15, no. 49, p. 5, 2013.

[24] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides, "Soft robotics for chemists," *Angewandte Chemie*, vol. 123, no. 8, pp. 1930–1935, 2011.

Modified Feedback Linearization for Underactuated Cable Robot Control: Case studies

Atal Anil Kumar
LCFC
Université de Lorraine, Arts et Metiers Institute of Technology, HESAM Université
Metz, France
atal-anil.kumar@univ-lorraine.fr

Jean-François Antoine
LCFC
Université de Lorraine, Arts et Metiers Institute of Technology, HESAM Université
Metz, France
jean-francois.antoine@univ-lorraine.fr

Vianney Papot
LCFC
Université de Lorraine, Arts et Metiers Institute of Technology, HESAM Université
Metz, France
vianney.papot@univ-lorraine.fr

Patrick Zattarin
LCFC
Université de Lorraine, Arts et Metiers Institute of Technology, HESAM Université
Metz, France
patrick.zattarin@univ-lorraine.fr

Gabriel Abba
LCFC
Université de Lorraine, Arts et Metiers Institute of Technology, HESAM Université
Metz, France
gabriel.abba@univ-lorraine.fr

Abstract—**This paper presents the results of the simulations done to validate and analyze the performance of the modified feedback linearization control for an underactuated four Cable-Driven Parallel Robot (CDPR). Different conditions are defined with varying payload, velocity and trajectory and the response of the system using the proposed control is presented. It is shown that the solution stabilizes the system behavior and performs efficiently under varying conditions.**

Keywords—modified feedback linearization, underactuated, cable robot

I. Introduction

Cable-Driven Parallel Robots (CDPRs) is a special variant of traditional parallel robots in which the moving platform (MP) is connected to the base frame by a set of cables whose lengths are adjusted by actuated winches [1]. A CDPR is fully constrained if the end effector pose can be completely determined when actuators are locked and, thus, all cable length are assigned. Conversely, a CDPR is underconstrained if the end-effector preserves some freedoms once actuators are locked. This occurs either when the end-effector is controlled by a number of cables smaller than the number of degrees of freedom (DoF) that it possesses with respect to the base or when some cables become slack in a fully constrained robot [2]. In addition, if the number of actuators is less than the number of generalized coordinates needed to completely describe the manipulator, the robot is underactuated and thus inherently underconstrained as well [3].

Application of underactuated CDPRs with a limited number of cables can be found in tasks requiring a limited number of controlled DoFs or when a limitation of dexterity is acceptable in order to decrease complexity, cost, set-up time, the likelihood of cable interference, etc [4].

The application of classical input-output feedback linearization has been presented in [8]. However, the work did not show the effect of internal dynamics on the platform behavior at various points. The main contribution of this work is to present the effects of internal dynamics on the MP and to propose a modified feedback linearization control to stabilize the values of cables tensions which in turn helps in stabilizing the DoFs of the moving platform (mainly the platform orientations). The simulation results indicate that the modified control law performs significantly better than the classical I/O feedback linearization and can be implemented in the real prototype for validation.

The paper is organized as follows: section II presents the classical dynamic model of the CDPR. Section III presents the proposed modified feedback linearization method along with the mathematical preliminaries and the simulation conditions. This is followed by the results section presenting the various simulation results obtained. The final section concludes the work by highlighting the insights from the work and the future work to be done.

II. Dynamic model of the CDPR

The modelling and analysis methods developed for conventional rigid link manipulators cannot be directly applied to the cable-driven robots because of the unilateral constraints where the tensions in the cables must be considered [5].

A general sketch of cable-driven parallel robot is shown in (Fig. 1).

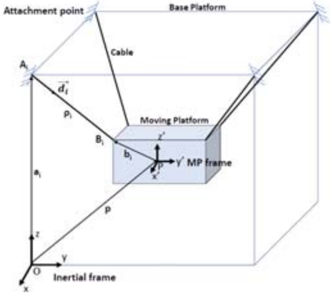


Fig. 1: Simple sketch of one of the cables of the CDPR

A fixed reference frame (*O, x, y, z*) attached to the base of a CDPR is referred to as the base frame. A moving reference frame (*P, x', y', z'*) is attached to the mobile platform where *P* is the reference point of the platform to be positioned by the mechanism. From (fig. 1), a_i and b_i are respectively defined as the vector connecting point O to point A_i and the vector connecting point P of the platform to the point B_i, both vectors being expressed in the base frame. The position p of the mobile platform is given by $\overrightarrow{OP}$. In order to reduce the complexity of computation in modelling, we assume the following [6]:

1) The mass of the cables is negligible and the cables are non-elastic.
2) The i^{th} cable is assumed to be taut between points and is therefore considered a straight segment and is denoted by ρ_i.
3) The moving platform is assumed to be a rigid body, defined by its mass and inertia matrix.

The equations of motion for a CDPR can be derived using Newton–Euler formulations provided all cables are in tension as shown in (1) [7].

$$\begin{bmatrix} mI_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_P \end{bmatrix} \begin{bmatrix} \ddot{p} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} 0_{3\times1} \\ \omega \times I_P \omega \end{bmatrix} + \begin{bmatrix} -mg \\ 0_{3\times1} \end{bmatrix} = -J^T\tau \qquad (1)$$

In this equation, m denotes the mass of the moving platform with the payload, I_P is a 3×3 matrix and denotes the inertia tensor of the end-effector about point P in the base frame, $I_{3\times3}$ is a 3×3 identity matrix, g denotes the gravity acceleration vector, τ denotes the vector of cables forces while scalar t_i denotes the tension force of the i^{th} cable, $\omega = [\omega_x, \omega_y, \omega_z]^T$ denotes the velocity vector of the orientation, $p = [p_x, p_y, p_z]^T$ denotes the position vector. Consider $X = [x, y, z, \alpha, \beta, \gamma]^T$ as generalized coordinates vector, in which $\theta = [\alpha, \beta, \gamma]^T$ denotes the vector of a set of Euler angles. With this definition, the rotation matrix can be written in terms of Euler angles as:

$$R = \begin{bmatrix} c\beta c\gamma & c\gamma s\alpha s\beta - c\alpha s\gamma & c\alpha c\gamma s\beta + s\alpha s\gamma \\ c\beta s\gamma & c\alpha c\gamma + s\alpha s\beta s\gamma & -c\gamma s\alpha + c\alpha s\beta s\gamma \\ -s\beta & c\beta s\alpha & c\alpha c\beta \end{bmatrix} \qquad (2)$$

where, s and c represent *sin* and *cos* functions, respectively. The angular velocity of the end-effector can be written in the following form,

$$\omega = E\dot{\theta} \qquad (3)$$

$$\dot{\theta} = [\dot{\alpha}, \dot{\beta}, \dot{\gamma}]^T \qquad (4)$$

in which,

$$E = \begin{bmatrix} \mathrm{c}\beta c\gamma & -s\gamma & 0 \\ \mathrm{c}\beta s\gamma & c\gamma & 0 \\ -s\beta & 0 & 1 \end{bmatrix} \qquad (5)$$

The equations of motion can be written in terms of X using the notations defined above. By some manipulations these equations may be derived as,

$$M(X)\ddot{X} + C(X, \dot{X})\dot{X} + G(X) = -J^T\tau \qquad (6)$$

where,

$$M(X) = \begin{bmatrix} mI_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_P E \end{bmatrix}$$

$$C(X, \dot{X}) = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_P\dot{E} + \left(E\dot{\theta}\right)_\times (I_P E) \end{bmatrix}$$

$$G(X) = \begin{bmatrix} -mg \\ 0_{3\times1} \end{bmatrix}$$

in which, the matrix $(E\dot{\theta})_\times$ is a skew-symmetric matrix. The Jacobian transpose of the CDPR is given by

$$J^T = \begin{bmatrix} d_1^O & \dots & d_i^O & \dots \\ b_1^O \times d_1^O & \dots & b_i^O \times d_i^O & \dots \end{bmatrix} \qquad (7)$$

where, d_i^O is the unit vector giving the direction of the i^{th} cable from its end point on the base frame *(O)* to its end point on the MP and b_i^O is the vector from the MP centre of gravity P to the end point B_i expressed in the inertial frame.
Equation (6) is finally represented as

$$M(X)\ddot{X} + N(X, \dot{X})\dot{X} = -J^T\tau \qquad (8)$$

where, $\quad N(X, \dot{X})\dot{X} = C(X, \dot{X})\dot{X} + G(X)$

Equation (8) is then used for the implementation of the input-output feedback linearization method.

III. MODIFIED INPUT-OUTPUT FEEDBACK LINEARIZATION

The simulation results of classical input-output feedback linearization (I/O FL) can be found in [8], [9]. A modified input-output feedback linearization approach has been proposed in [10] to reduce the effect of oscillatory internal dynamics due to the classical I/O FL and stabilize the system behaviour.

As the name suggests, this modified control is based on classical input-output feedback linearization. The novelty of this approach is that instead of using two different techniques, the same control law is executed on two separate branches. The output from each branch is then combined and given as the input to the system.

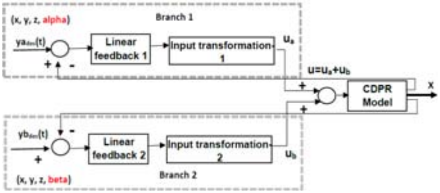


Fig. 2. Block diagram of the proposed modified feedback linearization

A. *Mathematical preliminaries for input-output feedback linearization*

The mathematical approach of the input-output feedback linearization (I/O FL) method for a nonlinear MIMO dynamic system of n^{th} order with m number of inputs and outputs is presented in this section. Further explanation of the technique in detail can be found in [11]. Consider a MIMO system described in the affine form as given below:

$$\dot{x}(t) = f(x,t) + g_1(x,t)u_1(t) + \cdots + g_m(x,t)u_m(t) \qquad (9)$$

$$y_1(t) = h_1(x,t)$$
$$\ldots$$
$$y_m(t) = h_m(x,t)$$

where, *i= 1..m – ith* inputs, *j=1..m – jth* outputs, $x(t) \in R^n$ is state vector, $u_i(t)$ is control input, $y_j(t)$ is the system output, $f(x,t)$, $g_i(x,t)$ and $h_j(x,t)$ are smooth nonlinear functions.

The basic principle of the input-output feedback linearization method is in finding an input transformation in the shape

$$u_i = \alpha_i(x) + \beta_i(x)v_i \qquad (10)$$

Where v_i is the new input, $\alpha_i(x), and, \beta_i(x)$ are nonlinear functions.

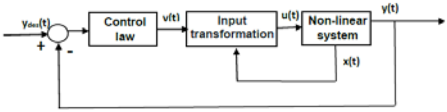


Fig. 3: Block diagram representation of the input-output linearization

Equation (10) helps in creating a linear relationship among the outputs y_i and the new inputs v_i decoupling the interaction between the original inputs and outputs. Following this decoupling, control algorithms for each subsystem with input and output independent of each other can be synthesized using the conventional linear control laws. In order to achieve this, each output is repeatedly differentiated until the input signals appear in the expression of derivation. The individual derivatives of outputs are calculated using lie derivatives which are marked as $L_f h$ and $L_g h$. The first derivative has the form

$$\dot{y}_j = L_f h_j(x) + \sum_{i=1}^{m} L_{g_i} h_j(x) u_i \qquad (11)$$

where, $$L_f h_j(x) = \frac{\partial h_j}{\partial x} f(x),\ L_{g_i} h_j(x) = \frac{\partial h_j}{\partial x} g_i(x)$$

If the expression $L_{g_i} h_j(x) = 0$ for all *i,*, it means that the inputs have not appeared in the derivation making it necessary to continue with the differentiation process till at least one input appears in the derivation. The resulting derivation takes the form

$$y_j^{r_j} = L_f^{r_j} h_j(x) + \sum_{i=1}^{m} L_{g_i} L_f^{r_j-1} h_j(x) u_i \qquad (12)$$

where, r_j represents the number of derivatives needed for at least one of the inputs to appear, also known as the relative order.

This approach is followed for each output y_j. The resulting *m* equations can be written in the form

$$\begin{bmatrix} y_1^{r_1} \\ \cdots \\ y_m^{r_m} \end{bmatrix} = \begin{bmatrix} L_f^{r_1} h_1(x) \\ \cdots \\ L_f^{r_m} h_m(x) \end{bmatrix} + E(x) \begin{bmatrix} u_1 \\ \cdots \\ u_m \end{bmatrix} \qquad (13)$$

where *E(x)* is a $m \times m$ matrix of shape

$$E(x) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1 & \cdots & L_{g_m} L_f^{r_1-1} h_1 \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m & \cdots & L_{g_m} L_f^{r_m-1} h_m \end{bmatrix}$$

If the matrix *E(x)* is regular, then it is possible to define the input transformation in the shape

$$\begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} = -E^{-1}(x) \begin{bmatrix} L_f^{r_1} h_1(x) \\ \vdots \\ L_f^{r_m} h_m(x) \end{bmatrix} + E^{-1}(x) \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \qquad (14)$$

Once the input transformation is completed as shown in (14) the linear control law is used to propose a feedback control for the linear system to ensure the desired behaviour of the nonlinear system using the conventional techniques. The relative degree (r_i) of the individual output is then used to calculate the overall vector relative degree of the system (*r*) to analyze the concept of internal dynamics.

$$r = r_1 + r_2 + \cdots + r_m \qquad (15)$$

From equation (15), we will be able to calculate to vector relative degree of the system (*r*). If the vector relative degree is less than the number of states of the system (*n*), there exists internal dynamics (ID) in the system. In order to apply the classical I/O feedback linearization, it is important to study the effect of ID on the overall behavior of the system.

B. Simulation Parameters

The application of classical input-output feedback linearization for a CDPR model can be found in [8]. The corresponding values of *α, β, γ* for the starting and final point is calculated from the static equilibrium program developed by the authors in [12]. The simulation parameters are as shown in table I and II.

TABLE I. SIMULATION PARAMETERS FOR THE CONTROL LAWS

Room dimension (m)	5*5*3
Platform dimension (m)	0.5*0.5*0.2
Max. and Min. cable tension (N)	500N and 1N respectively
Starting point*(t=0)*	x=2, y=0.5, z=1.5
Final point*(t=10)*	x=2, y=2, z=1.5
Mass of the platform including the object weight	30kg

TABLE II. CABLE ATTACHMENT POINTS FOR CENTRE OF MASS AT A HEIGHT OF 1.5M FROM BOTTOM

Cable no.	**MP**	**Base**
Cable 1	[2.25,2.25,1.7]	[0,0,3]
Cable 2	[2.25,2.75,1.7]	[0,5,3]
Cable 3	[2.75,2.75,1.7]	[5,5,3]
Cable 4	[2.75,2.25,1.7]	[5,0,3]

IV. RESULTS

The results of the simulation done are presented in this section. The initial results present the comparison of the proposed modified feedback linearization with the classical feedback linearization approach. Following this, two different conditions are presented. The performance of the proposed law with different payloads on the platform and with different payload on the input transformation block is also presented.

A. Comparison with classical feedback linearization

This section presents the comparison between the classical I/O FL and the proposed modified feedback linearization.

The simulation parameters are given in table I and II. A quintic polynomial was used to generate the desired trajectory to obtain smooth values for the acceleration and velocity.

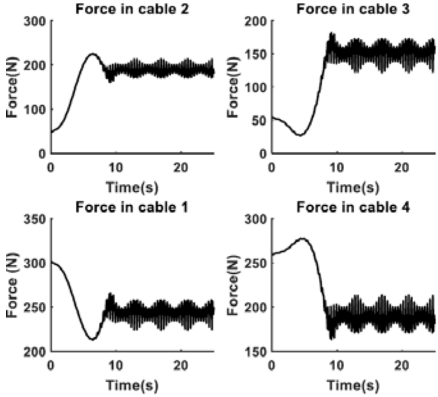


Fig. 4: Cable forces using the classical I/O FL for the simulation parameters considered

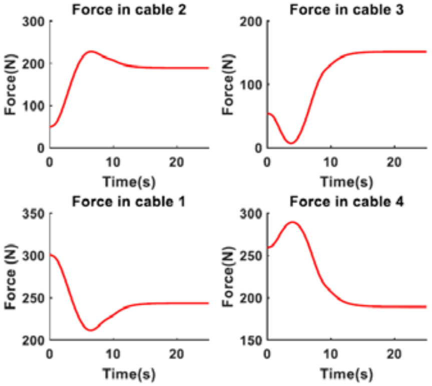


Fig. 5: Cable forces using modified FL

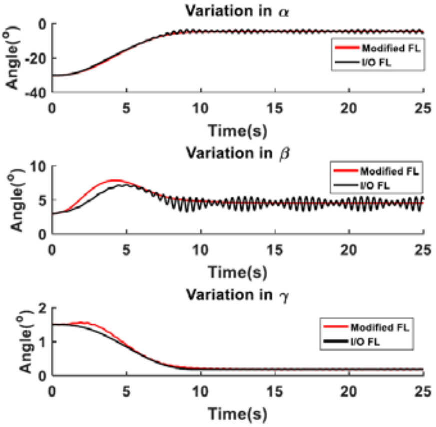


Fig. 6: Variation of platform orientations with modified I/O FL and classical I/O FL respectively

The cable forces generated by the modified control law to follow the desired trajectory is shown in fig. (5). It is seen that the values of the forces are positive and within the limits defined in table 1. Figure 6 shows the comparison of the platform orientation values generated by the modified control law and the classical I/O feedback linearization. It is clearly visible that the orientation values are more stable with the application of modified feedback linearization.

B. *Modified feedback linearization with different payloads*

The performance of the modified control for different payloads is presented here. Three different cases are considered and explained as follows:

Case a): The payload acting on the platform is reduced from 30kg to 25kg after 1 second and maintained at 25kg till the trajectory completion time is reached (10s). The payload is then increased to 30kg during the resting period (till t =25s).

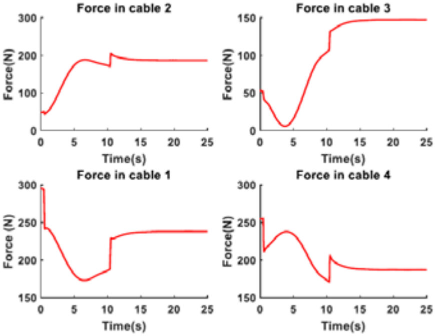


Fig. 7: Cable forces when mass on platform is reduced to 25kg during trajectory and then again to 30kg during resting time

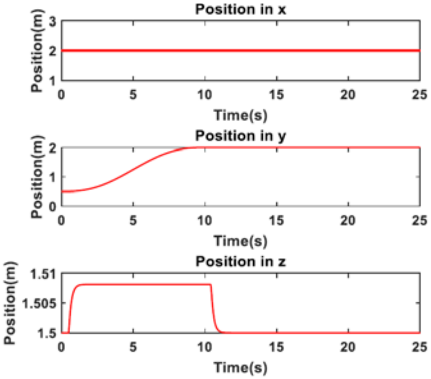


Fig. 8: Platform position when mass is reduced to 25kg during trajectory and then again to 30kg during resting time

Case b): The payload acting on the platform is increased from 30kg to 35kg after 1 second and maintained at 35kg till the trajectory completion time is reached (10s). The payload is then brought back to 30kg during the resting period (till t =25s).

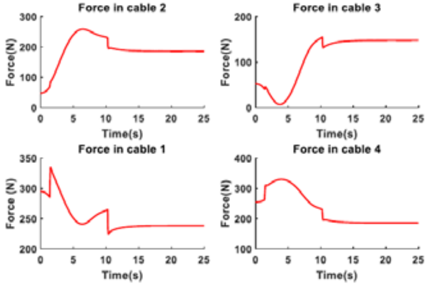


Fig. 9: Cable forces when mass on platform is increased to 35kg during trajectory and then again to 30kg during resting time

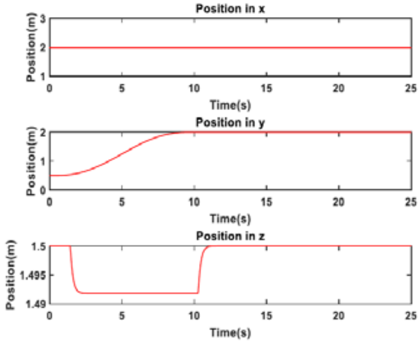


Fig. 10: Platform position when mass is increased to 35kg during trajectory and then again to 30kg during resting time

Case c): The payload acting on the platform is maintained at 30kg till the trajectory completion time is reached (10s). The payload is then reduced to 25kg during the resting period (till t =25s).

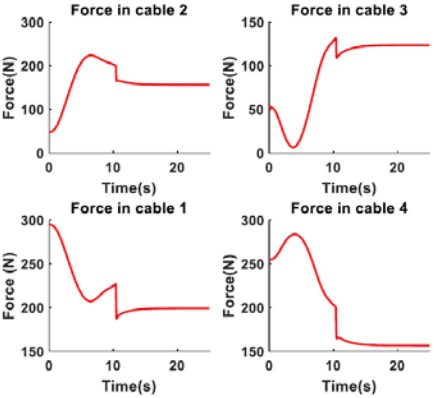


Fig. 11: Cable forces when mass on platform is 30kg during trajectory and then reduced to 25kg during resting time

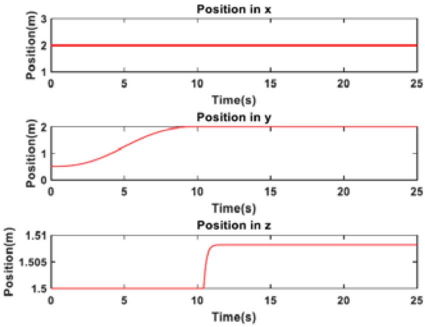


Fig. 12: Platform position when mass is 30kg during trajectory and then reduced to 25kg during resting time

It is evident from these cases that even though there is variation in the cable forces due to the changes in the payload, the control law is able to maintain the stability in the position and orientation of the platform. The cable tension values are also within the limit set for the study with no negative values generated.

C. Modified feedback linearization with different payload for the input transformation block

In this section, the payload in the input transformation block of the control law is kept at 25kg, while the actual payload acting on the platform is 30kg. This is done to understand the behavior of the system when the mass acting on the platform is more than the payload the control is designed to act on.

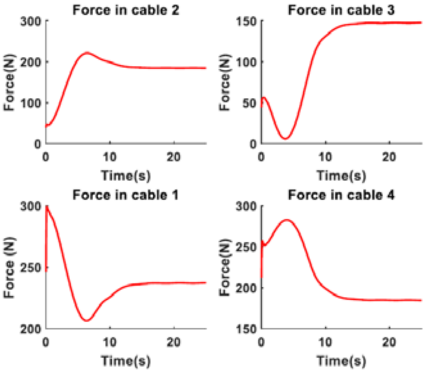


Fig. 13: Cable forces when mass on platform is 30kg and 25kg for input transformation block

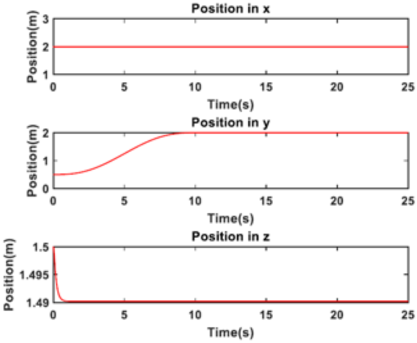


Fig. 14: Platform position when mass on platform is 30kg during trajectory and then reduced to 25kg during resting time

From the figures above, it is clear that the proposed control law is able to achieve the desired position. However, there is sudden change in the force values which has to be avoided to prevent damage to the actuators.

V. Conclusion

The performance of the proposed modified control law has been presented in this study. Several conditions were considered and the behavior of the system is shown through simulations. The results obtained indicates that the modified control law is able to perform efficiently keeping the system stable. Future works will involve the experimental validation of the law and the effect of cable elasticity on the system model.

Acknowledgment

We would like to thank the Robotix Academy for funding this work as a part of the project funded by INTERREG V-A Grand Region program.

References

[1] Merlet, J., Daney, D., 2010. “A portable, modular parallel wire crane for rescue operations”, in: Robotics and Automation (ICRA), 2010 IEEE International Conference On. IEEE, pp. 2834–2839.

[2] Verhoeven, R.: “Analysis of the workspace of tendon-based Stewart platforms” (Doctoral dissertation, Universität Duisburg-Essen), 2004.

[3] Ida, E., Bruckmann, T., Carricato, M., 2019. “Rest-to-Rest Trajectory Planning for Underactuated Cable-Driven Parallel Robots”. IEEE Trans. Robot. 1–14.

[4] Abbasnejad, G., Carricato, M., 2015. “Direct Geometrico-static Problem of Underconstrained Cable-Driven Parallel Robots With n Cables”. IEEE Trans. Robot. 31, 468–478.

[5] Alp, A.B., Agrawal, S.K., 2002. “Cable suspended robots: Feedback controllers with positive inputs”, in: American Control Conference, 2002. Proceedings of the 2002. IEEE, pp. 815–820.

[6] Gosselin, C., 2014. “Cable-driven parallel mechanisms: state of the art and perspectives”. Mech. Eng. Rev. 1, DSM0004–DSM0004.

[7] Begey, J., Cuvillon, L., Lesellier, M., Gouttefarde, M., Gangloff, J., 2019. “Dynamic Control of Parallel Robots Driven by Flexible Cables and Actuated by Position-Controlled Winches”. IEEE Trans. Robot. 35, 286–293.

[8] Kumar, A. A., Antoine, J. F., & Abba, G., 2019. “Input-Output Feedback Linearization for the Control of a 4 Cable-Driven Parallel Robot”. IFAC-PapersOnLine, 52(13), 707-712.

[9] Kumar, A., Antoine, J. F., & Abba, G., 2019. “Linéarisation à rétroaction partielle d'un robot parallèle commandé par cable”.

[10]Kumar, A., Antoine, J.F., & Abba, G. (2020, July, In Press). “Control of an Underactuated 4 Cable-Driven Parallel Robot using Modified Input-Output Feedback Linearization”.

[11] Isidori, A., 2013. Nonlinear control systems, Third edition, softcover reprint of the hardcover 3rd edition 2000. ed, Communications and control engineering. Springer, London.

[12] Kumar, A. A., Antoine, J. F., Zattarin, P., & Abba, G., 2019. “Workspace Analysis of a 4 Cable-Driven Spatial Parallel Robot”. In ROMANSY 22–Robot Design, Dynamics and Control (pp. 204-212). Springer, Cham.

A synchronized-HRC pick and place application using an intelligent vision system

Rainer Müller
ZeMA -Zentrum für Mechatronik und Automatisierungstechnik gemeinntzige GmbH, Saarbrücken, Germany

Nishant Ketan Gajjar
ZeMA -Zentrum für Mechatronik und Automatisierungstechnik gemeinntzige GmbH, Saarbrücken, Germany

Ahmad El Masri
ZeMA -Zentrum für Mechatronik und Automatisierungstechnik gemeinntzige GmbH, Saarbrücken, Germany

Khansa Rekik
ZeMA -Zentrum für Mechatronik und Automatisierungstechnik gemeinntzige GmbH, Saarbrücken, Germany

***Abstract*—In recent times, automation has been one of the most important aspects of industrial applications. Collaboration between the humans and robots has been a key factor for the development of industries of the future where mutually, humans and machines, can work and carry out important tasks together. The focus of this work is to create a method for Human-Robot Collaboration (HRC) application, with the goal of setting up robots with certain safety measures in such a way that it actively supports a human completing it. The paper starts with an introduction to synchronized human-robot collaboration and the safety aspect of the system. It is then elaborated by an extensive study on image processing and application of computer vision used here in order to accomplish a given set of tasks.**

***Keywords*—Synchronized-HRC, Safety, Pick and Place, Computer vision**

I. Introduction

Human-Robot Collaboration as the name suggests, is the process in which both humans and robots work and interact with each other, without the need of any physical barriers between the work areas. This increases the productivity by combining human's ability to judge, react and plan, along with the robot's capability to do repetitive and risky tasks. It permits human operators to pay attention to operations with high added value or demanding high levels of adroitness, thus freeing them from monotonous or potentially dangerous situations [11]. With collaborative robots being introduced to assembly lines, manipulators are able to work in closer approximation with human operators to a point that tasks or stages in the assembly process have their roles overlap. Human and manufacturing errors are important factors to take into account, nonetheless, with the preprogrammed nature of manipulators it would be impractical to implement a collaborative system without the use of sensors that account for such errors and make the handling of the task robust.

For the collaboration between humans and robots, safety and interaction are the key factors for it to be successful. Before carrying out any given set of tasks, the safety of human beings and the equipment in use around the machines should be guaranteed as it is quite essential even if the collaboration is not effective [18]. HRC-capable robots or cobots [17] as we call them, offer great flexibility and can work without safety apparatus in many applications, however, a certain level of collaboration can still be achieved without necessarily using them. Physical interaction between humans and automatons is unavoidable in most of the cases and even looked-for when they share a common workplace or even work hand in hand. Some of our goals regarding the safety aspect of the whole systems include developing a sensor system that monitors workplaces and detects contact between humans and robots even before it happens [3]. Along with that the usage of manipulator ensures safety during interaction and cooperation with humans without conventional separating barriers and lastly, testing the whole arrangement, sensors and controllers for use in shared work areas.

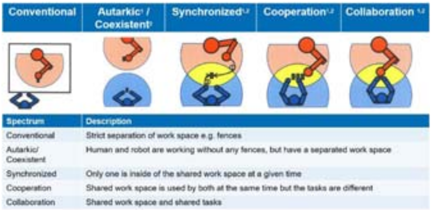
Using image processing along with robotics, is widely used in recent times and has added a lot of opportunities in productions. Considering the complications and monotony, using robots for this process is effective and shows to be enhancing productivity. Sorting is not an easy process rather it requires a number of complicated steps which include detail extraction, detection, training the data and at last processing. The set of objects that have to be sorted are pre-trained to the computer that is controlling the camera. This helps in the detection of these articles from the rest. The object locations and orientations are then sent to the controller, which in turn will then use this position to pick up the desired object and place it in a predefined location.

This work is divided into four main parts. First, the paper gives an overview about the safety aspect of human-robot collaboration in our trials. In the second part, the detailed theory behind the computer vision and image processing part in our setup is expounded, followed by a brief analysis of training and processing involved. Finally, the paper is concluded by giving a brief idea about the future works and the system in whole.

II. Safety aspects of the synchroniyed collaboration

A. *Synchronized-HRC*

Since the very introduction of modern robotics, a large amount of attention has been given to the safety pertaining to the Human as well as the robot. Conventional industrial robots are enormous, heavy and can move about at high speeds. These conditions make it necessary to prevent collisions between the robot and a human who may enter the robot workspace, to avoid harm to the individual. The approach set by the previous standard [1] to avoid such collisions or other occurrences that may result in injuries, was to establish an obligatory separation between human and robot workspaces, by detecting human interruptions in robot workspaces, and adjusting the robot actions accordingly.



Spectrum	Description
Conventional	Strict separation of work space e.g. fences
Autarkic/ Coexistent	Human and robot are working without any fences, but have a separated work space
Synchronized	Only one is inside of the shared work space at a given time
Cooperation	Shared work space is used by both at the same time but the tasks are different
Collaboration	Shared work space and shared tasks

Figure 2: Spectrum and definition of cooperation / Human and robot cooperation is not the same as Human-Robot-Cooperation [2,16]

There are several downsides that inhibit them from being extensively instituted to production environments. Even if the practical challenges of devising and deploying such systems have been surmounted, the operators' safety will always be the primary factor for achieving acknowledgment [5][15]. The present applications separate the human from the robots' working areas for the operators' safety to be guaranteed. Physical safeguards might not be practical in all cases when both humans and robots are working together, one might need to either use speed limitation or separate the workstations. It must be made sure that the robot is either constantly aware of its surrounding or that it has been programmed in a way that it does not create any conflicts with the cooperating human.

The general idea of the whole system is to prevent any injuries that might occur when both the human and the robot are working together. That being said, the human can continue working on one of the workspaces while the robot is working on another. Once the human has completed the task, they can change the cell in which they are working and go to another one. Simultaneously, the robot can search for the assembled part in the workspace where the human was working before and if it is completed, it can then move to grip the finished part. The safety system should ensure that under no circumstances a collision should occur. The sensors should check if an individual is in the cell and communicate the same information to the controller. If there is a human present, the robot should completely stop moving until the operator has left the cell.

B. *The system's safety aspects*

The system being discussed in this paper is built around KUKA KR6 [8] with a KUKA KRC4 Compact controller [9] and for it to work, there must be exactly one safety interface in use. There are many different interfaces on the controller, but the one that is being used for the safety of the system is the X11 [10] interface which is discrete and non-Ethernet based. Light curtains, safety laser scanner and a safety mat are connected to the X11 interface with the use of a safety PLC. Light curtains [13] (Figure 4: Light curtains emits a 2D protective grid which is used to detect human obstructions [13].) span a two-dimensional safety surface in front of the danger zone. Breaching the safety surface leads to a warning signal or an immediate machine stop. It consists of two main parts: a transmitter and a receiver. The transmitter comprises of multiple LEDs that sends pulses of infrared light to the receiver. Each pulse emitted are in sequence and are emitted at a specific frequency. The receiver pairs with the transmitter and expects the pulses in the right sequence and timing. It has to be made sure that the transmitter and the receiver are arranged such that persons or parts of the body are reliably detected when they enter the hazardous area and also in such a way that reaching under, over and around as well as moving the safety light curtain is prevented.

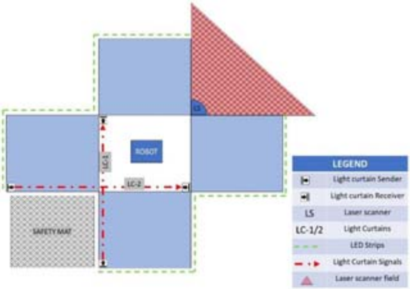


Figure 1: Basic outline of the system, the work cells are in blue surrounding the central robot cell, the safety mat and the laser scanner cover the areas where the human can stand.

Safety laser scanners [14] (Figure 3) are also electro-sensitive protection device (ESPE) which uses two-dimensional infrared laser beams to scan its environment. As soon as an object located in the protective field, created by the scanner, moves or obstructs the field, the scanner signals the detection by changing the signal at the safety output. It operates on the principle of time-of-flight that is it emits light pulses in regular, very short intervals, if the light strikes an object, it is reflected. The safety laser scanner receives the reflected light. The safety laser scanner calculates the distance to the object based on the time interval between the moment of transmission and moment of receipt. Safety mats are pressure-sensitive safeguarding equipment that are designed to detect presence of people on the sensing surfaces. They have two conductive hardened steel plates that are held apart by non-conductive compressible separators.

Figure 4: Light curtains emits a 2D protective grid which is used to detect human obstructions [13].

Figure 3: Safety laser scanner spans a protective laser field in an area which detects human obstructions [14].

The system is constructed in such a way that all the above-mentioned sensors can work individually. However, all the sensors can work together so that if any of the safety sensor is

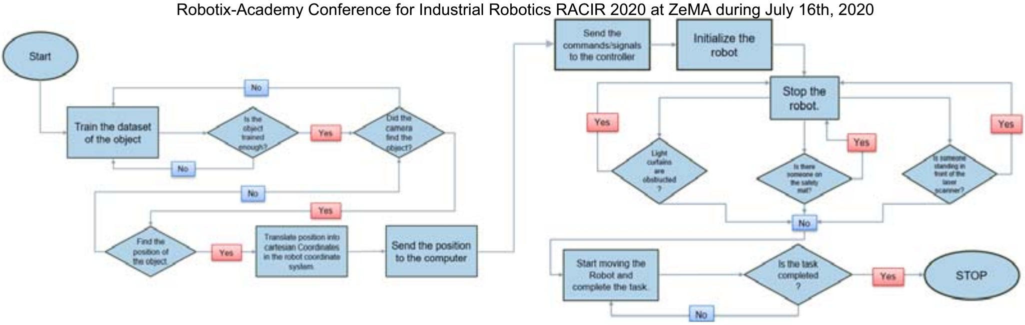

Start
Train the dataset of the object
Is the object trained enough?
No
Yes
Did the camera find the object?
No
No
Yes
Find the position of the object
Yes
Translate position into cartesian Coordinates in the robot coordinate system
Send the position to the computer
Send the commands/signals to the controller
Initialize the robot
Stop the robot.
Yes
Light curtains are obstructed ?
Yes
Is there someone on the safety mat?
Yes
Is someone standing in front of the laser scanner?
No
Start moving the Robot and complete the task
Is the task completed ?
Yes
STOP
No

Figure 5: Flowchart of the system in operation

obstructed, the whole system can take appropriate actions to make sure the environment is again safe to resume any stopped processes.

III. Usage of computer vision and image processing techniques

With the aid of computer vision and image processing techniques, the robots can see the results of human tasks and fulfil their own task in the process, reducing the effort needed for error checking. The techniques used will be abstracted for the purposes of this paper. The specific application of computer vision with robotics that is covered here is scanning a workbench for products that have been assembled by human operators and examining false assemblies from finished products. The position and orientation of the detected product are calculated then given as vectors, which are translated into coordinates for the manipulator to move to and grasp the object, which in turns moves to a predetermined location. This is essentially pick and place application.

A. Methodology

The test bench is a preassembled demonstration station with four sections in which a KUKA KR-SIXX 900 was installed in the middle of this station. The robot is controlled using the KUKA KRC4 Compact robot controller. The KR-6 was fitted with an attachment at the flange which allows for a depth camera to be mounted. The camera used for the application is an Intel RealSense D435i [6]. It has universal shutter with a 3µm x 3µm pixel size and has an Inertial Measurement Unit (IMU) inbuilt. It uses Active IR Stereo technology to obtain the depth data. Other than the depth stream, an RGB stream is also present which is used by our pre-trained model to find the required object.

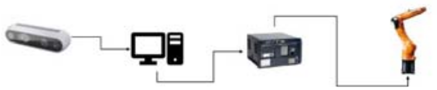


Figure 6: The pick and place application system components

- Given any set of objects on a bench within the view of the camera and the reach of the manipulator, an object or group of objects that have been pre-taught to the computer controlling the camera are detected from the rest:
  1. The object locations and orientations are given as vectors describing translation and orientation in the coordinate system of the camera
  2. The position vectors are then translated to the coordinate system of the controller in order to know the position the gripper must take in order to grasp the item properly
- Once the position has been received by the controller, it will then use this position to pick up the desired object and place it in a predefined location.

The whole operation can be divided into four main components – Feature extraction, Object detection, Training and Processing.

1) Feature extraction

Feature plays a very important role in the area of image processing. These methods are employed to get characteristics that will be useful in sorting and recognition of images. Feature extraction illustrates the pertinent shape data contained in an object model so that the task of categorizing the object is made easy by a formal process. In image processing, feature extraction is a distinct way of dimensionality reduction. The main goal of feature extraction is to acquire the most significant information from the original data and represent that information in a lower dimensionality space. It is mainly divided into two general stages: Detail selection and Classification.

To understand the context of feature extraction it would be helpful to provide a brief synopsis of stereoscopic vision. Stereoscopic vision is the ability to infer depth using two visual inputs that overlap the same area. This is the same ability that humans have on an intrinsic level when viewing objects with both eyes. The reason a second visual input, such as an eye or camera, allows for an image of depth is because stereoscopic vision takes advantage of several unique geometrical properties, such as epipolar geometry, in order to resolve ambiguity of depth, creating an accurate estimate of depth by use of triangulation. This essentially means that any point in the view of the depth camera will be encoded into a depth value associated with a pixel. This pixel will be a projection of 3D space onto a 2D depth map.

First the camera is calibrated on the flange of the robot using the 4-point method. Here four points belonging to the same plane are considered. The coordinates of these points are read in pixels from the camera, also the real coordinates with respect to the robot are read by jogging the robot to each point. Finally the transformation matrix from pixel to millimeter is calculated and used in what follows.

When the depth of an image is analyzed, the farther an object gets, the redder it would appear in the image. This is a visualization of a depth map where the values are normalized to be represented by visible colors, with blue being closest and red being the farthest. The actual values for each pixel are stored as a float representing distance in millimeters. In a semantic Segmentation of the image, each object group is assigned its own color so that visual cues can be used by human operators. What happens as a result is that any segmented object can be taken and its depth from the camera can be found using the same pixel range used in the segmented view. From there, a 3-Dimensional representation of the scene captured can be reconstructed in the coordinate system of the camera. Taking this 3-Dimensional map generated by the camera, We take the pose of the camera, represented by $C = [X\ \ Y\ Z\ \alpha\ \ \beta\ \gamma]^T$ the coordinate system of the manipulator. This information is then used in order to translate the points captured by the depth map into Cartesian points in the coordinate system of the manipulator. The regions that are of relevance (i.e. the detected objects) will have their position translated to the coordinate system of the manipulator as Cartesian points, which the manipulator can move to in order to grasp the desired object.

2) Object detection

Object Detection is a subset of computer vision and one of the most important part of it. As the name suggests, it is an automated method for locating objects of interest in an image with respect to the background. The key concern for object detection is that the number of objects in the foreground can differ throughout the whole image. The problem can be described as a labelling problem based on models of established objects. Given an image containing one or more objects of significance and a set of labels relating to a set of models known to the system, the system is supposed to designate correct labels to areas in the image. To overcome this, we have used a tradition computer network method of convolutional neural network (CNN). Convolutional neural network (CNN) is a class of deep, feed-forward artificial neural network that has been utilized to produce an accurate performance in computer vision tasks, such as image classification and detection [12]. CNNs are like traditional neural network, but with deeper layers. It has weights, biases and outputs through a nonlinear activation. The neurons of the CNN are arranged in a volumetric fashion such as, height, width and depth.

For this, most of the implementation is covered using OpenCV [4] and a machine learning library geared towards computer vision called ImageAI. The library takes in a set of images that have been pre-labelled with correct objects, where object names are held by strings, and it creates a dataset from which it can learn how to differentiate the desired object from a background. After using transfer learning to teach a pre-trained model, the objects are then detected, and that pixel range is passed to the depth stream so that the appropriate regions of the camera's view are calculated for coordinates. The main reason why OpenCV and ImageAI were used is because they allow for generalization. This means that given any set of objects, an operator can take a video stream or set of pictures that have been labelled and allow the program to train on the images in order to separate the desired object from the background when in practice. The RGB stream of the camera was used for this part as it deemed the only portion for learning relevant, because there is much more information to process when it comes to point clouds, and the point cloud that was being received had too much noise to produce an accurate geometric model of the object considering the size and distance from the camera.

3) Training and processing

Before working with an image classification model, we need to train it by showing many images of objects of interest. First, the objects are arranged on the table along with other insignificant objects in various positions to account for a large number of angles. Then, tens of pictures are taken from as many angles as physically possible with an emphasis on views from above since this is the most common viewpoint from the camera (Figure 7).

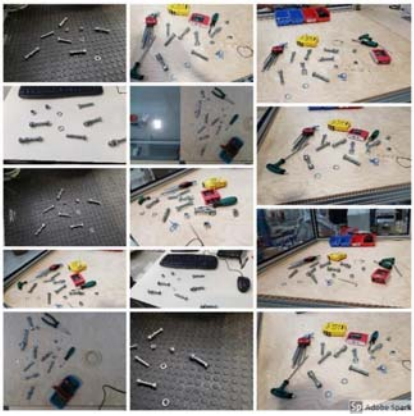


Figure 7: Training picture with different background, angle, and layout configurations

The objects are then rearranged with variations in the background and the above process is repeated for several times. Next, the images are all sorted into a folder and the objects needed to be detected are outlined and labelled in a software which allows users to label an image in a GUI and save the output as a corresponding csv file describing the label and its pixel range (Figure 8).

The dataset is then passed through a CNN [7] in order to train a model to identify the correct objects of interest. Once the model is trained, the camera stream starts, and the objects are then detected in the RGB stream. The corresponding points' depth is then taken and used to calculate the pose of the object. After the depth detail extraction and detection have been considered, the information is then translated into Cartesian Coordinates in the robot coordinate system. The robot then uses this information to move to the location, pick up the desired object, and then place it into predetermined bin locations.

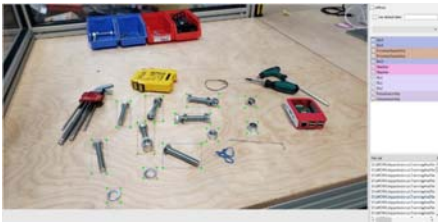


Figure 8: Significant objects are outlined and labelled to train the dataset correctly.

IV. Conclusion

The work presented herein explored how the collaboration between humans and robots has been a key factor in the development of recent productions. The paper tackles the goal of establishing robots with a variety of safety measures in such a manner that it diligently assists a human in accomplishing it. It shows how a certain amount of collaboration can still be attained without essentially using robots which are manufactured with the primary goal of maintaining human's safety. A pick and place application which involves both humans and the robot working together with the use of a specific camera technique and image processing has been described later. Currently, the application involves the robot to clutch the object from a 2D plane while in the future the concept would be developed to grasp the objects from a 3D environment. The future works also includes the expansion of the system to enable the robot to detect objects which are arranged in highly amorphous manner and then pick them up individually, also known as Bin-picking [19], and also to improve the measurement and gripping concept.

References

[1] ISO/TS 15066:2016, "Robots and robotic devices - collaborative robots," International Organization for Standardization, Standard ISO/TS15066:2016, Feb. 2016.

[2] Wilhelm Bauer, Manfred Bender, Martin Braun, Peter Rally, and Oliver Scholtz (2016). Leichtbauroboter in der manuellen montage– einfach einfach anfangen. Erste Erfahrungen von Anwenderunternehmen, pages 1–63, 2016.

[3] Julia Berg, Daniel Gebauer, Gunther Reinhart,Method for the evaluation of layout options for a human-robot collaboration,Procedia CIRP,Volume 83,2019,ISSN 2212-8271.

[4] Bradski, G. (2000). The OpenCV Library, Dr. Dobb's Journal of Software Tools.

[5] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero and J. PérezOria, "Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments," in *IEEE Access*, vol. 5, pp. 26754-26773, 2017, doi: 10.1109/ACCESS.2017.2773127.

[6] INTEL, D435i Camera. Retrieved May 18, 2020, from https://www.intelrealsense.com/depth-camera-d435i/.

[7] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in neural information processing systems, 2012.

[8] KUKA AG., KR 6 R900 sixx. Retrieved May 21, 2020, from https://www.kuka.com/en-in/products/robotics-systems/industrial-robots/kr-agilus.

[9] KUKA AG., KRC4 Compact. Retrieved May 22, 2020, from https://www.kuka.com/en-in/products/robotics-systems/robot-controllers/kr-c4.

[10] KUKA AG., X11 Safety Interface. Retrieved May 18, 2020, from https://www.kuka.com/en-in/products/robotics-systems/software/hub-technologies/kuka_safeoperation.

[11] Lewis M., Sycara K., Walker P. (2018) The Role of Trust in Human-Robot Interaction. In: Abbass H., Scholz J., Reid D. (eds) Foundations of Trusted Autonomy. Studies in Systems, Decision and Control,vol 117. Springer, Cham.

[12] Sharma, Kartik & Thakur, Nileshsingh. (2017). A review and an approach for object detection in images. International Journal of Computational Vision and Robotics. 7. 196. 10.1504/IJCVR.2017.081234.

[13] SICK, Safety light curtains. (n.d.). Retrieved May 21, 2020, from https://www.sick.com/dk/en/opto-electronic-protective-devices/safety-light-curtains/detec/c4c-ea19530a10000/p/p308169.

[14] SICK, Safety lasers scanner (n.d.) .). Retrieved May 21, 2020, from https://www.sick.com/ca/en/opto-electronic-protective-devices/safety-laser-scanners/microscan3/c/g295657.

[15] J. T. C. Tan, F. Duan, Y. Zhang, K. Watanabe, R. Kato and T. Arai, "Human-robot collaboration in cellular manufacturing: Design and development," 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, 2009, pp. 29-34, doi: 10.1109/IROS.2009.5354155.

[16] Stefan Thiemermann(2005). Direkte Mensch-Roboter-Kooperation in der Kleinteilemontage mit einem SCARA-Roboter: Zugl.: Stuttgart, Univ., Diss., 2005, volume 411 of IPAIAO- Forschung und -Praxis. Jost-Jetter, Heimsheim, 2005.

[17] Federico Vicentini, Terminology in safety of collaborative robotics, Robotics and ComputerIntegrated Manufacturing, Volume 63, 2020, 101921, ISSN 0736-5845.

[18] Villani, Valeria & Pini, Fabio & Leali, Francesco & Secchi, Cristian. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Mechatronics. 10.1016/j.mechatronics.2018.02.009.

[19] Schyja A, Kuhlenktter B. Virtual Bin Picking-a generic framework to overcome the Bin Picking complexity by the use of a virtual environment[C]//Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH), 2014 International Conference on. IEEE, 2014:133-140.

Hybrid Workstations: A Simultaneous Planning Method for Economic-oriented Selection between Industrial and Collaborative Robots

Christopher Schneider
Product Management
Yaskawa Europe GmbH
Allershausen, Germany
christopher.schneider@
yaskawa.eu.com

Thomas Suchanek
Safety
Yaskawa Europe GmbH
Allershausen, Germany
thomas.suchanek@
yaskawa.eu.com

Martina Hutter-Mironovova
Commercial Customer Service
Yaskawa Europe GmbH
Allershausen, Germany
martina.hutter@
yaskawa.eu.com

Francisco Hernandez
Chair for Ergonomics and Innovation
Technical University Chemnitz
Chemnitz, Germany
francisco.hernandez@mb.tu-chemnitz.de

Mohamad Bdiwi
Robotics Systems
Fraunhofer Institue for Machine Tools
and Forming Technology
Chemnitz, Germany
mohamad.bdiwi
@iwu.fraunhofer.de

Matthias Putz
Machine Tools, Production Systems
and Machining
Fraunhofer Institute for Machine Tools
and Forming Technology
Chemnitz, Germany
matthias.putz@iwu.fraunhofer.de

Abstract— **Current Human-Robot Interaction (HRI) planning methods focus firmly on the technical side while neglecting the economic complexity. Simple financial calculations are insufficient to counterbalance the occurring uncertainties. Furthermore, comparisons to fenceless industrial robots are ignored, which leads to incomprehensive solution space. Therefore, we present a planning tool to determine the economic-optimal fenceless robot-based system under consideration of all relevant factors.**

Keywords—Collaborative Robots, Hybrid Workstations, Human-Robot-Collaboration, System Design, Cell Planning

I. Introduction

Collaborative robots (cobots) have been a vast hype topic as associated with Industry 4.0 and digitization. Promises such as easy integration and fast return on investment (ROI) in the early phase of the upcycle have been proven to be too idealistic when the first projects faced the requirements of safety. The resulting operating speeds and the correspondent cycle times were often insufficient, and many cobots ended up behind a safety fence to ensure the required profitability [1]. In such cases, industrial robots with external safety would have been a suitable alternative but were not considered in the first place. The problem lies in the applied procedure: by setting a cobot as the prerequisite, planning of task, tooling, and cell need several iterations until the wished results are reached (see figure 1).

In contrast, classic planning procedures focus on the cell at the beginning, including system concept, workflow, cycle time, and profitability. Safety engineering, as a core element in human-robot-interaction (HRI), is already conceptualized in the early planning phases. Based on this primal concept, appropriate tooling is designed for the specific task to meet the requirements of safety, cycle time, and flexibility. Lastly, the decision about robot technology is executed. However, to make the philosophy of cobots come true, both industry and research undertake various attempts to break down integration obstacles.

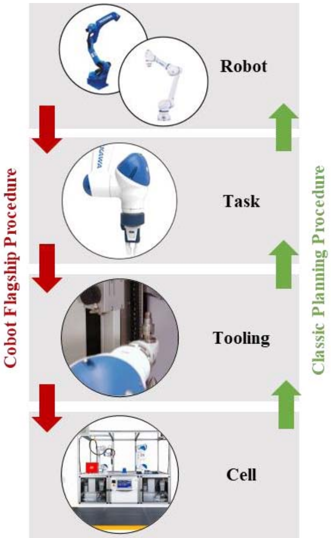


Figure 1: Planning Sights

To understand this in more detail, the different requirements on planning and commission a collaborative workstation must be considered. Those include the needs of industry and standards bodies on the practical side as well as the requirements of the single academic professions, such as factory planning, industrial engineering and economics. Industrial end users and system integrators aspire to identify the most profitable solution, independent on the utilized technology (industrial robot or cobot). Now, after the first Industry 4.0 flagship projects are completed (upcycle phase), more pragmatism arrives in the world of human-robot-interaction, which makes the utilization of a cobot not mandatory anymore. Therefore, comprehensive planning methods are needed that allow parallel comparative planning of both technology alternatives. Consistency of the interrelations between all economic-relevant figures is the essential premise. For the sake of cost reduction, those methods must be simple to use, so that occurring efforts in the early rough planning phase can be reduced to a minimum. High predictability of possible deviation factors and the final economic result is required to enable reliable investment decisions and prevent follow-up costs. Therefore, the aim of this paper is to develop a comprehensive planning methodology, that considers technical and economic aspects.

II. Methodology and Approach

A. *DMADV Cycle*

The previously described methodological problem requires a comprehensive solution by satisfying the needs of the respective stakeholders and disciplines. Although various methods exist, they still lack in terms of economic planning consistency and require, therefore, optimization. Lean Six Sigma (LSS) as a standard improvement toolset can be applied to this problem in slight modification. Initially, the DMAIC cycle is the core LSS principle, that consists of the steps Define, Measure, Analyse, Improve, and Control [2]. For designing non-manufacturing-based processes, innovations, or services, the DMADV cycle is utilized. This modified version replaces the last steps by Design and Verify. Fundamentally, an LSS optimization project begins with the definition of the procedure, its stages, and the venture itself. In the next phases, the required variables are measured and analyzed, followed up by designing a new service or in this case methodology. In the final verification phase, developed results are proven regarding their applicability to practice [3]. Verification is without the scope of this paper. In the following paragraph, the approach for this paper is going to be explained in further detail, which is synonymous with the define phase.

B. *Approach (Define)*

As the primary target, the development of an economic assessment method to plan industrial and collaborative robots comparatively is set. Therefore, technical enabling steps that directly or indirectly influence the economic key figures are collected, and the economic key figures are distilled by reviewing existing planning methods. Hence, the economic structure and interrelations can be concluded to an overall scheme. Scientific and methodological gaps are revealed by analyzing to what extent the previous planning methods consider all these factors. In the next step, several surveying methods are examined regarding their occurrence in the state of research and their suitability. Based on that information, a comprehensive planning method is concluded, that includes both technical and economic aspects as well as the alternatives collaborative and industrial robot. As a basic framework, the author's previous human-robot-collaboration (HRC) planning method is used and supplemented by the findings of this research [4]. To outline the current state of the art and research in fenceless production planning, scientific and practical methodologies from recent publications (2017-2020) have been reviewed [5–12].

III. Measure

A. *Collect Technical Enablers*

To understand the influences of engineering activities on the economic outcome, the standard foundation, as well as the technical sights on the topic, need to be reflected. By reviewing the previously mentioned papers, the reoccurring design fields system, process, safety, layout/ logistics, human/ robot, and gripper/workpiece can be revealed, which are explained as follows.

Generally, system engineering begins with the analysis of the current state in manual execution and the definition of automation targets. Therefore, the considered production level (cell, line, etc.) and the project phases are defined. To define the target state, potentials in improvement and automation must be collected and considered regarding their feasibility. Throughout the whole planning project, different technical alternatives are assessed with various key performance indicators (KPI's) and the net present value as an economic target figure.

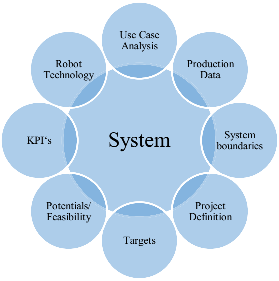


Figure 2: Activity Field "System"

At the process level, the manual execution can be described with various modeling methods by subdividing the chain into single subprocesses. Each process step can be modeled regarding the material and the information flow. Based on this process map, possible wastes according to the Lean framework TIMWOOD are identified as the first information base for automation potentials. Resulting automation concepts are assessed regarding the automation grade and the respective tasks and interactions between human and robot.

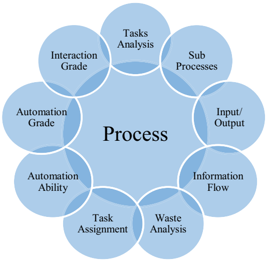


Figure 3: Activity Field "Process"

When planning robotic automation, logistics requires adjustment to enable material handling. In the manual state, the material provision is optimized for the anthropometry of humans. In hybrid workstations, however, both interaction partners must be able to handle the workpieces. Together with the gripper and vision system, the logistics build the fundament to enable the robotic operation. The resulting material flow is used to conclude the layout by incorporating restrictions regarding interrelations and space.

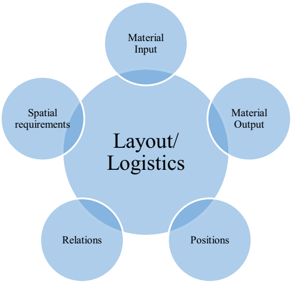


Figure 4: Activity Field “Layout/ Logistics”

The standards bodies set the requirements for robot systems, that must be matched to get the individual cell CE marked. Therefore, several standards must be considered already in an early stage of the system design. Fundamentally, the overall process of risk assessment and risk reduction is described by DIN EN ISO 12100 [13]. Further standards define the requirements on the robot system, the safety controls, and external safety equipment. For industrial and collaborative robots, DIN EN ISO 10218 -1 and -2 defines technical requirements. This standard is divided into two parts: while the first part is designed for safe design and building of industrial robots, the second part also addresses safety requirements regarding human-robot collaboration [14, 15]. ISO/TS 15066 specifies HRC in much more detail. It describes the operating modes safety-rated monitored stop (SRMS), hand-guiding (HG), speed & separation monitoring (SSM) as well as power & force limiting (PFL) [16]. Hence, it can be seen that HRI is possible with both industrial and collaborative robots. The mode selection for the individual application implies mode-dependent operating speeds, that must be included in the economic evaluation. For PFL, force and pressure measurements are required to identify the maximum allowed collaborative speed. In contrast to the widespread opinion that a cobot can always operate in 250 mm/s, the appropriate speed must be determined for the respective robot-gripper-workpiece combination for the specific task [16]. Safety controls are specified in DIN EN ISO 13849, especially regarding the required SIL/ PL levels [17]. Connected external safety devices, such as laser scanners, are described in DIN EN 61496 for general requirements [18]. DIN EN IEC 62046 and DIN EN ISO 13855 specify the design requirements further, especially regarding the zone design and distances that need to be maintained to ensure the operator’s safety [19, 20].

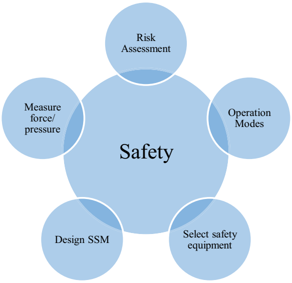


Figure 5: Activity Field “Safety”

From an industrial engineering perspective, movements and ergonomics are centralized. On the one side, the walking ways and body movements of the operator must be assessed and planned, i.e. with MTM or REFA. On the other side, the robot paths can be simulated, i.e. with offline programming simulation environments. The sequencing of both movements and the movement design under consideration of possible interactions is the main task of IE. Furthermore, the safety requirements of the standards must be realized within the scope of occupational health.

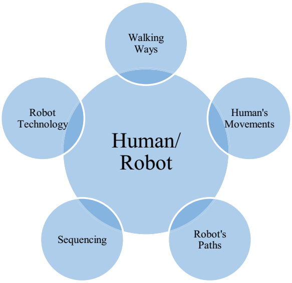


Figure 6: Activity Field “Human/Robot”

To ensure handling of the produced workpiece, the gripper strategy and model must be selected. Therefore, the payload chart of the robot can be used to identify the allowed handling weight depending on the lever. For collaborative applications, ISO TS 15066 demands rounded gripper edges to reduce the potential collision forces. Furthermore, the gripping power must be reduced to 140N [16]. Due to this reduced force, form closure principles should be considered for handling. Besides the gripper, also the workpiece must be assessed regarding its influence on the feasibility of the automation project. By executing the pressure and force measurements, the workpiece’s suitability for HRI can be assessed. Sharp edges or corners are usually an exclusion criterion due to the high injury potential.

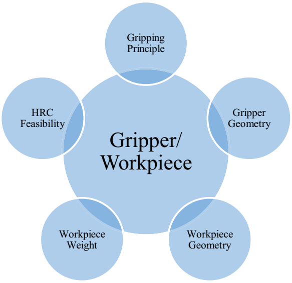


Figure 7: Activity Field "Gripper/ Workpiece"

B. *Distill Economic Key Figures*

After the main technical enablers have been identified and described, the economic side of HRI cell planning is considered next. The net present value (NPV) is utilized as an economic optimization factor, that consists of the elements initial investment (I_0) as well as the time-dependent (t) cash-out flows (COF) and cash-in flows (CIF). To identify the optimal robot technology, two separate NPV's are calculated for the alternatives "Industrial Robot" (IndRob) and "Collaborative Robot" (ColRob). By comparing both values, an economic-based decision about the advantageousness can be made [21, 22]. In contrast to the I_0 and COF values, that are equipment-dependent, the individual CIF's require more effort for data collection. Therefore, the composition of this factor is deeper considered in this paper. Fundamentally, positive cash flows can be expressed as the sum of the labor release grade (LRG) and the annual output deviation (ΔO). The LRG is the percental time slice that the operator is relieved from its task due to automation. Within this gained time, other, more value-adding activities can be executed. ΔO, on the other side, describes the change in produced units that come along with automation. For the calculation of both factors, the execution times (ET) between the automation alternative and manual (M) execution are set in ratio to each other.

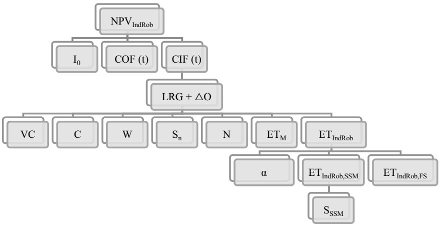


Figure 8: Tree diagram of the NPV composition for industrial robots

For exact cycle time identification, the single operating modes in fenceless production must be observed for each alternative. It is assumed that external safety devices (i.e., laser scanners) are used to enable the robot to switch between operating speeds according to the operator's proximity. Important to consider are the differences in available modes between industrial and collaborative robot regarding the proximity to the operator. Industrial robots utilize the operating modes full-speed (FS, operator absent), speed and separation monitoring (SSM, reduced speed, middle proximity), and safety-rated monitored stop (SRMS, no speed, high proximity). Collaborative robots, on the other side, replace the SRMS with PFL, which enables the cobot to continue working at collaborative speed even in high proximity to the operator. As the ratio between all three operating modes for each individual alternative, the human-robot interaction grade α is used. The execution times in mixed operation can, therefore, be determined, by setting the single execution times for the speed levels (S_{FS}, S_{SSM} and S_{CS}) in ratio to each other. To identify the speed-related execution times (i.e. via simulation), the feasible speeds need to be determined first. While the SSM speed can be determined via calculation, the collaborative speed must be identified by measurements (for clamping situations). However, to complete the calculation the CIF's, the following factors need to be considered as well: the annual labor cost of the operator C, the value creation per workpiece VC, the number of batches N, the number of workpieces per batch S_n and the annual working time W.

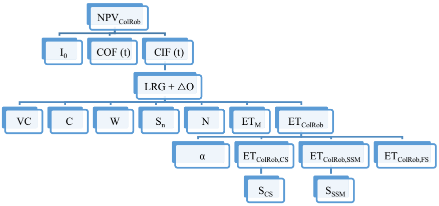


Figure 9: Tree diagram of the NPV composition for collaborative robots

IV. Analyze

To calculate the NPV's, all the described data needs to be gathered. This information can be fundamentally subdivided into values that are given and ones that need to be determined. Given values depend on the individual production, customer, or equipment that is used. Determination of values can be executed via calculation or various surveying methods. In this paper, only the techniques are considered, that are required for data collection (right column in figure 10).

The various papers have been assessed regarding their methodological coverage of the single economic factors (see figure 11). As a result, it can be seen that there is no planning method yet, that enables the user to comprehensively plan fenceless workstations neither to compare industrial and collaborative robots to each other. The methodological variety was evaluated for each method. To determine manual execution times, the methods MTM, REFA, and digital human models are conventional. In contrast, the robot times are either calculated or simulated with offline programming software, MTM-based robot time systems, or virtual sensors. As an input value, the actual operating speeds are required. Speed and separation monitoring can be designed with calculations or virtual sensors. The collaborative speed can be calculated or evaluated with virtual collision models or virtual sensors. Lastly, the interaction grade can be determined with a parallel robot and human digital modeling or within virtual reality environments.

	Customer-related	Equipment-related	Requires calculation	**Requires methods**
I_0	○	•	○	○
CIF (t)	○	○	•	○
COF (t)	○	○	•	○
LRG	○	○	•	○
C	•	○	○	○
△O	○	○	•	○
VC	•	○	○	○
N	•	○	○	○
S_n	•	○	○	○
W	•	○	○	○
ET_{IndRob}	○	○	○	●
ET_{ColRob}	○	○	○	●
ET_M	○	○	○	●
$ET_{IndRob,FS}$	○	○	○	●
$ET_{IndRob,SSM}$	○	○	○	●
$ET_{ColRob,FS}$	○	○	○	●
$ET_{ColRob,SSM}$	○	○	○	●
$ET_{ColRob,CS}$	○	○	○	●
S_{SSM}	○	○	○	●
S_{CS}	○	○	○	●
α	○	○	○	●

● most relevant • considered ○ not considered

Figure 10: Data sources of different variables

Even though Schröter and Zhang et al. described all factors in their methodologies, practical validity is not given. Some of the used methods are still in the theoretical concept phase and are not accepted by industry and, therefore, not applicable. A critical assessment of the methodological coverage is explained as follows. Although those methods have been presented in theory, their readiness for industrial practice is questionable. Simulations with virtual reality, sensors, or collision models are possible but rarely accepted in the industry. Lacking accuracy and mismatching with practical results lead to unsharp economic outcomes, that endanger the realization of automation projects, which is explained further as follows. Firstly, digital human simulation requires a tremendous modeling effort, especially when synchronizing the motions with a robot. Therefore, planning costs are higher compared to classical fenced-in robot cells, where the operation speeds and execution times can be simulated and optimized without external influencing factors. Secondly, virtual sensors systems are available but still must be designed according to the use case. Using virtual I/O's (inputs/ outputs) is already possible in offline programming to switch between operating speeds and to simulate different simulations. As data input, the determined operating speeds according to the zone definition and design are a prerequisite, though. Lastly, virtual collision models are not that accurate yet to replace the force and pressure measurements that are required by ISO TS 15066. The model would have to cover the whole system behavior, consisting of a robot, gripper, workpiece, and collision surface for the collision in free space and the clamping scenario. Factors such as pose-dependent kinematic behavior, rigidity, and yielding of all single components would need to be simulated. The actual measurements that are described in ISO TS 15066 are not reviewed academically yet.

	[5]	[7]	[10]	[6]	[12]	[8]	[4]	[9]	[11]	
	Bouchard, 2017	Gopinath et.al., 2018	Selevsek et.al, 2018	Malik et.al., 2017	Ore et.al., 2017	Michalos et.al., 2018	Schneider et.al., 2018	Schröter, 2018	Zhang et.al., 2017	**Proposed Approach**
IndRob	○	•	○	○	•	•	•	•	•	•
ColRob	•	○	•	•	○	○	•	•	•	•
ET_{IndRob}	○	○	○	○	•	•	•	••	•	••
ET_{ColRob}	○	○	○	•	○	○	•	••	•	••
ET_M	○	○	○	•	•	•	••	•	•	••
$ET_{IndRob,FS}$	○	○	○	○	○	○	•	••	•	••
$ET_{IndRob,SSM}$	○	○	○	○	○	○	○	••	••	••
$ET_{ColRob,FS}$	○	○	○	○	○	○	•	••	••	••
$ET_{ColRob,SSM}$	○	○	○	○	○	○	○	••	••	••
$ET_{ColRob,CS}$	○	○	○	○	○	○	•	••	••	••
S_{SSM}	○	○	○	○	○	○	○	•	•	••
S_{CS}	○	○	○	○	○	○	○	•	••	••
α	○	○	○	•	•	•	○	○	••	••

•• deeply considered • considered ○ not considered

Figure 11: Methodological Coverage

V. Design

Finally, the technical and economic dimensions are merged to conclude a comprehensive planning method. In classic factory planning, a production system is planned in different detailing steps. This procedure can also be seen when comparing the planning as mentioned above methods. Therefore, the three phases analysis, concept, and planning are used. As a fundamental framework, the author's previous planning method [4] was used. Technical enablers and the surveying of economic values that have not been considered in the earlier version were supplemented. To mark, where economic key figures are gathered, the respective planning field is colored orange. In the following, a planning paradigm with methodological suggestions is presented.

The analysis phase begins by collecting information about the use case. Firstly, the production is described by gathering information about annual operator cost, value creation per workpiece, number of batches, batch size, and annual working time. Furthermore, the process and its subprocesses, as well as the current layout, are documented. For a better understanding of the material flow, the material in-feed system, the handled range of workpieces, and their geometries, as well as the material deposit, are described. Lastly, the execution times of the single subprocesses are measured with either MTM or REFA. Alternatively, digital human models can be used, although modeling is tremendously higher. In the next step, the system is described in further detail by narrowing down the considered scope, such as a single workstation, cell, line, or whole factory. The project phase is defined; for this paper, only phases rough planning to fine planning are considered, and commissioning is out of scope. Furthermore, the technical alternatives that are compared to each other are defined as well.

This paper focuses on the comparison between industrial and collaborative robots, but further alternatives could be possible too. Then, the potential and feasibility analyses are executed by analyzing possible wastes within the process first. Those improvement possibilities open up a first input for collecting automation possibilities, i.e., with a morphological case. By analyzing the workpiece regarding its geometry and the available space, the feasibility for HRI can be assessed. If the workstation is not suitable for direct human-robot-collaboration, i.e., due to sharp workpiece edges, then only an industrial robot with external safety is left for consideration. In the final step of the analysis phase, the pursued automation grade is defined, i.e., with levels of automation (LoA). Then, stepwise extensions of the functionality range are defined in respective integration levels. That means that such a project can be executed successive, by starting with integrating the basic functionality and then adding more features to the system, which can be used as milestones within the project charter. Those milestones need to lead to a defined target state that is defined from both the technical and economic side. To track the project progress, the single evaluation criteria are assessed with different KPI's, such as the net present value for the economic side, the overall equipment effectiveness (O.E.E.) or the accessibility for the technical side and the risk priority number (RPN) or the result of the rapid upper limb assessment (RULA) for the social side. An overview of the evaluation criteria for HRI projects is proposed in [21] and [22].

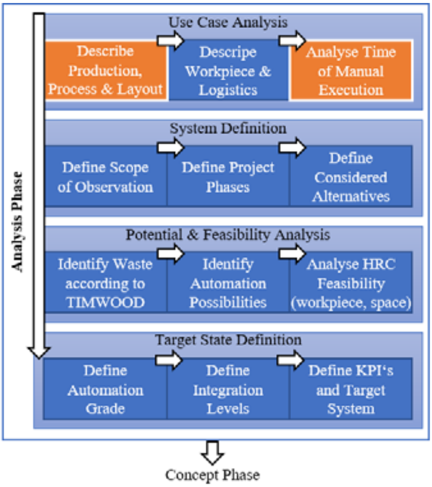

Analysis Phase
Use Case Analysis
Describe Production, Process & Layout
Descripe Workpiece & Logistics
Analyse Time of Manual Execution
System Definition
Define Scope of Observation
Define Project Phases
Define Considered Alternatives
Potential & Feasibility Analysis
Identify Waste according to TIMWOOD
Identify Automation Possibilities
Analyse HRC Feasibility (workpiece, space)
Target State Definition
Define Automation Grade
Define Integration Levels
Define KPI's and Target System
Concept Phase

Figure 12: Analysis Phase

After the analysis phase, a workstation concept is created, that already includes the most relevant planning items. In the basic concept, the planner defines the task division between human and operator, the single movements, and their sequencing. This step is closely linked with the previous feasibility assessment and the automation level definition. Then, the handling principle is planned by aligning the gripper with vision (if required) and the logistics systems. For the gripping, a suitable strategy (jaws, vacuum, etc.) and an available model is selected. For very specialized situations, customizations, such as jaw design, is required. To ensure permanent handling, the required holding forces are calculated, and the gripping strategy is adjusted if needed. As a next step, the logistic and material flow is analyzed, especially regarding material provision and depositing. Based on the possibilities to provide the workpieces in a robot-convenient way, a vision can be required. Therefore, a decision about the used technology (2D or 3D) and the installation (camera attached to the robot's flange, external) must be made. With this information, a rough concept of the racks and the camera is concluded.

As the next step, the layout is conceptualized. Therefore, all the single elements (material in-flow and deposit, robot, fixtures, etc.) are positioned first and then changed iteratively depending on the interrelations of the material flow. Furthermore, spatial restrictions are incorporated, such as traffic routes, safety areas, etc. Lastly, a safety concept is developed based on the layout variants. By identifying risk areas, i.e., clamping areas, appropriate measures can be conceptualized. By selecting the correct operation modes (PFL, SSM, SRMS), suitable safety equipment is concluded. The respective safety concept is developed in strong linking with the previous layout concept and is also an iterative process. To decide between different safety technologies (laser scanners, light curtains, camera, etc.), also the spatial restrictions of the layout must be considered.

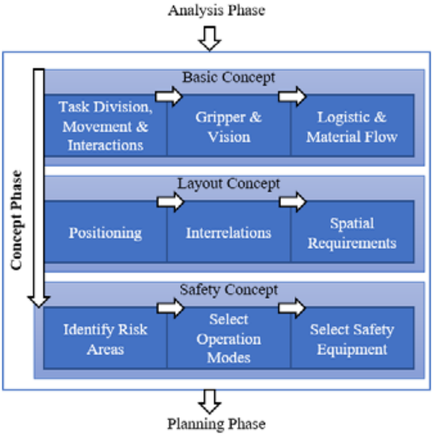


Figure 13: Concept Phase

Lastly, the workstation is planned in detail in the actual planning phase. At first, the gripper is designed in detail under consideration of the collaborative requirements of ISO TS 15066, the required gripping forces, and the gripping strategy. If needed, the vision system is set up and calibrated to detect the workpieces. Then, the logistics are designed by finalizing the logistics systems (i.e., flow racks) and calculating the required capacities. A final test up of all elements guarantees the functionality of the developed system. Secondly, the risk areas are assessed, and detail and the required safety zones are designed and implemented. That includes the SSM distances and stopping times dependent on the operator's proximity as well as the definition of different areas in the functional safety unit. For collaborative robots, the force and pressure measurement must be executed with a dedicated measurement device. These measurements are required for quasi-static and transient contact situations and deliver the allowed collaborative speed during operation. Then, the movements and interactions are planned by calculating and simulating the operator and robot movements. Appropriate methods are MTM, REFA, digital human models, offline robot programming, and simultaneous human-robot models. The advantage of simultaneous modeling lies in the precise determination of the human-interaction-grade α. For separate modeling, assumptions about the interaction phases are required, i.e., with Gantt diagrams. Finally, the layout is designed in a final stage by incorporating all restrictions that were collected during the previous phases. Layout optimization should be executed under the objective of cycle time and safety optimization. Therefore, the fine positioning is performed iteratively to match this optimization target.

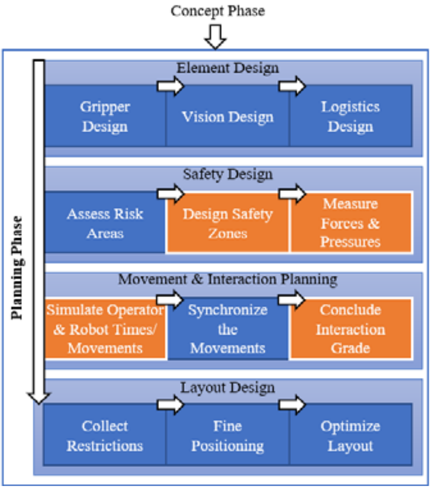


Figure 14: Planning Phase

VI. Results

This paper presents a planning framework to make an economic-based decision for identifying the optimal robot technology in fenceless production systems. Therefore, the technical enablers have been reviewed, that are required to develop such systems. Furthermore, an economic structure was presented that assists in calculating the net present values for the two alternatives industrial and collaborative robots. By reviewing existing practical and scientific planning approaches, it has been revealed that current methodologies lack the determination of all required factors to make a sound investment decision. Although some procedures describe all parameters, the surveying and determination methods are still insufficient regarding practical applicability. A planning methodology is presented that includes all the required steps to plan those systems and gather the required information. Fundamentally, the phases analysis, concept, and planning are proposed. For each substep, suitable methods have been reviewed and presented to give the planner a range of available tools.

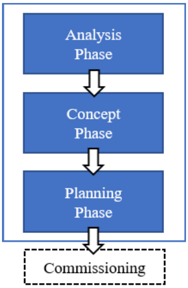


Figure 15: Overall Planning Framework

VII. Conclusion and Outlook

In this paper, we contribute to the development of an applicable planning method to streamline investment decisions between industrial and collaborative robots in fenceless production. To comprehend the exact mathematical relationships, another paper is currently prepared for publication. The considered factors focus primarily on those that are relevant for the cash-in flow determination. Since the initial investment (cost for equipment, training, commissioning, etc.) and cash-out flows (maintenance cost, service, etc.) are dependent on the use case, the determination of these figures was not described. In the future, practical tests are recommended to validate the practical applicability of this approach.

References

[1] W. Bauer, M. Bender, M. Braun, P. Rally, and O. Scholtz, *Lightweight robots in manual assembly - best to start simply!*, 2016.

[2] R. Shankar, *Process Improvement Using Six Sigma: A DMAIC Guide*. Milwaukee: American Society for Quality (ASQ), 2009.

[3] S. Lunau *et al., Design for Six Sigma + Lean Toolset: Implementing Innovations Successfully*. Berlin Heidelberg: Springer-Verlag, 2009.

[4] C. Schneider and H. Unger, Eds., *Methodik zur Grob- und Feinplanung von Arbeitsstationen zur Mensch-Roboter-Kooperation in hybriden Produktionssystemen*. Technische Universität Chemnitz, 2018.

[5] S. Bouchard, *Lean Robotics: A Guide to Making Robots Work in Your Factory*.

[6] A. A. Malik and A. Bilberg, Eds., *Framework to Implement Collaborative Robots In Manual Assembly: A Lean Automation Approach*: DAAAM International Symposium, 2017.

[7] V. Gopinath, F. Ore, S. Grahn, and K. Johansen, Eds., *Safety-Focussed Design of Collaborative Assembly Station with Large Industrial Robots*: Elsevier B.V., 2018.

[8] G. Michalos, J. Spiliotopoulos, S. Makris, and G. Chryssolouris, “A method for planning human robot shared tasks,” *CIRP Journal of Manufacturing Science and Technology*, vol. 22, pp. 76–90, 2018.

[9] D. Schröter, “Entwicklung einer Methodik zur Planung von Arbeitssystemen in Mensch-Roboter-Kooperation,” Dissertation, Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen, University Stuttgart, Stuttgart, 2018.

[10] N. Selevsek and C. Köhler, “Angepasste Planungssystematik für MRK-Systeme,” *Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF)*, vol. 1, no. 2, pp. 55–58, 2018.

[11] P. Zhang, S. Bauer, and T. M. Sontag, “Mensch-Roboter-Kooperation in der Digitalen Fabrik: Konzept zur Planung und Absicherung,” *Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF)*, vol. 1, no. 2, pp. 73–77, 2017.

[12] F. Ore, L. Hansson, and M. Wiktorsson, Eds., *Method for design of human-industrial robot collaboration workstations*, 2017.

[13] *Safety of machinery - General principles for design - Risk assessment and risk reduction*, 12100:2011-03.

[14] *Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots*, 10218-1:2012-01.

[15] *Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration*, 10218-2:2012-06.

[16] *Robots and robotic devices — Collaborative robots*, 15066:2016.

[17] *Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design*, 13849-1:2016-06.

[18] *Safety of machinery - Electro-sensitive protective equipment - Part 1: General requirements and tests*, 61496-1:2014-05;VDE 0113-201:2014-05.

[19] *Safety of machinery - Application of protective equipment to detect the presence of persons*, 2046:2019-03;VDE 0113-211:2019-03.

[20] *Safety of machinery - Positioning of safeguards with respect to the approach speeds of parts of the human body*, 13855:2010-10.

[21] U. Götze, M. Schildt, and B. Mikus, Eds., *Mensch-Roboter-Kooperationen – Interaktionsmodell der Zukunft?*: Verlag aw&I Wissenschaft und Praxis, 2015.

[22] B. Mikus, U. Götze, and M. Schildt, “Kooperation zwischen Mensch und Roboter – ein Beitrag zur nachhaltigen Produktion?,” *Der Betriebswirt*, pp. 25–31, 2016.

Practically Oriented Investigation of Sensitive Robots Regarding the Execution of Force Controlled Applications

1st Rainer Müller
Chair for Assembly Systems
Saarland University.
ZeMA - Center for Mechatronics and
Automation Technology gGmbH
Saarbrücken, Germany.
rainer.mueller@zema.de

2nd Ali Kanso
ZeMA - Center for Mechatronics and
Automation Technology gGmbH
Saarbrücken, Germany.
a.kanso@zema.de

3rd Marco Schneider
ZeMA - Center for Mechatronics and
Automation Technology gGmbH
Saarbrücken, Germany.
m.schneider@zema.de

***Abstract*—The use of sensitive robots is of great importance, especially since the introduction of human-robot and robot-object interactions. In addition to ensuring safe cooperation between human and machine, sensitive robot systems also offer the possibility of identifying their environment with the aid of their intrinsic sensors. This sensor technology, in the form of force and torque sensors, and the closely related control strategy, differs partial between different robot manufacturers. Furthermore there are numerous external sensors that enable a conventional robot to interact with its environment. This paper deals with a practical oriented comparison between the KUKA LBR iiwa and the Universal Robots UR 10e regarding their ability to perform sensitive or force controlled applications.**

***Index Terms*—Sensitive Robotic, KUKA LBR iiwa, Universal Robots UR 10e, Force-torque-sensor, Force Accuracy**

I. Introduction

Since the beginning of technical production, people have wanted to make production processes more effective and efficient (rationalisation). This development, up to the flexible automation demanded today, always has the same following goals in mind:

- Increase in productivity,
- Shortening of production times,
- Facilitation of human work,
- Reduction of costs,
- Increase in quality. [1]

With increasing globalization and networking, the process speeds of this world are increasing more and more. Increasing international competition, strongly fluctuating demand for products coupled with customer-specific product requirements have resulted in a continuously rising number of variants and at the same time enormous cost pressure for years. For most companies, this has made the field of action more dynamic, unpredictable and turbulent. [2] [3]

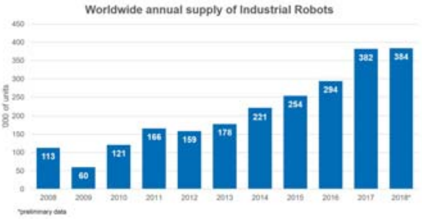


Fig. 1. Development of the use of industrial robots [4].

The history of industrial automation is characterized by periods of rapid change and development. In particular, the use of industrial robots, which began in the 1960s, is considered the latest trend in the context of automation of manufacturing and assembly processes [5]. This is underlined by the development of the use of industrial robots in figure1. Even today, industrial robots are primarily used for repetitive, dangerous or heavy tasks [6], making it possible to perform a large number of such manufacturing processes automatically and without the use of human workers [7].

In order to further increase the flexibility of robot-based production and assembly systems, the field of human-robot collaboration (HRC) and sensitive robots are considered to be particularly promising and are therefore massively developed. By means of HRC different forms of cooperation are tried to combine the strengths of humans and robots in synergy. [7] Especially the intelligent use of sensor technology enables a robot to interact not only with human being but also with its environment. The latter is the focus of this paper.

In order to realize the mentioned interaction with the environment, sensitive robot systems are used, which are able to perform force- or torque-controlled applications. Especially force-controlled applications offer a high application potential in industrial environments. On the one hand, it is possible to adapt the robot path based on the sensor technology and on the other hand to identify the environment.

In this context, different approaches are depending on the robot manufacturer. This means in particular that the design of the sensor technology and the associated control strategy can differ enormously.

II. Sensitive robots

There is no clear definition of the term sensitivity. Based on the measurement technology DIN 1319, sensitivity is defined as the change in the value of the output variable of a measuring instrument in relation to the causal change in the value of the input variable [8]. Derived from the context of robotics, the input variable can be understood as force or torque and the output variable as electrical voltage or current. Thus, the robot is not allowed to run a program in a purely position-controlled manner, but to adapt the path due to the change in electrical voltage and current, which are directly proportional to the speed and torque of the electric motors. This allows the robot to react in real time to the feedback of a force sensor and thus to deviate from its programmed path and speed [9] [10].

It is precisely this integration of additional sensors or modern control engineering methods that are increasingly leading to the use of robots with sensitive functionality. This technological extension paves the way for collaborative robot applications which, in addition to ensuring physically safe cooperation between humans and robots, also enable sensitive movements or tasks. The development of sensitive serial robots inspires further developments within robotics and numerous accompanying technologies. However, it also poses major challenges due to high expectations on the part of industry. [11]

Sensitive robots have been available on the market for several years now, in addition to the conventional industrial robots. Sensitive manipulators and conventional robots with adequate sensors make it possible to implement HRC applications (applications in the field of human-robot collaboration), as they can determine forces acting on the robot arm [11]. Industrially used systems include the KUKA LBR iiwa, the Universal Robots UR10e, the Franka Emika Panda, the Yaskawa MOTOMAN HC 10, the Doosan Robotics M1013, the DENSO COBOTTA and the Fanuc CR-35iA. These are shown in figure 2.

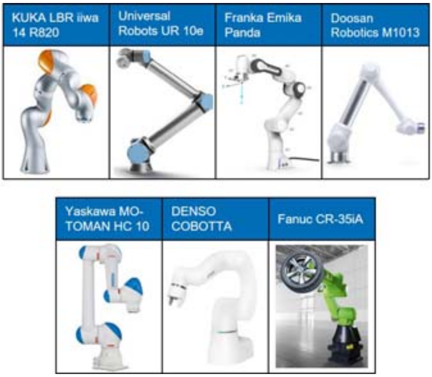


Fig. 2. Overview of sensitive robots suitable for industrial use [12] [13] [14] [15] [16] [17] [18].

The enablers that set these robots apart from conventional robots and thus enable sensitive tasks are internal software modules in conjunction with specific sensors. This enables the implementation of numerous safety functions of robots, such as force limitation or path monitoring. [19]

Furthermore, the sensor technology mentioned above offers far more potential than a human-robot cooperation application. By means of these sensors, various assembly tasks requiring sensitive behaviour can be transferred from the human to the machine, in this case a robot.

Interests at this point are first of all the concepts of the different robot manufacturers, how and especially what kind of sensors are used. The following overview shows the sensitivity respectively the Force/Torque recording of the integrated sensors of the industrial robots shown in figure 2.

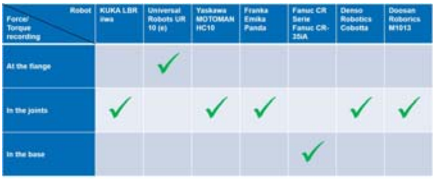
Property: Robot:	Force/Torque recording
KUKA LBR iiwa	Integrated torque sensors in each axis
Universal Robots UR 10e	Electrical current monitoring (dual channel), Integrated force/torque sensor in the flange
Yaskawa MOTOMAN HC10	Integrated force-torque sensors in each axis (optical sensors)
Franka Emika Panda	Integrated torque sensors in each axis
Fanuc CR-35iA	Force and torque sensor in the base
Denso Cobotta	Integrated speed and torque sensors in each axis
Doosan M1013	Integrated torque sensors in each axis

TABLE I
Overview of the sensitive properties of selected sensitive robots suitable for industrial use [12] [13] [14] [15] [16] [17] [18].

Table I shows how the different robots measure and recognize external forces and torques. The corresponding sensors usually base on so-called strain gauges, only the Yaskawa MOTOMAN HC10, in contrast, has optical sensors.

It is also noticeable that the place where the forces and torques are recognized is designed differently. As a result, the force/torque determination depends on the location, which is shown in Fig. 3. A differentiation is made between the determination

- at the flange
- in the joints
- in the base.



Robot Force/ Torque recording	KUKA LBR iiwa	Universal Robots UR 10 (e)	Yaskawa MOTOMAN HC10	Franka Emika Panda	Fanuc CR Serie Fanuc CR-35iA	Denso Robotics Cobotta	Doosan Roborics M1013
At the flange		✓					
In the joints	✓		✓	✓		✓	✓
In the base					✓		

Fig. 3. Overview of the operating principles of selected sensitive robots suitable for industrial use [12] [13] [14] [15] [16] [17] [18].

Some force-guided case studies or applications are shown in the figure 4.

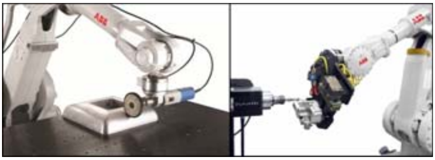


Fig. 4. Examples of force-controlled machining processes [20].

It shows common application scenarios of sensitive robots for the execution of force-controlled movements in the context of machining of components. In addition, there are also practical applications in this respect in the context of assembly operations. This is shown in Fig. 5.

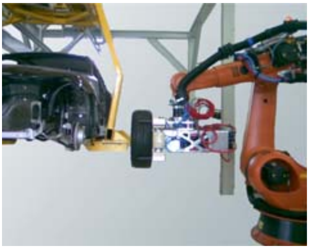


Fig. 5. Example of a force-controlled assembly process [21].

These two figures once again illustrate the potential of force-controlled applications of sensitive robots, since this is the only way to automate various processes.

III. State of the Art

A. Force Control

As mentioned in the introduction, not only sensitive robots with intrinsic sensors are able to perform force controlled applications. The beginning of the development of force-controlled applications goes back three decades. This illustrates the potential of force-controlled applications and especially the interest of research, science and industry in this area. Force control also plays a fundamental role in the achievement of robust and versatile behavior of robotic systems in openended environments, providing intelligent response in unforeseen situations and enhancing human–robot interaction. [22]

B. Strain gauges

Strain gauges are nowadays used in many technical areas. In addition to measuring strain, they are also ideal for building transducers for mechanical measurement quantities. In these transducers, strain gauges measure the strain that the mechanical quantity to be measured by the transducer causes in the measuring body of the transducer. In this way it is possible to measure indirectly e.g. forces, pressures or torques with the aid of strain gauges. However, the function of the strain gauges always remains the same. Due to mechanical tension, mechanical strain occurs, which causes the resistance to change accordingly. [23] The following illustration shows how it works schematically.

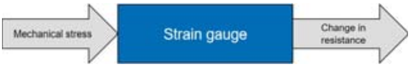


Fig. 6. Principle of how strain gauges work.

In order to generate a metrologically more attractive signal, namely an electrical signal, the Wheatstone bridge circuit is used. This transforms the measured quantity strain into an electrical quantity that can be easily processed further, which is also the greatest advantage of strain gauges. In addition, the Wheatstone bridge circuit offers excellent possibilities for compensating to a large extent for the effects of undesirable influences during this conversion, such as those that can occur with temperature changes. [23]

C. Force-torque sensors

After the function of strain sensors and strain gauges in particular was explained in the previous section, the following section describes how the measurement of forces and moments can be realized. The focus will be on measurement using resistance-based strain gauges.

According to DIN, the definition of the degree of freedom of a body in space is as follows. "The degree of freedom (DoF) f is the number of possible independent movements (displacements, rotations) of a rigid body in relation to a reference system." [24]. As a result, a freely moving rigid body has a maximum of f = 6 degrees of freedom in space, which are composed of three translational and three rotational movement possibilities. Analogously to the degrees of freedom of a freely moving rigid body in space, mechanical systems can also be described by six degrees of freedom - three forces and three torques associated with the directions in space, as shown in figure 7.

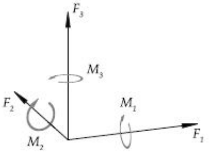


Fig. 7. Possible degrees of freedom for forces and torques [25].

At least six sensor values are required to measure these components. The link between sensor values S_j and the forces and moments can be described by the following system of equations by use of the Coupling matrix $\boldsymbol{k}$:

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix} \cdot \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{bmatrix} \quad (1)$$

D. Force-torque detemínation KUKA LBR iiwa and Universal Robots UR 10e

The robots relevant for this paper, namely the KUKA lbr iiwa and the Universal Robot UR10e, have two completely different ways of determining forces and torques.

While the Kuka LBR iiwa has torque sensors in each axis, the UR 10e uses a 6-DoF force/torque sensor installed on the flange of the robot. However, both have in common the evaluation by means of the Wheatstone bridge circuit.

The KUKA LBR iiwa calculates the external forces and torques $\underline{F}$ by the use of the Jacobian matrix $\boldsymbol{J}$ and the torques in each axis $\underline{\tau}$ (see eq. 2).

$$\underline{\tau} = \boldsymbol{J}^T \cdot \underline{F} \quad (2)$$

In contrast to that, the Universal Robots UR 10e uses a 6-DoF force/torque sensor. Figure 8 shows a frequently used design. There it can be seen that the elastic spring body has four axial bending beams on which the deformation is measured. The strains are produced by the relative displacement or rotation of the inner ring to the outer ring. The bending beams are connected to the outer ring via elastic elements, which primarily induces bending in the inner part of the beams. To achieve high sensitivity, two strain gauges are placed on each side of the beams. The four opposite strain gauges are connected to form a Wheatstone bridge as described in section III-B.

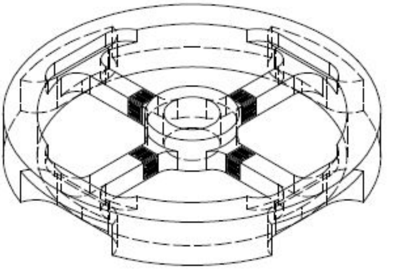


Fig. 8. Design and mode of operation of a force sensor with elastic spring body with four axial bending beams for force measurement in six degrees of freedom (from [26]).

IV. Control technology

In order to be able to execute force-controlled applications, suitable control technology is indispensable. Different types of controllers are used in the control of handling devices. These usually include position and force controllers. The former are used when a defined trajectory is traversed in space, while the latter are preferably used when the end effector is in direct interaction with the robot's environment. [27]

With regard to the subject matter of this paper and the available robots at the ZeMA -KUKA LBR iiwa and Universal Robot UR10e- the

- Position control
- Impedance control
- Force and torque control

are the most important controllers and are therefore described in more detail in the following sections. It is essential to break down both the similarities and differences between the two robots.

A. Position control

Basically, two different control strategies can be pursued - namely model-free and model-based controllers. As the name of the controllers already implies, they differ in their knowledge of the dynamic model of the robot. [28] [29]

Cascade control is the most widely used variant of model-free position control for robots with electric drives. Characteristics for this control structure are nested control loops, where the control is performed with increasing time constants of the controlled variable in different levels. Thus the dynamics of the system are taken into account. [30] Figure 9 shows the signal flow diagram of cascade control.

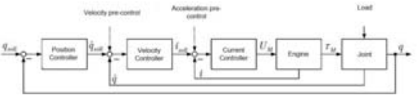


Fig. 9. Signal flow diagram of the model-free cascade control [30].

As can be seen in the figure, the cascade consists of a current, speed and position controller. Typically, the two innermost ones have a PI structure, while the position controller is usually designed as a proportional controller [30] With the help of this control strategy, good path accuracy can be achieved as long as there is no contact with the environment. If this is the case, pure position control is insufficient, which is why other control concepts must be used.

Besides the model-free control of a robot, there is also a model-based method whose control loop is shown in Fig. 10.

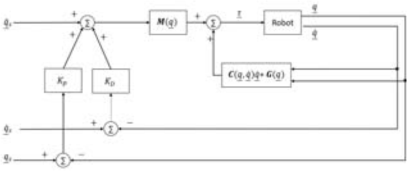


Fig. 10. Signal flow diagram of model-based position control, based on [5].

Further information in this regard can be found in the relevant literature [5] [31] [32].

B. Impedance control

Similar to the model-based position control mentioned above, the impedance control is also based on the dynamic model of the robot and therefore requires detailed knowledge of the model in the form of special, explicit equations (see Eq. 3).

If the Newton-Euler equations are evaluated symbolically for any manipulator, they result in a dynamic equation that can be written in the following form [5] [22]:

$$\underline{\tau} = \boldsymbol{M}(\underline{q})\underline{\ddot{q}} + \boldsymbol{C}(\underline{q}, \underline{\dot{q}})\underline{\dot{q}} + \boldsymbol{G}(\underline{q}) + \boldsymbol{J}^T \underline{F} \tag{3}$$

Here $\underline{\tau}$ is the vector of the torques produced at the joint by the drive, $\boldsymbol{M}$ is the mass inertia matrix, $\boldsymbol{C}$ is the matrix of Coriolis and centrifugal moments, $\boldsymbol{G}$ forms the matrix of weight moments, i.e. the torques produced by gravity. $\boldsymbol{J}^T$ represents the transposed Jacobian matrix and $\underline{F}$ he vector of the Cartesian forces and torques acting on the end effector. This equation of motion is used for the dynamic modeling of a robot and consists of nonlinear coupled differential equations of second order. Thus it is possible to link the location $\underline{q}$, the velocity $\underline{\dot{q}}$ and the acceleration $\underline{\ddot{q}}$ with the drive torques.

In general, model-based controls are less trivial than model-free controls and therefore more complex to implement [22] [29].

However, if the interaction of the robot with its environment is desired, impedance control is a frequently used control method. Basically, the impedance control is based on a virtual mass-spring-damper system.

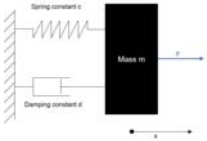


Fig. 11. Mass-spring-damper-system.

Figure 11 shows an example of a one-DoF mass-spring-damper system, whose dynamics can be calculated as follows

$$\boldsymbol{M}_m(\underline{\ddot{x}} - \underline{\ddot{x}}_s) + \boldsymbol{D}_m(\underline{\dot{x}} - \underline{\dot{x}}_s) + \boldsymbol{K}_m(\underline{x} - \underline{x}_s) = \underline{F}_{ext} \tag{4}$$

Where $\boldsymbol{M}_m$ is the mass inertia matrix, $\boldsymbol{D}_m$ is the damping matrix, $\boldsymbol{K}_m$ is the stiffness matrix, $\underline{F}_{ext}$ is the external force and $\underline{x}$, $\underline{\dot{x}}$, $\underline{\ddot{x}}$ are the position, velocity and acceleration. The index s means *desired* in this context. If this is now extended to the entire robot, it is possible to model the robot overall as a virtual mass-spring-damper system and to significantly influence the behavior of the robot during physical contact by adapting the sizes k and d [33]. Further literature can be found in [34] [35].

Figure 12 shows the block diagram of the impedance control.

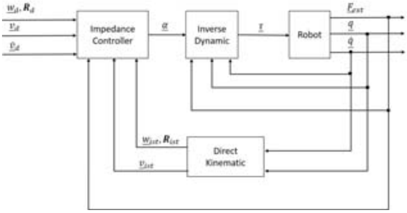


Fig. 12. Impedance control.

C. *Hybrid force/ position control*

In case of physical interaction of the end effector with the environment, as already mentioned, a pure position control is insufficient, but a pure force control is also insufficient. Typical tasks such as deburring, polishing or assembly often require force control in a Cartesian spatial direction. Therefore, it is useful to use a hybrid force/position controller, the aim of which is to ensure simultaneous control of both the end effector movement and the contact forces. This is achieved by a division into two areas, i.e. a decoupling of the sub-problems. [33] [22]

Depending on the operation it must be determined which of the translational and rotational degrees of freedom are either position- or force-controlled. This is done with the help of the so-called selection matrices $\boldsymbol{S}$ and $\tilde{\boldsymbol{S}}$. These have the dimension 6×6. The matrix $\boldsymbol{S}$ represents the force/torque controlled degrees of freedom and $\tilde{\boldsymbol{S}}$ the position controlled degrees of freedom. If the corresponding element on the main diagonal of the matrix has a 1, this direction is considered force- or torque-controlled. All other elements of the matrix are 0. Since the main diagonals of $\boldsymbol{S}$ and $\tilde{\boldsymbol{S}}$ are complementary to each other, using $\boldsymbol{I}$ as the unit matrix results in the following relationship. [30]

$$\tilde{\boldsymbol{S}} = \boldsymbol{I} - \boldsymbol{S} \tag{5}$$

Figure 13 shows the block diagram of hybrid force/position control.

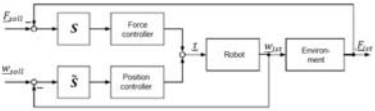


Fig. 13. Block diagram of the hybrid force/position control, based on [30].

Further information about a hybrid force/position control can be found in [36].

V. Comparison of the measured force curves

Often a typical practical application of sensitive robot applications is contour tracking and the associated contact with an object under constant force. For example, along a path the force along the Z-axis of the TCP (Tool Center Point) coordinate system should have a constant value. A comparison in this respect is shown in fig. 14. It is important that both robots have completed the trajectory under identical conditions. This means that the velocity and acceleration profile is the same for both.

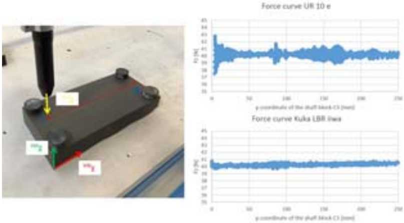


Fig. 14. Force curves with flat contour.

Because there is in practice rarely a very flat surface, it is also useful to expand the experiment shown in Fig. 14 to a more difficult contour in the form of an undulating profile. The result of this is shown in Fig. 15.

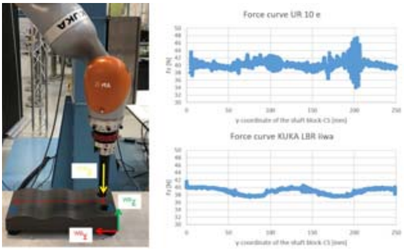


Fig. 15. Force curves with undulating contour.

When looking at the force curves, it quickly becomes apparent that the undulating profile poses a greater challenge for both robots.

In order to emphasize the importance of the impedance controller of the KUKA LBR iiwa and especially the parameterization, in figure 16 the force curve under the same basic conditions as in figure 15, only the impedance-specific parameters stiffness and damping were varied.

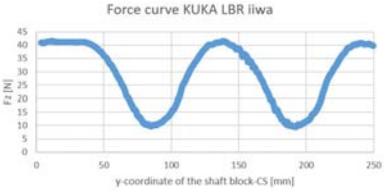
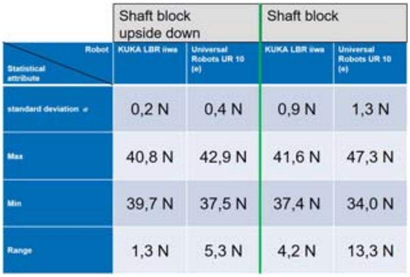


Fig. 16. Force curve of the KUKA LBR iiwa with bad parameterization of the impedance controller.

As the figure clearly shows, a force-controlled assembly task is only possible under certain conditions, namely the correct parameter identification of stiffness and damping constants.

The statistical comparison regarding the ability to perform sensitive tasks is shown in Fig. 17 in table form.



	Shaft block upside down		Shaft block	
Robot / Statistical attribute	KUKA LBR iiwa	Universal Robots UR 10 (e)	KUKA LBR iiwa	Universal Robots UR 10 (e)
standard deviation σ	0,2 N	0,4 N	0,9 N	1,3 N
Max	40,8 N	42,9 N	41,6 N	47,3 N
Min	39,7 N	37,5 N	37,4 N	34,0 N
Range	1,3 N	5,3 N	4,2 N	13,3 N

Fig. 17. Statistical comparison.

Since this is only a comparison of the two robots with regard to their sensitive capabilities and thus a pure test setup is used, there are no tolerance limits for the force component. Consequently, it is not possible to calculate the process capability, and the standard deviation σ serves primarily as an indicator of their capability.

VI. Conclusion

This paper aims to present the different methods of the determination of external force and torque regarding differnt robots. In particular, the location of the recording as well as the control technology are very different among robot manufacturers. By comparing the Universal Robots UR 10e with its 6-DoF force/torque sensor mounted at the flange and the KUKA LBR iiwa with its torque sensors in each axis, it could be shown that these different concepts, coupled with the respective control strategies, lead to different results with regard to their force accuracy.

However, it should be mentioned at this point that the accuracy of the force calculation of the KUKA LBR iiwa depends significantly on its configuration as well as on its dynamic modeling and the parameterization of the impedance controller. In contrast, the Universal Robot UR 10e offers the advantage that its kinematic configuration is completely independent of the determination of external forces and torques. This will be discussed in more detail in another publication.

Nevertheless, it became clear that the design and in particular the way in which the forces and torques are determined differ strongly from robot manufacturer to robot manufacturer, which results to different performances on force-controlled applications.

This comparison can be transferred to other robots, e.g. the Fanuc CR-35iA.

References

[1] B. Heinrich, P. Linke, and M. Glöckler, *Grundlagen Automatisierung: Erfassen - Steuern - Regeln*, 3rd ed., ser. Lehrbuch. Wiesbaden: Springer Vieweg, 2020.

[2] B. Vogel-Heuser, T. Bauernhansl, and M. ten Hompel, Eds., *Handbuch Industrie 4.0: Bd. 1: Produktion*, 2nd ed., ser. Springer Reference Technik. Berlin: Springer Vieweg, 2017. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-45279-0

[3] ——, *Handbuch Industrie 4.0: Bd. 2 : Automatisierung*, 2nd ed., ser. Springer Reference Technik. Berlin: Springer Vieweg, 2017. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-53248-5

[4] "Ifr international federation of robotics."

[5] J. J. Craig, *Introduction to robotics: Mechanics and control*, 3rd ed., ser. Pearson education international. Upper Saddle River, NJ: Pearson Prentice Hall, 2005.

[6] D. Busson, R. Bearee, and A. Olabi, "Task-oriented rigidity optimization for 7 dof redundant manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 588–14 593, 2017.

[7] A. Pott and T. Dietz, *Industrielle Robotersysteme: Entscheiderwissen für die Planung und Umsetzung wirtschaftlicher Roboterlösungen*. Springer Vieweg, 2019.

[8] DIN 1319, "Grundlagen der messtechnik: Begriffe für messmittel," 2005.

[9] ABB Automation GmbH, "Intelligente roboter durch kraft-moment-sensorik," 2014.

[10] verlag moderne industrie GmbH, "Sensitive roboter unterstützen die logistik." [Online]. Available: https://www.produktion.de/technik/sensitive-roboter-unterstuetzen-die-logistik-101.html

[11] M. Brandstötter, D. Mirkovic, and M. Hofbaur, "Mobile manipulation - eine altbekannte technologie findet durch sensitive robotertechnologie einzug in die industrie," 2017.

[12] KUKA AG, "Kuka lbr iiwa," https://www.kuka.com/de-de, 04.06.2020.

[13] Universal Robots, "Universal robots ur 10e," https://www.universal-robots.com/de/produkte/ur10-roboter/, 04.06.2020.

[14] Franka Emika GmbH, "Franka emika panda," https://www.franka.de/de/.

[15] Doosan Robotics, "Doosan robotics m1013," https://www.doosanrobotics.de/, 04.06.2020.

[16] Yaskawa Motoman Robotics, "Yaskawa motoman hc 10," https://www.motoman.com/en-us/products/robots/industrial/assembly-handling/hc-series/hc10, 04.06.2020.

[17] Denso Robotics, "Denso cobotta," https://www.densorobotics-europe.com/de/produkt-uebersicht/produkte/collaborative-robots/cobotta, 04.06.2020.

[18] Fanuc, "Fanuc cr-35ia," https://www.fanuc.eu/de/de/roboter/roboterfilter-seite/kollaborierende-roboter/collaborative-cr35ia, 04.06.2020.

[19] B. Dieber, A. Schlotzhauer, and M. Brandstötter, "Safety & security – erfolgsfaktoren von sensitiven robotertechnologien," *e & i Elektrotechnik und Informationstechnik*, vol. 134, no. 6, pp. 299–303, 2017.

[20] "Abb ltd."

[21] F. Lange, W. Bertleff, and M. Suppa, "Force and trajectory control of industrial robots in stiff contact," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 06.05.2013 - 10.05.2013, pp. 2927–2934.

[22] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Cham: Springer International Publishing, 2016. [Online]. Available: https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=4901894

[23] S. Keil, *Dehnungsmessstreifen*, 2nd ed. Wiesbaden: Springer Vieweg, 2017.

[24] VDI 2861, "Montage- und handhabungstechnik; kenngrößen für industrieroboter: Achsbezeichnungen," 1988.

[25] Ulrich Nolten, "Entwicklung und charakterisierung dehnungsbasierter kraft- und momentensensoren für medizinische anwendungen," Dissertation, Rheinisch-Westfälischen Technischen Hochschule Aachen, 2013.

[26] K. Miers, "Sensors. vol. 7: Mechanical sensors. hrsg. w. göpel, j. hesse, j. n. zemel; vch verlagsgescllschaft mbh, weinheim 1994. 674 s. preis dm 410,- bzw. dm 340,-," *Materials and Corrosion/Werkstoffe und Korrosion*, vol. 45, no. 5, pp. 322–323, 1994.

[27] R. V. Patel and F. Shadpey, *Control of Redundant Robot Manipulators: Theory and Experiments*, ser. Lecture Notes in Control and Information Sciences. Berlin Heidelberg: Springer-Verlag GmbH, 2005, vol. 316.

[28] W. Weber, *Industrieroboter: Methoden der Steuerung und Regelung*, 4th ed. Carl Hanser Verlag GmbH & Co. KG, 2019.

[29] Alin Albu-Schäffer, "Regelung von robotern mit elastischen gelenken am beispiel der dlr-leichtbauarme," Dissertation, Technischen Universität München, 2002.

[30] A. Winkler, "Sensorgeführte bewegungen stationärer roboter," Dissertation, Technische Universität Chemnitz, Chemnitz, 2015.

[31] C. H. An, C. G. Atkeson, and J. M. Hollerbach, *Model-based control of a robot manipulator*, ser. The MIT Press series in artificial intelligence. Cambridge, Mass.: MIT Press, 1988.

[32] V. Bargsten, P. Zometa, and R. Findeisen, "Modeling, parameter identification and model-based control of a lightweight robotic manipulator," in *2013 IEEE International Conference on Control Applications (CCA)*. IEEE, 28.08.2013 - 30.08.2013, pp. 134–139.

[33] K. M. Lynch and F. C. Park, *Modern robotics: Mechanics, planning, and control*. Cambridge and New York, NY and Port Melbourne: Cambridge University Press, 2017.

[34] S. Chiaverini and L. Sciavicco, "Force/position control of manipulators in task space with dominance in force," *IFAC Proceedings Volumes*, vol. 21, no. 16, pp. 137–143, 1988.

[35] A. Ibeas and M. de La Sen, "Robust impedance control of robotic manipulators," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. IEEE, 17.12.2004 - 17.12.2004, pp. 1258–1263 Vol.2.

[36] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 1981.

External communication of the robot ABB YuMi via virtual machine with analysis of hybrid position-force control

Wang Yiguo
Université de Lorraine, Art et Métier
Paris Tech, LCFC
Metz, France
wang_yiguonuaa@163.com

Meryem Taghbalout
Université de Lorraine, Art et Métier
Paris Tech, LCFC
Metz, France
meryem.taghbalout@gmail.com

Jean-François Antoine
Université de Lorraine, Art et Métier
Paris Tech, LCFC
Metz, France
jean-francois.antoine@univ-lorraine.fr

Gabriel Abba
Université de Lorraine, Art et Metier
Paris Tech, LCFC
Metz, France
gabriel.abba@univ-lorraine.fr

Abstract—**In order to achieve real-time control and convenient debugging of the robot, it is necessary to establish a stable, smooth and convenient communication and connection between the robot and the server (ROS/Robotstudio). In this paper, we establish this kind of communication through the virtual machine, where we can achieve ROS and Robotstudio running in the windows system at the same time. Besides, the latter part of the article introduces the process and result prediction of the hybrid position-force control.**

Keywords—ABB YuMi, virtual machine, communication, virtual sensor, hybrid position-force control

I. Introduction

The communication between ABB robot and server is mostly based on Linux or Windows system. And in the system of Linux, the server is mostly the ROS, while in the Windows, the server is used to be Robotstudio. Both provide very complete solutions, they have their own characteristics, and can achieve very stable communication with the robot.

The RAPID programming language of Robotstudio is specifically developed for ABB robots, and has good closure and executable, but it is not a general-purpose programming language. And then, it is not convenient for most robot researchers to learn and conduct in-depth research, such as hybrid position-force control. On the contrary, the ROS which is developed and based on Python and C++ has better openness and universal applicability and can adapt to most robots. To use these two methods at the same time, it is often necessary to prepare two computers or install dual systems, which will undoubtedly reduce the development progress.

The virtual machine, working as a virtual operating system running in windows, provides an alternative solution. By establishing a bridge between the virtual machine and the physical machine, it can be dis-guised as a real machine in the physical network to obtain an IP address and access this network, and thereby establish communication with the robot. At the same time, the physical machine can maintain its original communication. By now, at the network layer, a local area network has been established between robots, physical machines and virtual machines, and they can communicate with each other.

In addition, based on the establishment of communication, some deeper research can be carried out – hybrid position-force control. But for Yu-Mi's highly compact structure[1], it is not equipped with joint torque sensor, so this article also introduces the theory of virtual sensor.

II. The Establishment of communication

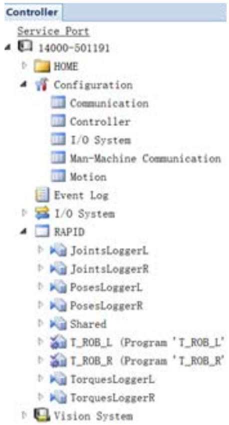
Fig.1. ABB IRB 14000 YuMi collaborative robot with two arms, 14 degrees of freedom

The key point for the communication is the establishment of a local area network. In the experiment, a network cable was used to connect the X23 service port of the robot to the network cable interface of the physical machine (PC). This kind of LAN (local area network) established by wired connection has the characteristics of high anti-interference, good confidentiality, and high transmission efficiency.

A. The installation of server and client control package

The control packages used in the experiment is provided by UC Berkeley's Automated Laboratory[2] which consist of server and client program. The server part is based on Python and is installed on the virtual machine, and the client part is based on RAPID and is installed on the robot. These two packages stipulate the specification and method of communication is the socket two-way communication. In the process of exchanging information, the server selects the

RAPID program to be called through the input network port of the client, and inputs variables. It can also read the state parameters of the robot through the output network port of the client.



```
Controller
Service Port
◢ 14000-501191
  ▷ HOME
  ◢ Configuration
      Communication
      Controller
      I/O System
      Man-Machine Communication
      Motion
    Event Log
  ▷ I/O System
  ◢ RAPID
    ▷ JointsLoggerL
    ▷ JointsLoggerR
    ▷ PosesLoggerL
    ▷ PosesLoggerR
    ▷ Shared
    ▷ T_ROB_L (Program 'T_ROB_L'
    ▷ T_ROB_R (Program 'T_ROB_R'
    ▷ TorquesLoggerL
    ▷ TorquesLoggerR
  ▷ Vision System
```

Fig.2. Configuration of client-side in Robotstudio

The installation of the client-side package needs the help of ABB's Robotstudio which only works in the Windows. The client-side mainly contains the RAPID control program and several values about the robot's initial position, joint angle, and joint torque.

In Linux, the server-side package can be installed and run using python alone or in ROS.

B. *Configuration of ports about IP address*

The Internet Protocol address (IP address) is a numerical label assigned to each device connected to a network that uses the Internet Protocol for communication. It serves two main functions: host or network interface identification and location addressing. The local area network (LAN) is a network that interconnects devices within a limited area, and these devices recognize and connect to each other through specific types of IP addresses (as shown in figure 3).

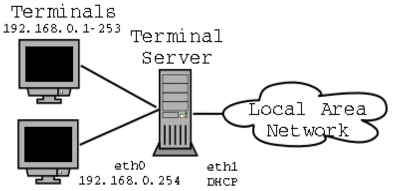


Fig.3. The composition and working principle of a simple local area network, DHCP is a dynamic IP address allocation service.

To ensure that the machines can communicate with each other, the IP address of each port must be configured to guarantee that they are in the same local area network.

The requirement of different IP addresses being located in the same local area network is that, when the commonly used IPv4 standard decimal IP address is converted into a binary form, and a logical AND operation is performed with the binary form of the own subnet mask, their results are always the same, and then, they are in the same local area network.

The robot's IP address is set by the client's RAPID program (can be modified) as shown below.

IP address: 192.168.125.1

Subnet mask: 255.255.255.0

Default gateway: 192.168.125.1

Based on this, in terms of physical machine, it is necessary to fix the IP address of the server to the same LAN to ensure communication. The IP addresses used here are all static IP addresses, and the DHCP service of the host needs to be turned off to avoid repeated addressing. IP address in the Internet Protocol Version 4 (TCP/IPv4) Properties of the Ethernet Properties (with Wired network card, ex. Realtek PCIe GBE Family Controller) in the Internet Connection of the physical machine needs to be set to manual, and fill in the static IP address as follows where The setting of the IP address is not unique.

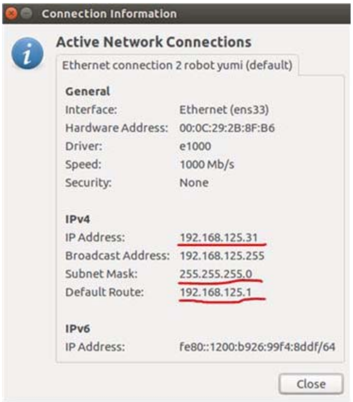
IP address: 192.168.125.5

Subnet mask: 255.255.255.0

Default gateway: 192.168.125.1

In terms of virtual machines, a connection port must be increased or modified to "bridge mode" in the option of Virtual Network Editor (N), and the virtual Network Adapter is directed to this connection port to ensure that the bridge between the physical machine and the virtual machine is officially established.

Meanwhile, in order to ensure that the virtual machine's network can correctly point to this "bridge", the IP address of its connection can be set as figure 4 (IP address of the same LAN).



Connection Information

Active Network Connections

Ethernet connection 2 robot yumi (default)

General

Interface:	Ethernet (ens33)
Hardware Address:	00:0C:29:2B:8F:B6
Driver:	e1000
Speed:	1000 Mb/s
Security:	None

IPv4

IP Address:	192.168.125.31
Broadcast Address:	192.168.125.255
Subnet Mask:	255.255.255.0
Default Route:	192.168.125.1

IPv6

IP Address:	fe80::1200:b926:99f4:8ddf/64

Fig.4. Virtual machine or Linux network connection settings, the IP address setting here is not unique either.

Note that when setting the IP address of each port, do not set the same as the robot, to avoid the situation of IP address preemption. Since the DHCP service was turned off in the experiment, IP could not be dynamically allocated, so the same IP would cause addressing failure and the failure to connect.

C. Test of communication and results

The test results after the establishment of the entire local area network show that the robot and the virtual machine can maintain stable and smooth communication.

In the hardware part, Linux has provided a very concise and effective command "ping" to test the connectivity of each host in the network. If the target host is running and connected to the network, it will respond to the current host's echo signal and return a data packet with a certain byte.

By using the following instruction,

ping 192.168.125.1

By monitoring the data traffic of the network port, the connection status of the host computer and the robot in the local area network can be intuitively displayed.

The results show that the response time for transmit-ting a 64-bit data packet is between 1ms and 6ms, with an average of 3ms. And no signal is lost (as shown in figure 5).

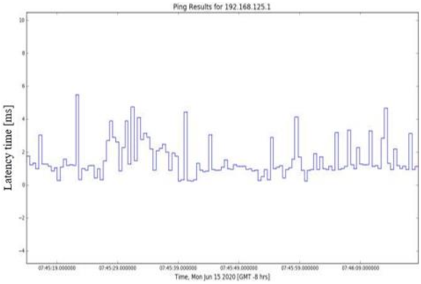


Fig.5. The monitoring process of the connection status of robots and virtual machines, and response time for sending and receiving data packets

In the software part, to test the functional integrity of the control package of the server and client, a simple test program needs to be written and run. In this test program, “YuMiSubscriber”, “YuMiState”, “YuMiRobot”, YuMiArm” in the control package, as well as “get_state()”, “goto_state()”, “get_pose()”, “goto_pose()”, “get_torque()” and other functions are used. In addition, python's drawing module “matplotlib” and multi-threading module “threading” have also been tested, and the results show that they are compatible with each other.

YuMi's left arm is controlled to move in the X, Y, and Z directions, and then returns to the initial point along the diagonal, and the position and state of the left arm at the turning point are shown below in the figure 6.

In addition, the real-time trajectory diagram of the end effector drawn by reading the joint angle is as follows, and the projections of the spatial trajectory on the X-Z, Y-Z, X-Y plane are shown in figure 7 and 8. The figure 9 shows the torque of joints by reading the real-time data.

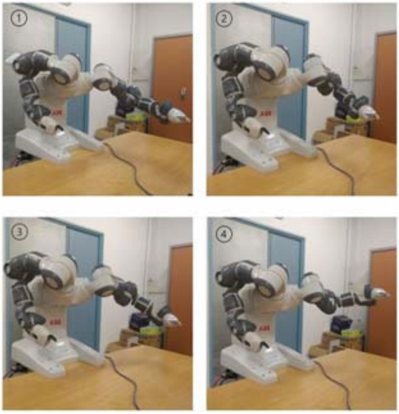


Fig. 6. Demonstration of the movement of the left arm of the YuMi robot, the movement sequence is from left to right, from top to bottom

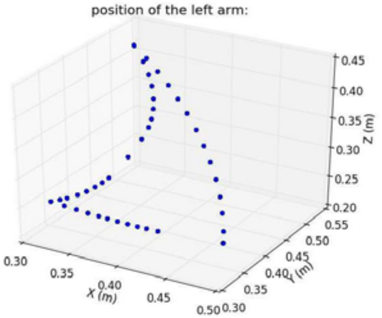


Fig.7. The trajectory of the end effector in three-dimensional space.

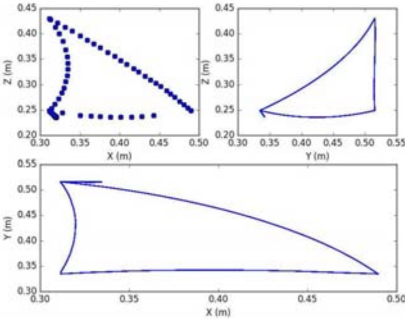


Fig.8. Projection of three-dimensional trajectory on three planes, from left to right, from top to bottom are the X-Z, Y-Z, X-Y plane

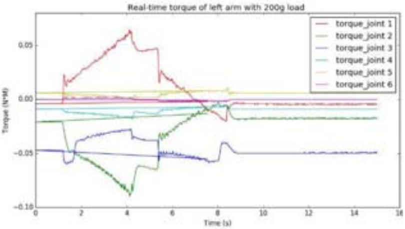


Fig.9. The torque of the 6 joints of the left arm (load of 200g, movement of 0.3m in Y direction of the base coordinate and going back), and the 7th joint torque is not included.

III. THEORY OF HYBRID POSITION-FORCE CONTROL

The prerequisite for the generation of contact force is the existence of contact, and the existence of contact is determined by the relative position, that means, the contact force is affected by the position. So, the hybrid position-force control based on position control is a form which is more common and easier to implement. The basic idea of the hybrid position-force control is that, all three-dimensional workspaces have 6 degrees of freedom (3 in translations and 3 in rotations), and these 6 degrees of freedom provide 6 controllable aspects (as shown above in figure 10).

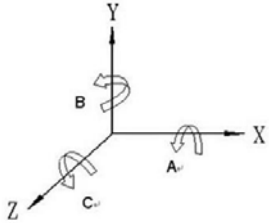


Fig.10. Demonstration of six degrees of freedom in three dimensions

For the sum of single force control and position control, they have 6 position variables and 6 force variables separately to occupy these 6 controllable aspects, but (between position control's variables and force control's variables) they are not linked to each other.

The hybrid position-force control combines the two and affects these 6 controllable aspects of both force and position. Note that one aspect can only be as-signed to one of those 12 variables (the 6 position variables and 6 force variables). In other words, it is impossible to carry out a position command in one direction while effecting a force command in the same direction. For rigid bodies, this kind of redundant control at one aspect cannot be achieved

In order to realize this theory, the force and position parameters need to be obtained first. Since the communication between the robot and the computer has been established, the data available through the control package are the real-time position of the joint and the torque of the joint motor. The former can be used directly, but the latter is not a real joint torque and needs to be converted using virtual sensors.

A. Implementation of the virtual sensor

The core idea of the virtual sensor is the conversion of force, including the spatial transfer (offset and rotation) of force, and the compensation of force [3][4].

After the process of force analysis, it can be figured out that when the torque of the motor is transmitted to the joint, it will be affected by the inertial force, Coriolis force, friction, and gravity[5][6]. Conversely, if the motor torque is subtracted from these disturbance forces, then the torque generated by the external force acting on the end effector and transmitted to each joint can be obtained.

The theoretical formula is as follows,

$$Q = M(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + F(\dot{q}) + G(q) \qquad (1)$$

Or

q is the vector of generalized articular coordinates describing the pose of the joint,

$\dot{q}$ is the vector of joint velocities,

$\ddot{q}$ is the vector of joint accelerations,

C describes the effects of Coriolis and centripetal - centripetal couples are proportional to $\dot{q}_i^2$, while Coriolis couples are proportional to $\dot{q}_i \dot{q}_j$,

M is the symmetric space-joint inertia matrix, or inertia tensor of the manipulator,

F describes viscous and coulombic friction and is not generally considered to be part of the dynamics of the rigid body,

G is gravity loading,

Q is the vector of the generalized forces associated with the generalized coordinates q, and the theoretical difference between the motor torque and the actual torque of joint.

In the process of analyzing Q, the motor torque can be obtained through the command "get_torque()" in the package yumipy which has been introduced in the previous chapter. Besides, it can be figured out that the factor of gravity $G(q)$ is only related to the joint angle. The factor of friction $F(\dot{q})$ is composed of sliding friction and viscous friction where the former is a fixed value, the latter is related to angular velocity of the joint, and both of them need to be measured through separate external rotation experiments. The factor of inertial force $M(q) \cdot \ddot{q}$ consists of two parts, $M(q)$ which is only related to the joint angle and $\ddot{q}$. And $M(q)$ can be obtained by RNE algorithm. For low-speed and low-continuous motion states, the factors of Coriolis force and inertial force have little effect, and the robot is mainly affected by factors of gravity and friction. For high-speed and high-continuous motion, the situation is reversed.

In the actual programming process, the calculation of these compensation forces uses the Recursive Newton Euler algorithm (RNE)[1]. And the setting of the coordinate system is based on the standard DH (as shown in figure 11).

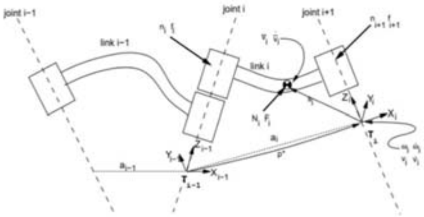


Fig.11. Notation used for inverse dynamics, based on standard Denavit-Hartenberg notation

RNE calculates the inverse dynamics of the arm, which means the calculation of joint torque, by a given set of angles, speeds, and joint accelerations. The algorithm is shown below.

Outward recursion, $1 \leq i \leq n$

$$\omega_{i+1}^{i+1} = R_i^{i+1}(\omega_i^i + z_0\dot{q}_{i+1})$$

$$\dot{\omega}_{i+1}^{i+1} = R_i^{i+1}\{\dot{\omega}_i^i + z_0\ddot{q}_{i+1} + \omega_i^i \times; l(z_0\dot{q}_{i+1})\}$$

$$v_{i+1}^{i+1} = \omega_{i+1}^{i+1} \times P{*}_{i+1}^{i+1} + R_i^{i+1}v_i^i$$

$$\dot{v}_{i+1}^{i+1} = \dot{\omega}_{i+1}^{i+1} \times P{*}_{i+1}^{i+1} + \omega_{i+1}^{i+1} \times \{\omega_{i+1}^{i+1} \times P{*}_{i+1}^{i+1}\} \\ + R_i^{i+1}\dot{v}_i^i$$

$$\dot{\bar{v}}_i^i = \dot{\omega}_i^i \times s_i + \dot{\omega}_i^i \times \left(\dot{\omega}_i^i \times s_i\right) + \dot{v}_i^i$$

$$F_i^i = m_i\dot{\bar{v}}_i^i$$

$$N_i^i = J_i\dot{\omega}_i^i + \omega_i^i \times (J_i\omega_i^i)$$

Inward recursion, $n \geq i \geq 1$

$$f_i^i = R_{i+1}^i f_{i+1}^{i+1} + F_i^i$$

$$n_i^i = R_{i+1}^i\{n_{i+1}^{i+1} + \left(R_i^{i+1}P^*{}_i\right) \times f_{i+1}^{ii+1}\} \\ + \left(P{*}_i^i + s_i\right) \times F_i^i + N_i^i$$

$$Q_i = \left(n_i^i\right)^T(R_{i+1}^i z_0)$$

And where ω, and v represent the common angular velocity, linear velocity, as well as the accelerations extended from them; F and N are the force/torque of the center of gravity of the joint; f and n are the force/torque between the joints; Q is the effect of external forces on each joint; R and P are rotation and offset matrices; s is the position of the center of gravity in the joint coordinate system; z_0=[0,0,0].

It has two recursive loops. Forward recursion propagates kinematic information - such as angular velocities, angular accelerations, linear accelerations - from the reference coordinate system to the end effector. Backward recursion propagates the forces and moments exerted on each link from the end effector to the reference coordinate system.

It can be found that RNE always recurs from the starting point of force or position, which has the advantage of greatly simplifying the difficulty of starting recursion, because the state of the starting point of force or position is often relatively simple, and this state often depends only on itself.

B. Classic double-loop control block

The classic form of hybrid position-force control is a double loop block - force control loop and position control loop (as shown in figure 12).

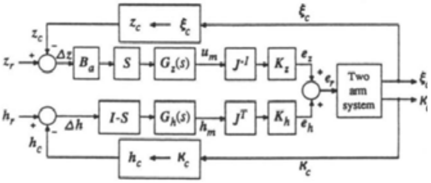


Fig.12. Theoretical diagram of the hybrid position-force control

This was first proposed by Masaru Uchiyama[1].

The upper loop performs position control and the lower loop performs force control. As mentioned above, there are a total of 12 variables in force and position, and these 12 variables are selected by the selection matrix S.

S is a 6×6 diagonal matrix, and its diagonal is composed of 0 or 1, where 1 represents position control at this azimuth, and 0 represents force control.

K is the 6×6 gain matrix, which is also a diagonal matrix, and its diagonal elements is determined by the gain required by the system, and it can be used to compensate for random errors in the system, as well as system errors due to inaccurate modeling of other parameters (friction estimation, inertia estimation, etc.). J is the 6×6 Jacobian matrix, which can be obtained by differential method, spatial algebra operator (SOA)[1] or others. Zr and hr represent the target position and target force. The remaining elements are determined by the different types of robots.

C. Expected result of hybrid position-force control

The expected result of the hybrid position-force control is derived theoretically (as shown in figure 13).

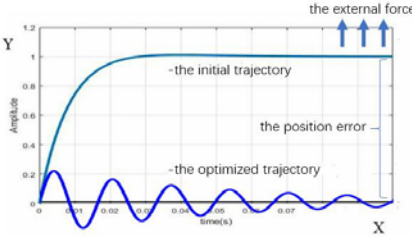


Fig.13. Comparison of trajectory between controlled and uncontrolled under external force by theoretical analysis

The ideal experimental state is as follows. In the base coordinate system, the end effector of robot is in contact with an object and needs to maintain a contact force along the direction of the Z axis. At the same time, the end effector of the robot will move repeatedly along the predetermined trajectory in the X-Y plane. Due to the influence of friction, in the process of the movement in the X-Y plane, the trajectory of the end effector will be affected.

This motion state simulates the process of the yumi robot massaging and scrubbing the patient's body during the medical care. Considering the complex situation where the muscle tissue of the human body is soft and the internal bones are relatively hard, the robot needs to ensure that the contact force is not too small to be effective or too large to cause injury, and the range of motion needs to be completely covered while being limited to a specific body area.

In the initial trajectory where the end effector moves in the X direction, due to the influence of the friction, the trajectory curve of the end effector will be more and more distant from the ideal trajectory (straight line), and the accuracy of final positioning also has a big error. After adding a hybrid position-force control, the negative feedback effect generated by the command program will be added to the trajectory of the end effector. In theory, due to the influence of inertia, gravity and other factors, an overshoot will occur and cause the trajectory to oscillate, but the final overall controlled performance should be better than the first, and the positioning accuracy should also be higher.

IV. Conclusion

This paper provides a method for establishing communication between the virtual machine and ABB robots, and experiments show that this method has high stability and transmission efficiency and can greatly simplify the operation process of the establishment of connection.

In addition, this paper also analyzes and predicts the hybrid position-force control of the ABB YuMi robot, and expounds the realization theory and process of the virtual sensor, which paves the way for further research on the hybrid position-force control.

Reference

[1] http://robots.ros.org

[2] https://berkeleyautomation.github.io/yumipy/install/install.html

[3] A. Wahrburg, J. Bös, K. D. Listmann, F. Dai, B. Matthias and H. Ding, "Motor-Current-Based Estimation of Cartesian Contact Forces and Torques for Robotic Manipulators and Its Application to Force Control," *in IEEE Transactions on Automation Science and Engineering,* vol. 15, no. 2, pp. 879-886, April 2018, doi: 10.1109/TASE.2017.2691136.

[4] M. Capurso, M. M. G. Ardakani, R. Johansson, A. Robertsson and P. Rocco, "Sensorless kinesthetic teaching of robotic manipulators assisted by observer-based force control," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 945-950, doi: 10.1109/ICRA.2017.7989115.

[5] Shih-Hsiang Yen, Pei-Chong Tang, Yuan-Chiu Lin, Chyi-Yeu Lin, “Development of a Virtual Force Sensor for a Low-Cost Collaborative Robot and Applications to Safety Control”, Sensors 2019, 19(11), doi.org/10.3390/s19112603

[6] M. Linderoth, A. Stolt, A. Robertsson and R. Johansson, "Robotic force estimation using motor torques and modeling of low velocity friction disturbances," 2013 IEEE/RSJ Interna-tional Conference on Intelligent Robots and Systems, Tokyo, 2013, pp. 3550-3556, doi: 10.1109/IRO S.2013.6696862.

[7] P. I. Corke, "A robotics toolbox for MATLAB," in IEEE Robotics & Automation Magazine, vol. 3, no. 1, pp. 24-32, March 1996, doi: 10.1109/100.486658.

[8] Masaru Uchiyama & Pierre Dauchez (1992) Symmetric kinematic formulation and non-master/slave coordinated control of two-arm robots, Advanced Robotics, 7:4, 361-383, DOI: 10.1163/ 156855393X00221.

[9] G. Rodriguez, A. Jain and K. Kreutz-Delgado, “Spatial operator algebra for multibody system dynamics”, The Journal of the Astronautical Sciences, 1992,40:27-50.

Mobile Robot Lifting Mechanism Design for Manipulation and transportation task

B. Hichri
Manufacturing Engineering Group
University, Luxembourg
Luxembourg
bassem.hichri@uni.lu

P. Plapper
Manufacturing Engineering Group
University, Luxembourg
Luxembourg
peter.plapper@uni.lu

***Abstract*—This article considers a design methodology for creating cooperative robots capable to manipulate and transport payloads. The strategy is based on tightening a payload between a set of similar mobile robots called m-bots. A lifting mechanism with two degrees of freedom mounted on each mobile robot allows then to lift the payload and put it on each m-bot top platform to be transported. Structural and dimensional analysis are detailed in order to develop the proposed mechanism based on the stability analysis of the payload on the top platforms of mobile robots. 3D multi-body dynamic software simulation results are presented to validate the proposed strategy.**

***Index Terms*—Cooperative mobile robots, Design of Lifting mechanisms, Object manipulation and transportation.**

I. Introduction

The growing sector of logistics requires specifically designed machines and could highly benefit from robotics. Some logistics solutions require heavy infrastructure such as ground landmarks or guiding rails for Automated Guided Vehicles (AGVs) [1] or specific stacked storage racks as for Automated Storage and Retrieval System (ASRS). Human assistance could also be needed to put the object on the transporting platform (*e.g.* for scissors elevators [2]). Forklifts [3] use forks to lift and transport the object but require to store the object on a pallet. Grabbing systems such as robot hands [4] limit the manipulated payload size and shape. Considering Manual transportation, many researches in the domain of Manual Material Handling (MMH) prove that operators have a better performance and less body suffering when keeping the payload low and close to the body [5]–[8]. According to the previous mentioned systems and to the previous studies linked to MMH, one can conclude that for a better stability, an object should be better transported on the robot body [9], [10] or as close as possible to the robot body. Using this approach, it can be ensured to keep the gravity center above the polygon of support. Keeping the gravity center as low as possible also ensures a better stability margin on slopes.

A group of Robots working together for a task achievement presents several advantages compared to a single robot with a complex kinematics, such as a reduced cost, robustness, efficiency and improved performance [11]–[13]. Particularly for manipulation and transportation tasks, many collaborative robotic systems could be found in literature. Using different techniques, a group of similar [14], [15] or heterogeneous robots [16] can ensure payloads transportation. Different strategies can be found in literature for multi-robot transportation. Pushing strategy proposed in [15] was used while a payload is on the ground. This strategy may face some difficulties depending on the friction generated by the contact surface with the ground and it can also affect the quality of the transported object. Other robots are using grabbing tools [17] for transportation which limits the shape of the objects that can be manipulated and requires geometries and shapes that could be gripped by the grippers. Some robots need the human assistance for putting payload on their transport platform such as the Arnold robot presented in [14]. In the proposed work, a strategy based on tightening a payload by a multi-robot system to manipulate it, lift it and autonomously put it on the robots platform autonomously is proposed. For our system we have supposed to use a mobile robot on which a manipulation mechanism is going to be mounted. The proposed solution will not be limited to a simple object category but will have to lift and transport objects of any shape and dimensions.

To ensure object lifting, a mechanism has to be chosen to ensure the movement of the object from an initial position on the ground to a final position on the robot body. For a better adaptability, a terminal organ ensuring a contact surface with the payload is used and the use of grippers is avoided because it limits the object shapes that can be manipulated and it also requires more actuators. To lift the object from the ground with a constant orientation, a variety of mechanisms that can ensure this function with different trajectories will be investigated. This general architecture allows to ensure the payload stability by putting it on the robot body. So a structural and dimensional synthesis for this mechanism are required to avoid collision problems and to ensure a better stability of the whole mechanism.

In this paper, a design strategy and implementation of cooperative robots for co-manipulation and transport of payloads of any shape and mass is proposed. Each robotic unit, called mono-robot or m-bot, is particularly characterized by its mechanical structure simplicity comparing to [9] and [18].The resulting poly-robot, or p-bot, obtained by combining several m-bots around the payload, has the advantages of modularity while using a swarm of elementary robots [15], [16], adaptability to objects of any shape and mass and ability to provide a fully autonomous system, without human mediation, contrary for example to the robotic system proposed in [19] and [20].

This paper is organized as follows. Section II presents the paradigm and requirements of the poly-robot, as well a the structural synthesis of the lifting mechanism on each mono-robot; Section III is dedicated to the dimensional synthesis and design of manipulation mechanism. Section IV presents the simulation and experimental results. Finally, section V presents a conclusion of the achieved work.

II. Project paradigm

A. Specification

This project aims to design identical mobile robots called **m-bots** equipped with a manipulation mechanism. The proposed work deals with collaborative tasks in which a group of similar entities are able to cooperate in order to achieve the task. It is dedicated to payloads of any shape co-manipulation and transport. The group of m-bots will be able to lift, co-manipulate and transport a payload which has to be laid on the top platform of each m-bot. Consequently, in addition to an end-effector, the m-bot manipulator has to include a lifting mechanism. The formed poly-robot, that we call **p-bot** (m-bot + payload), is characterized by its reconfigurability depending on the overall system stability and the success of task achievement. The set of robots configuration is obtained according to the positioning algorithm developed in [21]–[23] to ensure stability of the overall system (payload + m-bots) during the different task steps: co-manipulation and lifting, transportation and putting down the payload. The reconfigurabilty is needed to ensure the modification of the formation of the set of robots depending on the participant number of m-bots and in case of one or multiple robots break down. This reconfigurabilty is needed to allow the maintain of p-bot stability with respect to Static Stability Margin **SSM** and Force Closure Grasping **FCG** developed in [23]. The former is a criterion that ensures the stability during transportation phase and the latter ensures stability during manipulation phase. The m-bots architecture allows also the p-bot to maneuver in any direction and this is guaranteed by developing a centralized controller based on Virtual Structure **(VS)** Navigation developed in [21]. This controller ensures the control of each entity in a way that the set of robots evolve in a specified direction or have the same ICR to ensure coordinated rotation without loss of stability.

The general architecture of a m-bot is defined by the following requirements R_i presented in table I and relative to the environment in which it will operate.

For simplicity reasons, the end-effector is considered here to be a rigid contact plate in order to fit variable payload contact surfaces. According to the previous requirements, the global co-manipulation method will be described.

Requirement	Definition
R_1	Lift a payload in collaboration with similar m-bots using a manipulation mechanism
R_2	Transport a payload.
R_3	Collision-free payload trajectory from the ground to the robot top platform with constant orientation.
R_4	Evolve in structured terrain.
R_5	Ensure manoeuvrability.
R_6	Ensure stability.
R_7	Ensure reconfigurability.
R_8	Tighten the contact payload/m-bot.
R_9	Detect other m-bots.
R_{10}	Detect obstacles.

TABLE I
M-bot requirements

B. Co-manipulation method

For a better stability of the payload and to avoid the risk of payload slipping and falling down between the m-bots end-effectors, the strategy of Army Ants transportation [10] was adopted for putting the payload on the m-bots top platform. Finally the co-manipulation and transportation method was decided and illustrated in Fig. 1

The process of co-manipulation and transportation of a payload was initially described in [21]–[25]. The different phases of payload prehension, lifting and transportation are presented in Fig. 1. The first phase consists in locating the payload and surrounding it using distance sensors. The m-bots have to be oriented toward the object in order to face it (cf. Fig. 1(a)). Secondly, the payload is held by the m-bots end-effectors which exert a collective pressure using wheel propulsion (Fig. 1(b)). Submitted to collective pressure and to the proposed co-lifting manipulation, the object is elevated and laid on the m-bots top platforms (Fig. 1(c)). Finally, locomotion and transportation tasks are performed where each m-bot # m is steering by a suitable angle θ_m to ensure to the p-bot a unique Instantaneous Center of Rotation (ICR) (Fig. 1(d)).

C. Pre-dimensioning the lifting capacity

The forces applied to a m-bot are represented in Fig. 2 and denoted with a triple index $f_{m,j,k}$, with m the m-bot number, j the nature of the contact (g for ground, p for payload) and k the component of the force (n for normal, t for tangential).

A m-bot # m, with a mass M, could apply a pushing force $f_{m,p,n}$ at the contact point $C_{m,p}$ on the payload with a friction coefficient μ_p, which generates a lifting force $f_{m,p,t}$ counting on wheel propulsion. The contact point $C_{m,g}$ (wheel/gound) is characterized by a friction coefficient μ_g. The maximal lifting force for the m-bot # m can be written as:

$$f_{m,p,t} = \mu_p f_{m,p,n} = \mu_p f_{m,g,t} = \mu_p(\mu_g f_{m,g,n}) = \mu_p(\mu_g Mg) \quad (1)$$

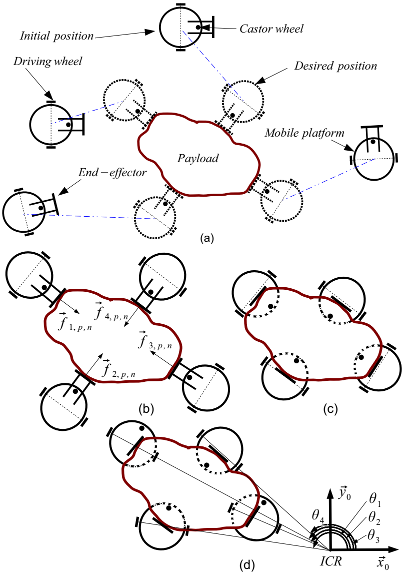


Fig. 1. Co-manipulation method: a) Target reaching; b) Object holding; c) Object set on robot bodies; d) Object transport: a unique Instantaneous Center of Rotation (ICR) requires different steering angles θ_m

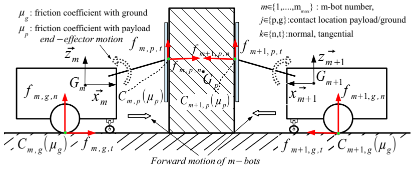


Fig. 2. Payload lifting by two m-bots.

The maximal total lifting force is

$$f_{p,t} = \sum_{m=1}^{m_{max}} f_{m,p,t} = m_{max}\mu_p(\mu_g M g) \qquad (2)$$

With the simplifying assumption $\mu_g = \mu_p = 0.5$
$\Rightarrow f_{p,t} = \frac{M m_{max} g}{4}$.
One can conclude that to increase the p-bot lifting capacity $f_{p,t}$, the total number m_{max} of m-bots, their mass M or the friction coefficients μ_g and μ_p have to be increased. As the environment and payload may be of different materials, the μ_g and μ_p coefficients are not precisely known and may be variable. They can be maximized by using adherent materials on the wheels and contact plate.

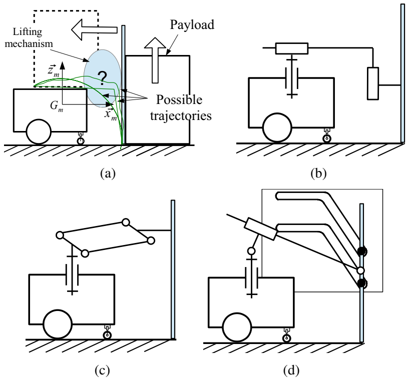


Fig. 3. Elementary lifting systems: a) Payload initial and final position with possible trajectories; b) 2 DOF solution; c) 1 DOF solution based on parallelogram mechanism; d) 1 DOF solution based on cam mechanism

III. Designing a lifting mechanism

A. Specification of the lifting mechanism

The lifting and manipulation mechanism used for object lifting must ensure the following requirements R_{li} presented in table II:

Requirement	Definition
R_{l1}	Manipulate payload via an end-effector.
R_{l2}	Allow object lifting.
R_{l3}	Ensure fittability on the robot mobile platform.
R_{l4}	Avoid collision with robot platform and the ground.
R_{l5}	Tighten contact payload/mechanism using the end-effector.
R_{l6}	Ensure fittability of the robot to the payload.
R_{l7}	Ensure orientability of the robot platform with respect to the payload.
R_{l8}	Put the payload on the robot body.

TABLE II
Manipulation mechanism requirements

B. Structural and dimensional synthesis of the lifting mechanism

Structural selection: The various system requirements R_i (cf. Table I) and manipulation mechanism R_{li} (cf. Table II) will influence directly the kinematic structure. R_5 and R_{l7} can be satisfied by supporting the lifting mechanism on a turret. As a consequence, a revolute joint with z axis will support the mechanism (cf. Fig. 3(b)), 3(c) and 3(d))). R_3 defines the initial and final poses P_1 and P_2 of the lower point P of the end-effector that holds the object. The latter

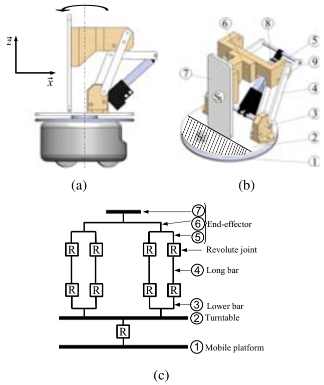
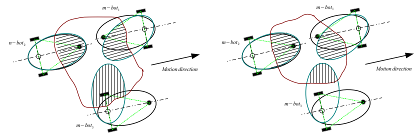


Fig. 4. Elementary lifting systems: a) 3D CAD for a m-bot; b) 3D CAD view for the manipulation mechaism; c) Binding graph

will keep its orientation constant during the lifting motion. The trajectory must start with a vertical lifting motion (+z_m) and finish with a backward horizontal motion (-x_m) towards the m-bot platform (Fig. 3(a)). R_3 and R_{l4} imply not to start the horizontal motion too early in order to avoid collision with the m-bot platform. Different trajectories are allowed (Fig. 3(a)) among which the square and the circular motions are the most obvious. A square trajectory could be achieved using two orthogonal prismatic joints and two actuators (Fig. 3(b)). A complex trajectory could also be ensured by using a cam mechanism (Fig. 3(d)). A circular trajectory would lead to a simpler solution using only one actuated revolute joint. However, to keep the payload orientation along the circular trajectory, a parallelogram mechanism is preferred (Fig. 3(c)) while keeping the control simplicity with a single actuator. The proposed mechanism will be fixed on the top of a unicycle mobile platform.

Structural analysis: Fig. 4 describes the proposed lifting mechanism. A turntable (Part *2*) is connected to the base (Part *1* fixed on the mobile platform) via a revolute joint (z_m axis) which allows the mobile platform of the robot to steer freely when the payload is on robot bodies (laid on surface S_2 on the top of *2*). Two identical parallelogram mechanisms are mounted on *2*. Each one is composed of a lower bar *3*, two long bars *4* and an end-effector support *5, 6, 7*. The payload to be manipulated is held by the contact surface S_1 of the end-effector. An actuator *8* is used to ensure object lifting and to control the parallelogram mechanism via an additional lever *9*. The actuator allows to maintain the pressure force on the payload.

Dimensional synthesis: Robotic platform and landing position



(a) Stable transportation (b) Unstable transportation

Fig. 5. M-bots possible configuration for payload transportation

The choice of a m-bot architecture depends on the system requirements previously defined. It also depends on several criteria to be ensured during the task achievement such as stability. So a m-bot must remain stable during the phase of target reaching and during the lifting and transporting phases. For experiments a three wheel robot existing in our laboratory will be used. This robot architecture is considered then, and is sufficient to ensure stability of the m-bot by maintaining the m-bot center of mass inside its polygon of support, when it evolves in the environment. Its stability margins could be calculated using different developed methods [26]–[29]. The adopted strategy for the transport as presented in the co-manipulation method (cf. Fig. 1) is based on transportation on robot bodies, and a suitable landing position is another constraint added to ensure the overall system stability. According to Fig. 5, one can conclude that depending on the payload positioning on robot body, the m-bot could be stable or unstable. Depending on the position of landing point P_2, a normal force $\vec{F}_{p,m}$ (cf. Fig. 6) applied by the payload on the m-bot, when it is laid on its turntable could either keep its stability or induce the m-bot reversal. A m-bot remains stable if the following conditions are satisfied:

$$\bar{M}_{(w_c w_r)}(\vec{F}_{p,m}) + \bar{M}_{(w_c w_r)}(\vec{P}_m) \geq 0 \text{ if } \psi \in [0, \frac{\pi}{2}] \tag{3}$$

$$\bar{M}_{(w_c w_l)}(\vec{F}_{p,m}) + \bar{M}_{(w_c w_l)}(\vec{P}_m) \leq 0 \text{ if } \psi \in [-\frac{\pi}{2}, 0] \tag{4}$$

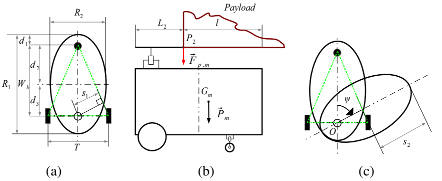
$$\bar{M}_{(w_r w_l)}(\vec{F}_{p,m}) \leq \bar{M}_{(w_r w_l)}(\vec{P}_m) \text{ if } \psi \in [-\frac{\pi}{2}, \frac{\pi}{2}] \tag{5}$$

In Fig. 7(a), the payload is laid on m-bot body in a manner that satisfies equation (4) and avoids the robot reversal. However, in Fig. 7(b) the generated torque by the payload position is able to make the m-bot$_3$ on the bottom side tip-over if it exceeds the torque generated by its weight. As a conclusion, if both forces $\vec{F}_{p,m}$ and $\vec{P}_m$ are in the same half space separated by the vertical plane passing through $(w_c w_r)$ or $(w_c w_l)$, then the m-bot remains stable during the task. In the other case, if the application points are in two different half spaces than the state of the m-bot will be defined as follow:

$$\begin{cases} \text{The m-bot is stable if} \\ \bar{M}_{(w_i w_j)}(\vec{F}_{p,m}) < \bar{M}_{(w_c w_i)}(\vec{P}_m) \mid i\#j \textit{ and } i,j = l,r,c \\ \text{or} \\ \bar{M}_{(w_c w_i)}(\vec{F}_{p,m}) = 0 \mid i\#j \textit{ and } i,j = l,r,c \\ \text{The m-bot is unstable if} \\ \bar{M}_{(w_c w_i)}(\vec{F}_{p,m}) > \bar{M}_{(w_c w_i)}(\vec{P}_m) \mid i\#j \textit{ and } i,j = l,r,c \end{cases} \tag{6}$$

The p-bot is developed in order to co-manipulate and transport payload while ensuring the overall system stability and successful task achievement. The payload must be laid in a manner that keeps every m-bot stable. This allows to define and to optimize the landing position P_2 of the payload on the robot turntable with respect to (3) and (4).
For the optimization problem, an objective function l, which corresponds to the landing position, is defined as follow, depending on robot parameters (cf. Fig. 6):

$$l > d_1 + W_b - s_1 = d_1 + W_b - \frac{W_b T}{2\sqrt{4W_b^2 + T^2}} \tag{7}$$



(a) (b) (c)

Fig. 6. M-bot parameters: a) mobile platform parameters; b) Payload laid on m-bot body; c) turntable steered by an angle ψ w.r.t the mobile platform

The objective function l must respect the following constraints:

$$l > s_2 = d_1 + W_b - s_1 \tag{8}$$

$$l \leq R_1 - L_2 \tag{9}$$

$$\psi \in [-\frac{\pi}{2}, \frac{\pi}{2}] \tag{10}$$

where L_2 presents the necessary length on the platform, which will be used to mount the manipulation mechanism. This parameter is defined as constant. The decision of the usefulness of a mobile platform depends on this parameter. For a specified platform, if $L_2 \geq R_1 - W_b - d_1 + s_1$, it will be impossible to use it with the proposed design for the task achievement, because the landing position will be out of the support polygon of the m-bot.

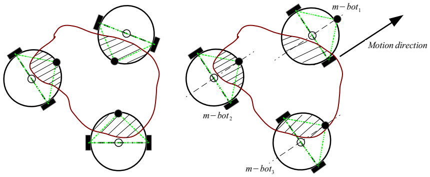
The function l is expressed as follow, with respect to the previous analysis:

$$\begin{cases} l = d_1 + W_b \text{ when it is maximum} \\ l = d_1 + W_b - \frac{W_b T}{2\sqrt{4W_b^2 + T^2}} \text{ when it is minimum} \end{cases} \tag{11}$$

Circular mobile platform with centred wheels axis

When considering a mobile robot with a circular shape (with a radius R) and centred wheels axis (*e.g.* Khepera mobile robot), fixed parameters are defined such as the distance L_2. To ensure stability, conditions to define P_2 position has to be checked. For a circular robotic platform it is assumed that:

- $d_1 = 0$ - the castor wheel is on the front of the robot;
- $W_b = R$ - the robot rear wheels axis is centred relative to the robot platform;
- $T = 2R$ - the robot wheels are on the robot platform side;



(a) Payload lifting (b) Payload transportation

Fig. 7. Payload transportation by cicular mobile robots

- $L_2 = R$ - the half space on robot body will be used for the manipulation mechanism mounting.

In this case, the landing position P_2 that ensures the m-bot stability during all phases is constrained as follow:

$$R - \frac{R}{\sqrt{2}} \leq l \leq R, \tag{12}$$

which is a possible condition that could be ensured. This means the m-bot can support the payload and ensure co-manipulation and transport in a secure way.

In Fig. 8 and 9, P_2 represents the final landing position of the lower point of the end-effector P. This point is defined according to the analysis of the previous section with respect to the m-bot stability criteria. Two clearance parameters, δ_1 and δ_2, are defined in order to avoid collision between P and the robot platform, during payload lifting at position P_3. Constant and variable parameters are defined in Fig. 9.

The position of P_2 is defined according to section III-B and P_3 is defined by the clearances δ_1 and δ_2. The trajectory radius r is equal to the bar lengths l_{AB} and l_{CD}. Using a geometric construction, the center of trajectory could be determined on the lower side of the robot turntable. Fig. 8 presents the geometric construction to obtain the trajectory center and the position of P_1. The trajectory center is obtained by the intersection of both circle C_1 and C_3. α presents the inclination angle of the bars AB and CD during the payload lifting and the initial value α_0 must be well chosen in order to avoid the system blocking state. The normal pushing force generated by robot wheels, is transmitted and converted to a lifting force on the end effector, if and only if $\alpha_0 > 0$. By imposing an initial value of α, P_1 could be found by the intersection of the line passing through the trajectory center and which have an angle α_0 with respect to the horizontal ground. The trajectory radius is then determined as it will be explained in next section.

Trajectory radius determination: To calculate the trajectory radius the method consists in calculating the distances a and b (cf. Fig. 10) and solving the following second order equation:

$$r^2 = (h + r\sin\alpha_0)^2 + (a+b)^2. \tag{13}$$

The first step is to identify the constant a by using geometrical relations into right angle triangles:

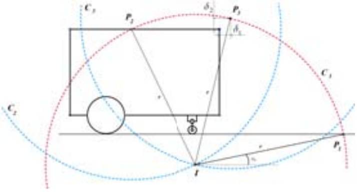


Fig. 8. Determination of the trajectory center and the position of P_1

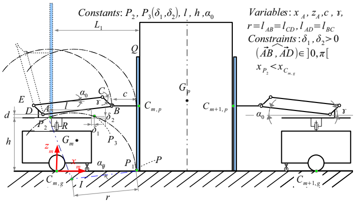

Constants: P_2, $P_3(\delta_1,\delta_2)$, l, h, α_0
L_1
Variables: x_A, z_A, c, γ,
$r=l_{AB}=l_{CD}$, $l_{AD}=l_{BC}$
Constraints: $\delta_1, \delta_2 > 0$
$(\widehat{\vec{AB}, \vec{AD}}) \in]0,\pi[$
$x_{P_2} < x_{C_{m,g}}$
Q
G_p
E
α_0
C
c
γ
$C_{m,p}$
$C_{m+1,p}$
D
A
l
B
d
δ_2
P_2
R
δ_1
h
G_m
P_3
z_m
x_m
α_0
P_1
P
$C_{m,g}$
I
r
α_0
γ
$C_{m+1,g}$

Fig. 9. Dimensions synthesis

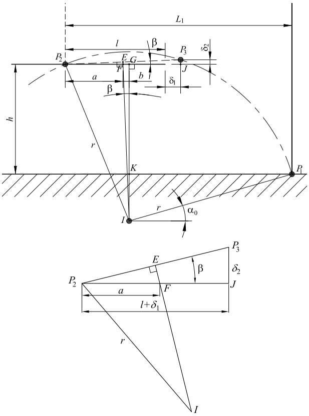


Fig. 10. Determination of the trajectory center I

In triangle P_2P_3J orthogonal in J,

$$\cos\beta = \frac{l+\delta_1}{l_{P_2P_3}}$$

In triangle P_2EF orthogonal in E,

$$\cos\beta = \frac{\frac{l_{P_2P_3}}{2}}{a}$$

This means:

$$\frac{l+\delta_1}{l_{P_2P_3}} = \frac{\frac{l_{P_2P_3}}{2}}{a}, \tag{14}$$

$$a = \frac{(l+\delta_1)^2+(\delta_2)^2}{2(l+\delta_1)}. \tag{15}$$

The second step is to find the constant b by using geometrical relations into right angle triangles:

In triangle FGI orthogonal in G,

$$\tan\beta = \frac{b}{h+r\sin\alpha_0}$$

In triangle P_2P_3J orthogonal in J,

$$\tan\beta = \frac{\delta_2}{l+\delta_1}$$

which means:

$$\frac{b}{h+r\sin\alpha_0} = \frac{\delta_2}{l+\delta_1}, \tag{16}$$

$$b = \frac{\delta_2(h+r\sin\alpha_0)}{l+\delta_1}. \tag{17}$$

Now that the constant term $(a+b)$ of equation (13) is identified, the equation can be reformulated into a second order equation of unknown r. Solving 13 means to solve the following equation:

$$mr^2+nr+p=0 \tag{18}$$

with

$$m = -\frac{[(l+\delta_1)^2+\delta_2^2](\delta_2+2h)\sin\alpha_0}{(l+\delta_1)^2};$$

$$n = \frac{(l+\delta_1)^2\cos^2\alpha_0-\delta_2^2\sin^2\alpha_0}{(l+\delta_1)^2};$$

$$p = \frac{[(l+\delta_1)^2+\delta_2^2][(l+\delta_1)^2+\delta_2^2+4h(\delta_2+h)]}{4(l+\delta_1)^2}$$

Finally r is equal to:

$$r = l_{AB} = l_{CD} = \frac{-m+\sqrt{m^2-4np}}{2n} \tag{19}$$

The distance between P_1 and P_2 can be deduced in function of constant parameters as follows

$$L_1 = \frac{(l+\delta_1)^2+(\delta_2)^2+2\delta_2(h+r\sin\alpha_0)}{2(l+\delta_1)}+r\cos\alpha_0 \tag{20}$$

$$x_{P_1} = x_{P_2}+L_1;\ z_{P_1}=0 \tag{21}$$

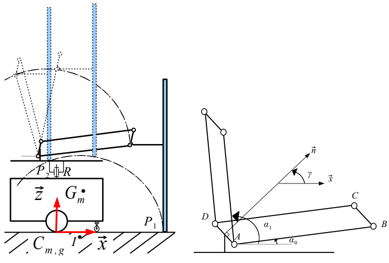
Now the position of A and B can be written as:

$$x_A = x_{P_1}-r\cos\alpha_0-c = x_{P_2}+\frac{(l+\delta_1)^2+(\delta_2)^2+2\delta_2(h+r\sin\alpha_0)}{2(l+\delta_1)}-c \tag{22}$$

$$z_A = h+d = z_{P_2}+d \tag{23}$$

$$x_B = x_A+r\cos\alpha_0 \tag{24}$$

$$z_B = z_A + r \sin \alpha_0 \tag{25}$$



(a)

(b)

Fig. 11. Extreme positions of the parallelogram mechanism

Singular positions: To avoid singular positions of the parallelogram mechanism, $\widehat{BAD}$ must satisfy a constraint along the course between α_0 and α_1 which is:

$$\widehat{BAD}] \in 0, \pi[\tag{26}$$

When this constraint is satisfied along the trajectory between initial and final positions, the parallelogram mechanism would never have a flattened configuration as presented in Fig. 11(a). This constraint implies a suitable choice of γ angle, the angle of the normal vector $\vec{n}$ to segment AB with respect to horizontal.

From Fig. 11(b) one can conclude, to avoid the parallelogram flattening, that γ must be less than $\pi - \alpha_1$ and while considering always $\alpha_0 > 0$:

$$\gamma = \frac{\alpha_0 + \alpha_1}{2} \in [0, \pi - \alpha_1] \tag{27}$$

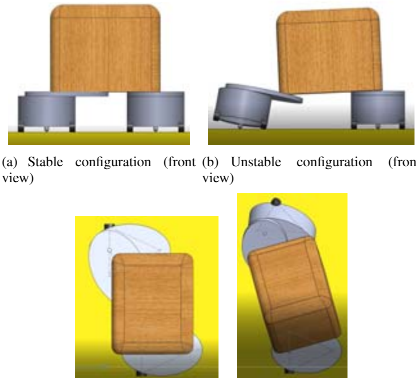
where α_0 and α_1 are the extreme angular positions of the link AB.

IV. Multi-body dynamic system and Manufactured prototypes results

In order to validate the stability analysis, ADAMS multi-body dynamic simulation software was used. Simulation results are presented in Fig. 12.

Using ADAMS software, three robots were positiond around a payload to lift it using the proposed methodology. The results are shown in Fig. 13(a) and 13(b). Videos for simulation are visible under [30].

Two versions of prototypes were manufactured in order to validate the proposed strategy of co-manipulation and transport. Fig. 13(c) presents the first prototype of manipulation mechanism mounted on Khepera mobile robot and Fig. 13(d) presents the second prototypes tested for lifting and transport for both alone as a m-bot or cooperatively as a p-bot. The lifting and transport process by two m-bots is presented in Fig. 13 (c-f).



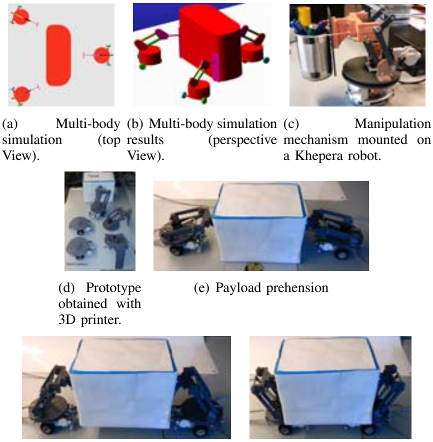
(a) Stable configuration (front view)

(b) Unstable configuration (front view)

(c) Stable configuration (Up view)

(d) Unstable configuration (Up view)

Fig. 12. M-bot Stability Simulation according to the landing position



(a) Multi-body simulation (top View).

(b) Multi-body simulation results (perspective View).

(c) Manipulation mechanism mounted on a Khepera robot.

(d) Prototype obtained with 3D printer.

(e) Payload prehension

(f) Payload lifting

(g) Payload transport

Fig. 13. a, b) Multi-body 3D simulation; c) m-bot prototyping; d, e, f) payload co-lifting.

The mechanism that ensures the co-lifting process is illustrated in Fig. 13 (c-f) based on parallelogram structure that ensures a circular trajectory to lift the payload from the ground and put it on robots platform. Manufactured prototypes allow to experiment the proposed strategy of co-manipulation and co-transportation. The lifting and transport process by two m-bots is presented in Fig. 13(e), 13(f) and 13(g).

Several simulations and experimental validations for lifting are given in [30]. These developed systems will be used for future experiments and validation of the global strategy for co-manipulation and co-transportation proposed in this work.

The proposed co-manipulation strategy and transport were validated using the manufactured prototypes and the videos for experiments can be found in [30].

V. Conclusion

In this paper, it has been considered the problem of payload co-manipulation and transportation using a multi-robot system. The task was defined by several phases achieved by using several m-bots. A mono-robot is mainly composed of two parts: a mobile platform and a manipulation mechanism used to lift and put the payload on robot bodies. The overall system composed of the used mono-robots and the payload is called poly-robot, which is modular and can gather a variable number of m-bots depending on the task to be achieved. The m-bot structure has been studied and the lifting mechanism has been presented in order to obtain a functional system that ensures stability and successful task achievement. A future work will focus on the interaction modeling between the robots and the payloads and a study about using compliant end-effector for better prehension capabilities.

Acknowledgment

This project was achieved with the support of J-C. Fauroux, L. Adouane, Y. Mezouar and Ioan Doroftei for this I would like to acknowledge their contribution to the success of this porposed design and methodology of co-manipulation and transportation process using multiple mobile robotic system.

References

[1] I. F. Vis, “Survey of research in the design and control of automated guided vehicle systems,” European Journal of Operational Research, vol. 170, no. 3, pp. 677–709, 2006.

[2] C. J-P, “Elevateur pour mise en place des equipements de chantier,” Sept.

[3] J.-Y. Wang, J.-S. Zhao, F.-L. Chu, and Z.-J. Feng, “Innovative design of the lifting mechanisms for forklift trucks,” Mechanism and Machine Theory, vol. 45, no. 12, pp. 1892–1896, 2010.

[4] E. Mattar, “A survey of bio-inspired robotics hands implementation: New directions in dexterous manipulation,” Robotics and Autonomous Systems, vol. 61, no. 5, pp. 517–544, 2013.

[5] S. M. Hsiang, G. E. Brogmus, and T. K. Courtney, “Low back pain (lbp) and lifting technique:a review,” International Journal of Industrial Ergonomics, vol. 19, no. 1, pp. 59–74, 1997.

[6] A. Plamondon, A. Delisle, S. Bellefeuille, D. Denis, D. Gagnon, and C. Larivi‘ere, “Lifting strategies of expert and novice workers during a repetitive palletizing task,” Applied ergonomics, vol. 45, no. 3, pp. 471–481, 2014.

[7] R. R. Fox and J. L. Smith, “A psychophysical study of high-frequency arm lifting,” International Journal of Industrial Ergonomics, 2012.

[8] D. Rabinowitz, R. Bridger, and M. Lambert, “Lifting technique and abdominal belt usage: a biomechanical, physiological and subjective investigation,” Safety science, vol. 28, no. 3, pp. 155–164, 1998.

[9] Y. Aiyama, M. Hara, T. Yabuki, J. Ota, and T. Arai, “Cooperative transportation by two four-legged robots with implicit communication,” Robotics and Autonomous Systems, vol. 29, no. 1, pp. 13–19, 1999.

[10] J. S. Bay, “Design of the ”army-ant” cooperative lifting robot,” Robotics & Automation Magazine, IEEE, vol. 2, no. 1, pp. 36–43, 1995.

[11] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella, “Task-oriented motion planning for multi-arm robotic systems,” Robotics and Computer-Integrated Manufacturing, vol. 28, no. 5, pp. 569–582, 2012.

[12] S. Liu, D. Sun, and C. Zhu, “A dynamic priority based path planning for cooperation of multiple mobile robots in formation forming,” Robotics and Computer-Integrated Manufacturing, vol. 30, no. 6, pp. 589–596, 2014.

[13] C. Schou, R. S. Andersen, D. Chrysostomou, S. Bogh, and O. Madsen, “Skill-based instruction of collaborative robots in industrial settings,” Robotics and Computer-Integrated Manufacturing, vol. 53, pp. 72–80, 2018.

[14] M. Abou-Samah and V. Krovi, “Optimal configuration selection for a cooperating system of mobile manipulators,” in ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 1299–1306, American Society of Mechanical Engineers, 2002.

[15] L. Adouane, L. Fort-Piat, et al., “Hybrid behavioral control architecture for the cooperation of minimalist mobile robots,” in Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on, vol. 4, pp. 3735–3740, IEEE, 2004.

[16] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, et al., “Swarmanoid,” IEEE Robotics & Automation Magazine, vol. 1070, no. 9932/13, 2013.

[17] O. Khatib, K. Yokoi, O. Brock, K. Chang, and A. Casal, “Robots in human environments: Basic autonomous capabilities,” The International Journal of Robotics Research, vol. 18, no. 7, pp. 684–696, 1999.

[18] B. H. Wilcox, T. Litwin, J. Biesiadecki, J. Matthews, M. Heverly, J. Morrison, J. Townsend, N. Ahmad, A. Sirota, and B. Cooper, “Athlete: A cargo handling and manipulation robot for the moon,” Journal of Field Robotics, vol. 24, no. 5, pp. 421–434, 2007.

[19] T. Sugar and V. Kumar, “Decentralized control of cooperating mobile manipulators,” in Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, vol. 4, pp. 2916–2921, IEEE, 1998.

[20] H. Yamaguchi, M. Mori, and A. Kawakami, “Control of a five-axle, three-steering coupled-vehicle system and its experimental verification,” in World Congress, vol. 18, pp. 12976–12984, 2011.

[21] B. Hichri, L. Adouane, J. Fauroux, I. Doroftei, and Y. Mezouar,. Flexible co-manipulation and transportation with mobile multi-robot system. Assembly Automation, 2019.

[22] B. Hichri, J. Fauroux, L. Adouane, I. Doroftei, and Y. Mezouar, “Design of cooperative mobile robots for co-manipulation and transportation tasks,” Robotics and Computer-Integrated Manufacturing, vol. 57, pp. 412–421, 2019.

[23] B. Hichri, L. Adouane, J.-C. Fauroux, Y. Mezouar, and I. Doroftei, “Cooperative mobile robot control architecture for lifting and transportation of any shape payload,” in Distributed Autonomous Robotic Systems, pp. 177–191, Springer, 2016.

[24] B. Hichri, J. C. Fauroux, L. Adouane, Y. Mezouar, and I. Doroftei, “Design of collaborative, cross & carry mobile robots” c3bots”,” Advanced Materials Research, vol. 837, pp. 588–593, 2014.

[25] B. Hichri, J.-C. Fauroux, L. Adouane, I. Doroftei, and Y. Mezouar, “Lifting mechanism for payload transport by collaborative mobile robots,” in New Trends in Mechanism and Machine Science, pp. 157–165, Springer, 2015.

[26] R. B. McGhee and A. A. Frank, “On the stability properties of quadruped creeping gaits,” Mathematical Biosciences, vol. 3, pp. 331–351, 1968.

[27] D. E. Orin, R. B. McGhee, and V. Jaswa, “Interactive compute-control of a six-legged robot vehicle with optimization of stability, terrain adaptibility and energy,” in Decision and Control including the 15th Symposium on Adaptive Processes, 1976 IEEE Conference on, vol. 15, pp. 382–391, IEEE, 1976.

[28] B. S. Lin and S.-M. Song, “Dynamic modeling, stability, and energy efficiency of a quadrupedal walking machine,” Journal of Robotic Systems, vol. 18, no. 11, pp. 657–670, 2001.

[29] C. Grand, F. Benamar, F. Plumet, and P. Bidaud, “Stability and traction optimization of a reconfigurable wheel-legged robot,” The International Journal of Robotics Research, vol. 23, no. 10-11, pp. 1041–1058, 2004.

[30] B. HICHRI, “Simulation and experimental results for a group of mobile robots manipulating a payload.” https://www.dropbox.com/sh/0adqqudgvl2w1gp/AADsuTf5UzOSX-4J0XJZKiAaa?dl=0, July 2015.

Mobile Robots Target Reaching and Virtual Structure Navigation Based on Limit-Cycle Method

B. Hichri
Manufacturing Engineering Group
University, Luxembourg
Luxembourg
bassem.hichri@uni.lu

P. Plapper
Manufacturing Engineering Group
University, Luxembourg
Luxembourg
peter.plapper@uni.lu

***Abstract*—This paper addresses Target Reaching problem of mobile robots and obstacle avoidance based on Limit-Cycle Method. The Limit-Cycle method is also used for hidden Targets reaching around the payload. The chosen methodology to achieve optimal positioning and define the robot's targets around the payload to lift it and to transport it while maintaining a geometric multi-robot formation is presented. This appropriate configuration of the set of robots is obtained by combining constraints ensuring stable and safe lifting and transport of the payload. A suitable control law is then used to track a virtual structure in which each elementary robot has to keep its desired position with respect to the payload. Several simulation results validate our proposal.**

***Index Terms*—Cooperative mobile robots, Control architecture, Payload co-manipulation and co-transportation, Robots positioning, Navigation in formation, Virtual structure approach.**

I. Introduction

Compared with single robot setups, multi-robot systems provide more efficient and robust task completion, and enable behaviors having a higher degree of complexity and sophistication. In order to ease transportation tasks, the payload can be adequately distributed among a group of inexpensive robots due to simpler kinematics and architecture and the payload handling dexterity may be increased. The robots may be reconfigured in order to fit a payload of any shape and to adapt to the environment in which they evolve. Each of the robot can be rather simple and be manufactured at a low cost. Additionally, the failure tolerancy of a multi-robot system can be very high provided that spare robots are available to replace damaged robots in the system. There have been a significant researches related to payload transportation using multiple robots [1]–[7], [17].

Multi-robot transportation tasks can be seen as a navigation in formation control problem. This is a classical problem that has attracted the attention of the research community in the last decade [8]. The approaches proposed to solve it can be classified into three main groups: behavior-based approach, leader-follower approach and virtual structure approach. In **behavior-based** approaches [9], [10], a behavior or motion primitive for each entity is designed (e.g., obstacle avoidance, formation keeping, target seeking, trajectory tracking). Then more complex motion patterns can be generated by using a weighted sum according to the relative importance of these behaviors. The main drawback of this approach is the complexity of the group dynamics and as a consequence the convergence to the desired formation configuration cannot be guaranteed. **Leader-follower** approach [8] is a strategy in which a robot is the leader while others act as followers. The main advantage of using this approach is the reduction of the strategy to a tracking problem where stability of the tracking error is shown through standard control theoretic techniques: the leader has to track a predefined trajectory and the followers track the leader with some prescribed offset. A disadvantage of this approach is that there is no feedback from followers to the leader, so that if a follower is perturbed then the formation cannot be maintained, which characterizes a lack of robustness. The last approach is **virtual-structure** (VS) [11], [12] in which the entire formation is considered as a rigid body and the notion of hierarchy does not exist. The control law for each entity is derived by defining the VS dynamics and then translated to the motion of the VS into the desired motion of each robot. The main advantages of this approach are its simplicity to prescribe the behavior of the group and its ability to maintain the formation during maneuvers. However, the potential application is limited by the VS rigidity, especially when the formation shape needs to be frequently reconfigured.

Our goal in the proposed work is to control several mobile robots, called **m-bots**, with a simple mechanical architecture that will be able to autonomously co-manipulate and transport payloads of any shape. The resulting poly-robot system, called **p-bot**, will be able to solve the so-called *removal-man-task* to transport any payload on the top platform of m-bots (dorsal transport). Reconfiguring the p-bot by adjusting the number of m-bots allows to manipulate payloads of any mass, whereas modifying the poses of the m-bots inside the p-bot permits to adjust to any payload shape. The Limit-Cycle method [13], [14] for Target Reaching Problem is used for safe and smooth navigation of one robot and also for the group of robots during the transportation phase. The Limit-Cycle Method is also used if the affected target to mobile robot around the payload is hidden.

Many robotic systems used for objects manipulation and transport can be found in literature. Using different techniques, a group of similar [1], [2] or heterogeneous robots [15] can ensure payloads transport. Different strategies can be found in literature for multi-robot transportation. Pushing strategy proposed in [1] was used while a payload is on the ground. This strategy may face some difficulties depending on the friction generated by the contact surface with the ground and it can also affect the quality of the transported object. Other robots are using grabbing tools [16] for transportation which limits the shape of objects that can be manipulated and requires geometries and shapes that could be gripped by the grippers. Some robots need the human assistance for putting payload on their transport platform such as the Arnold robot presented in [2]. In the proposed work, a strategy based on tightening a payload by a multi-robot system to manipulate it, lift it and autonomously put it on the robots platform autonomously is proposed. To ensure the payload stability during the different phases, an optimal multi-criteria positioning of a set of m-bots around a payload of any shape and mass is proposed. The robots configuration ensures Force Closure Grasping (**FCG**) which allows to ensure stability during the payload lifting and manipulation phase. It ensures also the Static Stability Margin (**SSM**) which maintains the whole system (m-bots and payload) stability during the transportation phase. These two criteria were detailed in [4], [7].

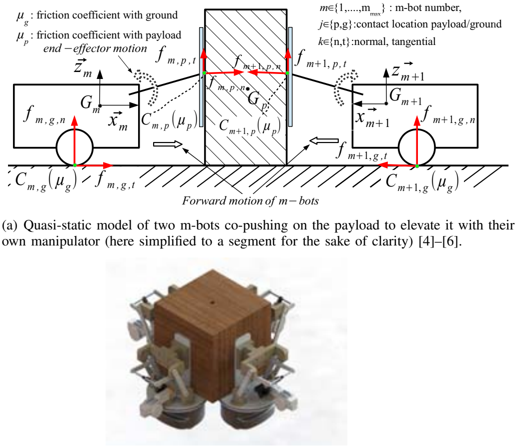
This paper is organized as follow: Section II introduces the paradigm of the C^{3}Bots project; Section III will present the multi-criteria optimization to achieve the targeted multi-robot tasks. It will also present the used controller and Limit-Cycle method for Target Reaching and VS navigation; Section IV presents the results of the simulation results. Finally, Section V is dedicated to the conclusion and some prospects.

II. Paradigm and problem statement

As shortly introduced above, the paradigm of this project consists in co-manipulating and co-transporting a common payload through collaboration between several similar elementary m-bots. Each m-bot is built by connecting a manipulation mechanism on the top platform of a single-axle mobile base [17]. The payload is supported on the edge of this transporting platform. The platform can rotate freely with respect to a central vertical axis on the mobile base. This mobility allows each robot to rotate around itself while maintaining the payload static on its top [5]. The resulting p-bot system (cf. Fig. 1(b)) is thus allowed to rotate around any point on the ground, located at the intersection of all the axle axes, and to translate along any direction.

The manipulator has a parallelogram structure with a single degree of mobility to bring the payload from the ground to the m-bot top platform with a circular trajectory [5], [6].

Before starting the transport task, the m-bots have to achieve the co-manipulation process using the mechanism presented and detailed in [5], [17]. Its role is to hold firmly the payload and to ensure Force Closure Grasping (FCG) [18], [19] to lift the object by applying a sufficient normal force $f_{m,p,n}$ (cf. Fig. 1(a)) which generates a vertical tangential lifting force $f_{m,p,t}$ (cf. Fig. 1(a)).



(a) Quasi-static model of two m-bots co-pushing on the payload to elevate it with their own manipulator (here simplified to a segment for the sake of clarity) [4]–[6].

(b) CAD view of four m-bots transporting a cubic payload on their top [17].

Fig. 1. Co-manipulation of a box by a group of m-bots to ahieve the co-lifting task

III. Overall Proposed Multi-Robot Control Architecture

A. Optimal Positioning According to Multi-Criteria Task Constraints

The proposed overall cooperative manipulation and transport strategy, for payloads of any shape, by a group of m-bots is presented in Figure 2. This figure gives the most important steps to be achieved during this cooperative task. Step 1 allows the payload detection and the estimation of its mass and gravity center position. Step 2 consists in determining the minimum number of m-bots (m_{min}) that could be used to ensure the payload lifting and transport using the equations developed in [5]. Step 3 presents the main contribution of this paper. It is detailed by the flowchart in the right side of Fig. 2 and will be precisely discussed below. The algorithm considers the external shape of the payload as a set of finite positions defined by the chosen step $\Delta\theta$. An initial grasp is then generated based on successive positions that respect inter distances to avoid the collision of m-mbots. Than the algorithm will run through all possible configurations to output a final optimal positioning of the robots [4]. In the proposed strategy, the m-bots positioning is optimal when **F**orce **C**losure **G**rasping (**FCG**), **S**tatic **S**tability

Margin (**SSM**) and **R**estricted **A**reas (**RA**) are ensured. Finally, Step 4 corresponds to target reaching phase and multi-robot transport of the payload toward the assigned final pose.

More details concerning the Positioning algorithm could be found in [4], [5], [7]

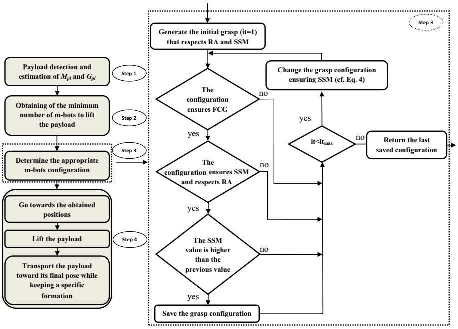

Payload detection and estimation of M_{pl} and G_{pl}
Step 1
Obtaining of the minimum number of m-bots to lift the payload
Step 2
Determine the appropriate m-bots configuration
Step 3
Go towards the obtained positions
Lift the payload
Step 4
Transport the payload toward its final pose while keeping a specific formation
Step 3
Generate the initial grasp (it=1) that respects RA and SSM
The configuration ensures FCG
no
yes
The configuration ensures SSM and respects RA
no
yes
The SSM value is higher than the previous value
no
yes
Save the grasp configuration
Change the grasp configuration ensuring SSM (cf. Eq. 4)
yes
$it<it_{max}$
no
Return the last saved configuration

Fig. 2. Flowchart given the sequenced steps for the co-manipulation and co-transportation of any payload shape

B. Target reaching and Obstacle avoidance

At any moment of the formation motion we can determine the robots positions with respect to the object position and orientation. During the transportation phase, the robots have to track a dynamic target defined with respect to the payload center of mass.

1) Control law: Considering a unicycle mobile robot, the state vector $X_m = (x_m, y_m, \theta_m)^T$ denotes the position of the m^{th} robot center of mass $G_m(x_m, y_m)$ and the orientation θ_m of the robot with respect to $\vec{x}$ axis of the global reference frame. The m-bot control inputs are the forward velocity V and the angular velocity ω.

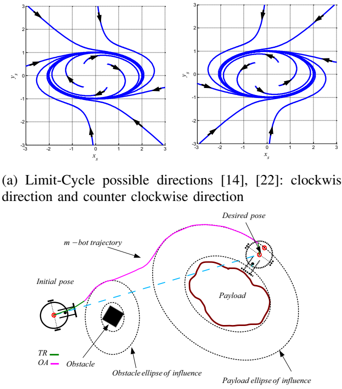
Let e be the error between the m-bot current pose and the desired pose $X_{dm} = (x_{dm}, y_{dm}, \theta_{dm})^T$ defined by: $e = X_{dm} - X_m$. The used control law [20] is given by (1):

$$\begin{aligned} V_m &= V_{max} - (V_{max} - V_d)e^{-(d_m^2/\sigma^2)} \\ \omega_m &= \omega_{Sm} + k\theta_m \end{aligned} \tag{1}$$

- V_m and ω_m are the linear and angular velocities of the m-bot,
- V_{max} is the maximum linear speed of the m-bot,
- V_d is the desired velocity of the p-bot and it is considered as constant,
- $d_m = \sqrt{e_x^2 + e_y^2}$ is the current distance between the m^{th} robot and its desired target,
- ω_{Sm} is the angular velocity of set point angle θ_{Sm} applied to the robot in order to reach the desired goal: $\omega_{Sm} = \dot{\theta}_{Sm}$,
- σ, k are the control law gains (positive constants).

2) Limit-Cycle method for obstacle avoidance and target reaching: The control law used to simulate the obstacle avoidance for desired targets reaching in the proposed work uses the Limit Cycle method (cf. Fig 3(a)) [13], [14] which is a path planning method developed initially for obstacle avoidance behavior and it is one of the trajectory methods defined by differential equations [21]. This technique has been adopted in this paper to perform both: Target Reaching phase and Virtual Structure navigation (cf. Fig 6(b)). The differential equations of the elliptic limit-cycles are:

$$\begin{aligned} \dot{x}_s &= \quad m(By_s + 0.5Cx_s) + x_s(1 - Ax_s^2 - By_s^2 - Cx_sy_s) \\ \dot{y}_s &= \quad -m(Ax_s + 0.5Cy_s) + y_s(1 - Ax_s^2 - By_s^2 - Cx_sy_s) \end{aligned} \tag{2}$$



(a) Limit-Cycle possible directions [14], [22]: clockwise direction and counter clockwise direction

(b)

Fig. 3. Target Reaching strategy with obstacle avoidance using Limit-Cycle method

with $m = \pm 1$ according to the avoidance direction (clockwise or counter-clockwise, cf. Fig. 3). (x_s, y_s) corresponds to the position of the m-bot according to the center of the ellipse. The variables A, B and C are given by:

$$A = (\sin(\Omega)/b_{lc})^2 + (\cos(\Omega)/a_{lc})^2 \tag{3}$$

$$B = (\cos(\Omega)/b_{lc})^2 + (\sin(\Omega)/a_{lc})^2 \tag{4}$$

$$C = (1/a_{lc}^2 - 1/b_{lc}^2)\sin(2\Omega) \tag{5}$$

where a_{lc} and b_{lc} characterize respectively the major and minor elliptic semi-axes and Ω gives the ellipse orientation when it is not equal to 0.

The set point angle that the robot must follow to avoid the obstacle is given by:

$$\theta_{S0a} = \arctan(\frac{\dot{y}_s}{\dot{x}_s}) \tag{6}$$

The control architecture for the m-bot navigation is presented in Fig. 4. This architecture, with specific elementary controller blocks (attraction to the target, obstacle avoidance), aims to manage the interactions among elementary controllers while guaranteeing the stability of the overall control to obtain safe and smooth navigation.

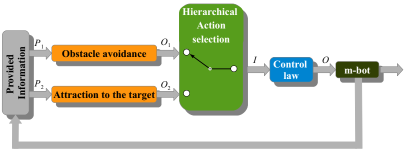


Fig. 4. Control architecture for mobile robot navigation during the target reaching phase (cf. the first phase of step 4 in Fig. 2)

After positioning the m-bots, they must keep their desired position (x_{dm}, y_{dm}) with respect to the payload center of mass G_{pl} and must respect the following conditions during the task achievement:

$$\begin{aligned} x_{dm} &= x_{G_{pl}} + l_{xm}\cos\theta_{dm} - l_{ym}\sin\theta_{dm} \\ y_{dm} &= y_{G_{pl}} + l_{xm}\sin\theta_{dm} + l_{ym}\cos\theta_{dm} \end{aligned} \qquad (7)$$

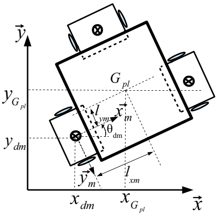


Fig. 5. Robot position relative to the object

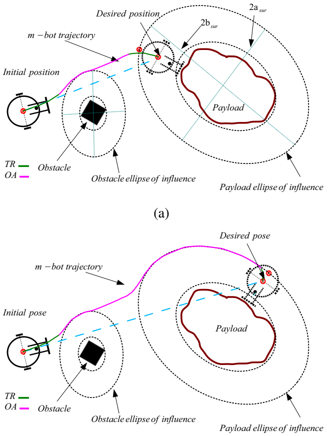
where l_{xm} and l_{ym} (cf. Fig. 5) are the relative distances G_mG_{pl} according the axis $\vec{x}_m$ and $\vec{y}_m$ respectively. These two distances define rigid links maintaining the robot position with respect to G_{pl}.

The robots must avoid the collision with the payload if its target is not apparent to it. The limit-cycle method was adopted to avoid the m-bots collision with the payload as presented in Fig. 6. The payload is assumed to be surrounded by an ellipse and an obstacle, if exists, also is surrounded by an ellipse. These ellipses are presented in Cartesian form with an orientation Ω and semi-axes a_{sur} and b_{sur} (cf. Fig. 6(a)). An ellipse of influence is then defined having the same center and orientation of the ellipses surrounding the payload or the obstacle with a semi axes a_{inf} and b_{inf} respecting the following equation:

$$\begin{aligned} a_{inf} &= a_{sur} + R + Marg \\ b_{inf} &= b_{sur} + R + Marg \end{aligned} \qquad (8)$$

Where R is the robot radius and $Marg$ is a security margin to avoid the collision between m-bots and the payload or obstacle.

The m-bot will proceed by the payload avoidance using the limit-cycle method until a position error ε, between the robot real position and the projection of the desired position on the ellipse of influence, is satisfied. This error, if it is satisfied, allows to switch the robot controller from obstacle avoidance to target attraction (the intermediate projected position is presented by red points in Fig. 6(b)). In case where the desired position is apparent to the m-bot, then the robot will be attracted first to the intermediate position and then goes to its target. The intermediate position are defined in function of the final desired orientation of the robot. This condition allows the m-bots to reach the final position with the required orientation.

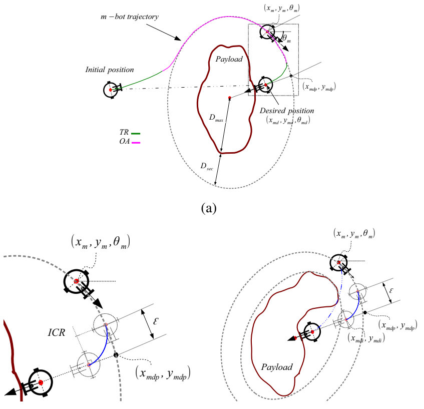


(a)

(b)

Fig. 6. Target Reaching strategy with obstacle avoidance: a) apparent position reaching; b) hidden position reaching

In order to ensure a smooth and secure m-bot evolving during the target reaching phase, a suitable choice of ε is required. As presented in Fig. 6(a), the payload is surrounded by an ellipse of influence that will be followed by the m-bot during the attraction to the target if the desired position is not apparent. ε is defined as follow: If ε is equal to zero, then the robot would reach the position of the projection of desired target and then turn around itself to reach the final goal. This allows to have a discrete motion of the robot. In order to avoid this, ε is chosen with a strictly positive value that does not exceed the robot platform radius. The reasons why this value is limited to R is that the robot is not so far from the position of controller switch and to avoid the collision between the robot and the payload during the final desired target reaching (cf. Fig. 7(b)).



(a)

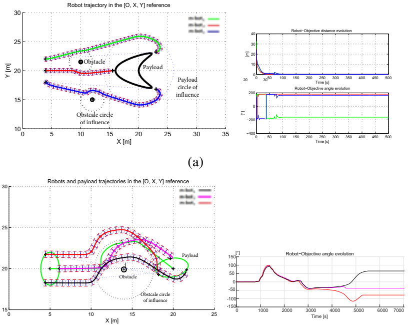
(b)

(c)

Fig. 7. General principle of smooth target Reaching

IV. Proposal Validation

A. Simulation Results



(a)

(b)

Fig. 8. Simulation results: a) Robots positioning simulation around a payload and corresponding system of wrenches; b) Target reaching simulation of three m-bots and their objective distances and orientations evolution; c) Collective payload co-transportation and their objective distances and orientations evolution

The simulations were simulated by using an Intel Core i5 2400 CPU 3.1 GHz system. The controller parameters are set to k=22 and $\sigma = 0.1$. These parameters were chosen to obtain a safe and smooth trajectory, fast response and velocity value within the limits of the m-bots capacities.

Fig. 8(a) and Fig. 8(b) show respectively the trajectories of the mobile robots during Target Reaching and the transportation phases. It can be noted the smoothness of the vehicle trajectories along the navigation and the non-collision with obstacles thanks to the limit-cycle method.

The right graph of Fig. 8(a) shows respectively (from top to down): the values of the position errors e_m between each m-bot and its assigned virtual target around the payload; the value of the angular set-point θ_{Sm} which is tracked with stable way by each m-bot (cf. equation 6). It shows the convergence of the position error to zero and it shows the evolution of the robot trying to reach its target. Fig. 8(b) illustrates the navigation in formation of the whole structure while maintaining the assigned desired position of each m-bot w.r.t. the payload (according to Eq. 7) to ensure the whole system stability and to avoid the transportation task failure. The position error evolution is kept close to zero. For both Target Reaching and Transportation, the ellipse of influence was considered as a circle ($a_{lc} = b_{lc}$) since the obstacles have a circular shape. The radius of the circle of influence was chosen in a manner that the obstacle avoidance is guaranteed by keeping a safety margin.

V. Conclusion

The main challenge addressed in this paper is the use of the obstacle avoidance controller, based on limit-cycles, which is used for two aspects: firstly when each m-bot aims to reach its position around the payload (the robot may need to avoid other robots or any other obstacles to reach its assigned pose); secondly when the p-bot is in the navigation phase and has to avoid any obstructing obstacle. The p-bot navigation arises also interesting issues related to multi-robot navigation in formation. It is planed in near future to perform more experiments of the overall defined strategy for cooperative payload co-lifting and co-transportation.

Acknowledgment

This project was achieved with the support of J-C. Fauroux, L. Adouane, Y. Mezouar and Ioan Doroftei for this I would like to acknowledge their contribution to the success of this proposed design and methodology of co-manipulation and transportation process using multiple mobile robotic system.

References

[1] L. Adouane, L. Fort-Piat, et al., "Hybrid behavioral control architecture for the cooperation of minimalist mobile robots," in Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, vol. 4, pp. 3735–3740, IEEE, 2004.

[2] M. Abou-Samah and V. Krovi, "Optimal configuration selection for a cooperating system of mobile manipulators," in ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 1299–1306, American Society of Mechanical Engineers, 2002.

[3] S. Kernbach, E. Meister, F. Schlachter, K. Jebens, M. Szymanski, J. Liedke, D. Laneri, L. Winkler, T. Schmickl, R. Thenius, et al., "Symbiotic robot organisms: Replicator and symbrion projects," in Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, pp. 62–69, ACM, 2008.

[4] B. Hichri, L. Adouane, J.-C. Fauroux, Y. Mezouar, and I. Doroftei, "Flexible co-manipulation and transportation with mobile multi-robot system," Assembly Automation, 2019.

[5] B. Hichri, J. Fauroux, L. Adouane, I. Doroftei, and Y. Mezouar, "Design of cooperative mobile robots for co-manipulation and transportation tasks," Robotics and Computer-Integrated Manufacturing, vol. 57, pp. 412–421, 2019.

[6] B. Hichri, J.-C. Fauroux, L. Adouane, I. Doroftei, and Y. Mezouar, "Lifting mechanism for payload transport by collaborative mobile robots," in New Trends in Mechanism and Machine Science, pp. 157–165, Springer, 2015.

[7] B. Hichri, L. Adouane, J.-C. Fauroux, Y. Mezouar, and I. Doroftei, "Cooperative mobile robot control architecture for lifting and transportation of any shape payload," in Distributed Autonomous Robotic Systems, pp. 177–191, Springer, 2016.

[8] Z. Peng, Formation Control of Multiple Nonholonomic Wheeled Mobile Robots. PhD thesis, Ecole Centrale de Lille, 2013.

[9] J. Vilca, L. Adouane, Y. Mezouar, and P. Lebraly, "An overall control strategy based on target reaching for the navigation of an urban electric vehicle," in Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pp. 728–734, IEEE, 2013.

[10] A. Benzerrouk, L. Adouane, L. Lequievre, and P. Martinet, "Navigation of multi-robot formation in unstructured environment using dynamical virtual structures," in Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 5589–5594, IEEE, 2010.

[11] A. Sadowska, T. v. den Broek, H. Huijberts, N. van deWouw, D. Kosti ́c, and H. Nijmeijer, "A virtual structure approach to formation control of unicycle mobile robots using mutual coupling," International Journal of Control, vol. 84, no. 11, pp. 1886–1902, 2011.

[12] H. Mehrjerdi, J. Ghommam, and M. Saad, "Nonlinear coordination control for a group of mobile robots using a virtual structure," Mechatronics, vol. 21, no. 7, pp. 1147 – 1155, 2011.

[13] J. Vilca, L. Adouane, and Y. Mezouar, "Adaptive leader-follower formation in cluttered environment using dynamic target reconfiguration," in Distributed Autonomous Robotic Systems.

[14] L. Adouane, "Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation," in 9th Conference on Autonomous Robot Systems and Competitions, 2009.

[15] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, et al., "Swarmanoid," IEEE Robotics & Automation Magazine, vol. 1070, no. 9932/13, 2013.

[16] O. Khatib, K. Yokoi, O. Brock, K. Chang, and A. Casal, "Robots in human environments: Basic autonomous capabilities," The International Journal of Robotics Research, vol. 18, no. 7, pp. 684–696, 1999.

[17] B. Hichri, J. C. Fauroux, L. Adouane, Y. Mezouar, and I. Doroftei, "Design of collaborative, cross & carry mobile robots" c3bots"," Advanced Materials Research, vol. 837, pp. 588–593, 2014.

[18] T. Yoshikawa, "Multifingered robot hands: Control for grasping and manipulation," Annual Reviews in Control, vol. 34, no. 2, pp. 199–208, 2010.

[19] Y.-H. Liu, "Qualitative test and force optimization of 3-d frictional form-closure grasps using linear programming," Robotics and Automation, IEEE Transactions on, vol. 15, no. 1, pp. 163–173, 1999.

[20] A. Benzerrouk, L. Adouane, and P. Martinet, "Stable navigation in formation for a multi-robot system based on a constrained virtual structure," Robotics and Autonomous Systems (RAS), vol. 62, no. 12, pp. 1806 – 1815, 2014.

[21] A. Stuart and A. R. Humphries, Dynamical systems and numerical analysis, vol. 2. Cambridge University Press, 1998.

[22] L. Adouane, A. Benzerrouk, and P. Martinet, "Mobile robot navigation in cluttered environment using reactive elliptic trajectories," in 18th IFAC World Congress, (Milano-Italy), August 28, September 2 2011. 9

Unconventional path planning for a serial kinematics robot with Reinforcement Learning using the example of the wire loop game

1st Prof. Dr.-Ing. Rainer Müller
CEO, Chair of Assembly Systems
ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gGmbH
Saarbrücken, Germany
rainer.mueller@zema.de

2nd Ali Kanso M.Sc
Chair of Assembly Systems
ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gGmbH
Saarbrücken, Germany
ali.kanso@zema.de

3rd Stefan Marx B.Eng
Chair of Assembly Systems
ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gGmbH
Saarbrücken, Germany
stefan.marx@zema.de

***Abstract*— Modern customer-specific production systems in small and medium-sized enterprises are facing the challenges to have a high flexibility and adaptability and simultaneously be economically efficient. An automated, adaptive motion planning for industrial robot via reinforcement learning and simulation is necessary to fill this gap without expert persons. In this paper we represent motion planning method based on reinforcement learning which adapts to variations in the robot's environment, and therefore efficient when implementing new similar tasks. The Method is evaluated in use case wire loop game. The use case aims to move the industrial robot along a metal wire without any contacts between the wire and the loop.**

Keywords—industrial robot, motion planning, reinforcement learning, Q-learning, wire loop game

I. Introduction

Over the last few years, the trend can be observed that product life cycles are shortening or that products frequently get updates such as a facelift of an automobile. For production and assembly lines this fact implies that they have to be very flexible and easy to adjust. This poses a problem especially for handling processes, whereby a number of influences must be taken into account. As a result, in the past a lot of handling processes were automated with problem adapted, individual and expensive solutions. [1] Nowadays, when high level of automation is intended, industrial robots are more and more implemented in the production and assembly lines to generate high flexibilities. These robots are freely programmable and there are some common ways to do this which all more or less involve the problem that they are not flexible and adaptive regarding to changes of the environment. Considering a welding application, where a robot follows a fixed programmed nonlinear path along the edge of two components, the process will lead to poor results when the components get shifted relatively to the robot, or the components have changed due to an update. Teaching a new nonlinear path can be very time consuming and complex when using common ways to realize the motion planning for an industrial robot, which leads to the question, if there are ways to implement an adaptive motion planning whereby the flexibility of a robot increases and enable it to react on variations in its direct environment.

In this paper, we present one possible solution to this question based on a reinforcement learning algorithm called Q-learning. In addition, our approach contains a camera to observe the environment of the robot visually and a virtual environment to reduce the operating time and enable offline training of the reinforcement learning system. During the training phase the systems gains knowledge about how to solve a certain motion task, which will be then adaptable to further similar motion tasks.

For a demonstrative use case we implemented this system with the Universal Robot UR10e, an HRC capable robot with six degrees of freedom to play the well-known skill-based wire loop game. The goal of the game is to avoid any contact between a metal loop and a metal wire while moving the loop along the wire from a start to an end point (Fig. 1 a). Based on a single picture of the wire the reinforcement learning system learns to play wire loop game virtually and teaches the robot to navigate along the wire without any further knowledge of the wire configuration or a fixed programmed motion plan. After a certain training phase, the system is also able to generalize the problem to a set of well-known states, which allows the robot to immediately play arbitrary wire configurations without any additional learning.

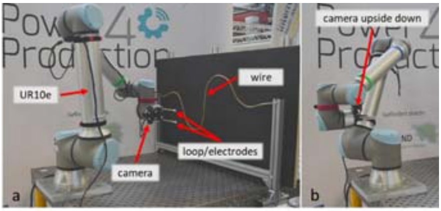


Fig. 1 a) Experimental Setup, b) Image taking position of the robot

The realization of this project took place in the context of the seminar „Cognitive Collaborative Robotics“ (CoCoRo) at the University of Saarland during the winter seminar 2019/20 supervised by the ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gGmbH in cooperation with the DFKI - Deutsches Forschungszentrum für Künstliche Intelligenz GmbH.

II. State of the Art

In general, there are several common ways to program an industrial robot. Haun [2] divides the programming modes in the first instance in online and offline methods. For online programming, the robot is physically needed to program it. Whereas with the offline method a usage of the robot is unnecessary to create a program. Furthermore, a distinction is made between the following types of programming:

- Programming through examples: The end effector of the robot is guided along the desired path. In doing so a sufficient number of path points (position and orientation) are stored in the computer.

- Programming through training: Here the robot system is shown the action to be performed. Sensors picking up the action and the robot system repeatedly executes the action until it satisfies certain quality criteria, for example with regard to accuracy, speed, etc.
- Robot oriented textual programming: The robot gets controlled by a program which contains explicit motion commands e.g. like "move linear from point A to point B"
- Task oriented textual programming: Here is only told to perform a certain task without the programming of fixed motion plans. The interaction of sensors and intelligent programs enable the motion planning on basis of an environment model. [2]

With respect to the definitions above, motion planning based on Q-learning can be classified as an offline task oriented programming method.

The concept of reinforcement learning is based on the idea of Markov Decision Processes (MDPs). MDPs represent a framework for problems, where an agent/system tries to maximize its received reward when solving this problem [3]. MDPs have been used to solve several motion planning tasks e.g. for mobile robots [4, 5]. Furthermore, there exist some solutions where reinforcement learning strategies are used for motion planning of different robots. A frequently used method is the Deep Q-learning or more general Deep Reinforcement Learning, where neural networks are trained with reinforcement learning algorithms like Q-learning [6–9]. Reinforcement learning is also combined with other tools like a fuzzy controller [10]. Less powerful but easier to implement are solutions purely based on Q-learning as described in Meyes et al. [11] which also act as foundation of this paper.

Motion planning with the help of Q-learning is presented here as an alternative way to program a serial kinematic robot compared to common options described above. As advantages of programming with artificial intelligence one hopes for more flexibility and better adaptability of the robot when implementing new tasks or operating in a dynamic environment. However, the solution presented here should not be seen as a ready-made solution for the industry, but only as an overview of a possible approach to implementing and presenting the possibilities that result from the use of Q-learning.

The used Q-learning algorithm represents a model-free method, which means that the system does not have to be given any further knowledge about the effects of its executed actions or the concept of the wire loop game. The system contains no information about the new state it reaches after performing an action. For the wire loop game, it is not necessary to look ahead, because there are no illegal states the system could reach, which would lead into a total failure of the learning process. [12]

To understand the idea of the q-function the model in Fig. 2 can be used. When starting in an arbitrary state s the system, or in our specific application the robot, switches to next state s' by performing an action a (Fig. 2). This action *a* releases some reinforcement *r* (reward) to the system. In this scenario the Q-function models the quality of this state-action pair by mapping them with values called Q-values, where $Q\colon S \times A \rightarrow \mathbb{R}$.

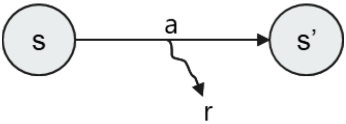


Fig. 2 Cut-out of MDP-chain

To calculate or rather to learn and approximate the Q-values an equivalent of the Bellmann equation can be used.

$$Q(s,a) = (1-\alpha) * Q(s,a) + \alpha * (\mathrm{r} + \gamma * max_{a'}Q(a',s'))$$

In principle the equation contains two parts.

- *Q(s,a)* represents the actual Q-value of the current state *s* when performing action *a* and therefore the already known information.
- In contrast to that *(r + γ*maxQ(s',a')* describes new information composed of the received short-term reward *r* and an estimate of the expected optimal long-term reward.
- The parameter *α* comprises values in the interval of 0 to 1 and is called the learning rate. It determines to what extent newly acquired information overrides old information and thus how much the q-learning algorithm is prone to trust the old already known information. This becomes clearer when observing the extreme values/boundary values of *α*. When setting *α=0* the Q-values would not change at all. The system exclusively relies on its prior knowledge to solve the problem and will, in case of an untrained system, fail the task. On the other hand, *α=1* will make the system consider only the most recent information and ignoring current q-values when updating them.
- The discount factor *γ* determines the importance of future rewards and also can be adjusted in the interval of 0 to 1. *γ=0* represents a short-sighted strategy when updating the Q-values whereas *γ=1* will make the system strive for a high long-term reward. [3, 11]

III. Approach/Method

The three main units for the use cases system are image acquisition, the simulation tool and the control system (Fig. 3).

Image acquisition is done by a Logitech c920 HD Pro camera, mounted on the robot end effector. The images are imported to the ROS-industrial software by the means of an integrated OpenCV package. ROS also integrates the image processing, the virtual environment and the Reinforcement Learning Brain. As the wire is configured in one plane, the image processing and simulation only have to operate in the two-dimensional space. The image processing converts the incoming picture into pixel path array. Based on this array the virtual environment simulates the possible movements of the robot in a simplified two-dimensional model. The environment closely interacts with the Q-learning part, from which it receives the next actions to be performed.

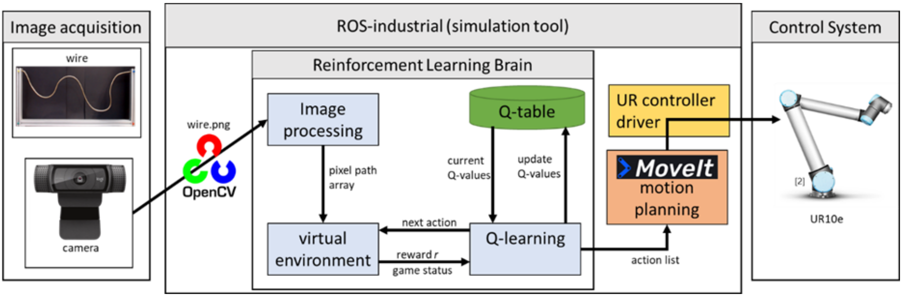


Fig. 3 Structure of the intelligent system in use case wire loop game [13-15]

In response it sends a reward to the Q-learning, which is used to update the Q-values in the Q-table. When the Q-learning was successful playing the whole wire, it creates a list containing all actions the robot has to perform to play the real wire loop game. The necessary motion planning is visualised in the MoveIt package, which is integrated in the ROS software. The provided universal robot drivers for the e-series controller then enable the communication with the robot UR10e.

In general, this paper follows an approach closely inspired by Mayes et al. [11] especially concerning the basic idea and structure of the virtual environment and the Q-learning algorithm.

IV. Image Processing

The picture of the wire has to be edited and transformed into an array in order to make it usable for the minimalistic virtual environment. The aim of the image processing is to generate a pixel path with the exact width of one pixel and then transfer it to an array containing ones for the wire and zeros for the free space around of it.

Because of the chosen setup (Fig. 1 b) the camera takes pictures upside down. The first editing process is to rotate 180 degrees and crop the original image (Fig. 4 a). The cropping is essential since the aluminium frame, in which the wire is mounted, complicates subsequent processing steps because of its low contrast to the wire. With the OpenCV library and our developed functions to optimize the image we realized the image processing in the following:

Conversion to Grayscale: The reduction from RGB channels to only one grayscale channel increases the complexity of the picture and makes it easier to separate the wire form the background (Fig. 4 b).

Threshold function: We then apply gaussian blur for noise reduction, followed by a threshold function to receive a binary image, where the wire is represented by white and the background by black pixels (Fig. 4 c).

Dilation: With a dilation function the edges of the wire are then smoothened.

Average functions: The goal of these two steps is to reduce the width of the wire to one pixel. Therefore, we first looked column-wise for the average row value of the white pixels and added them to an empty black image. The resulting images has some bigger gaps between the pixels where the wire runs vertical (Fig. 4 d). To fill these gaps the same process is done row-wise (Fig. 4 e).

Cost function for shortest path: Last operation in the pixel space is the implementation of some kind of cost function , which searches for the shortest path through this image and ensures that all missing pixels are added, and that the width of the resulting path is exactly one pixel (Fig. 4 f). Finally, the image gets converted into an array as described above.

V. Virtual Environment

The final pixel path array is then the foundation for the virtual environment, which simulates the wire loop game and

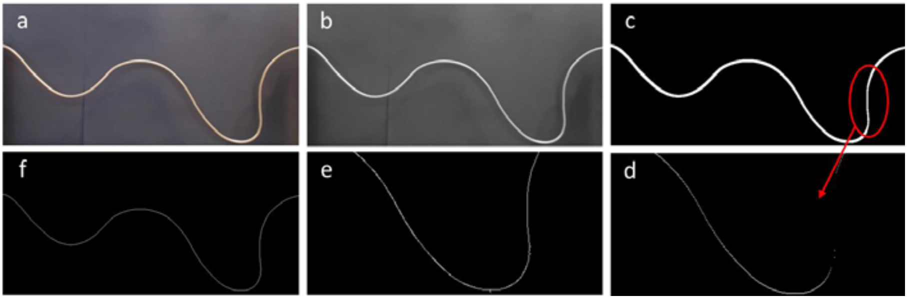


Fig. 4 Step-by-step illustration of the image processing

dramatically reduces the time for learning a new wire configuration. Without the simulation the q-learning had to be practiced with the physical robot and wire setup, which would be very time consuming with an estimated number of required episodes of over 100.000. Even though the virtual environment of the use case wire loop game is only represented by a series of small two-dimensional figures, it supplies all relevant information to transfer the game to the robot.

The whole process of simulation takes place in small 5x5 matrixes as shown in (Fig. 5). Basically, they get extracted out of the large pixel path array, what in fact reduces the perception of the Q-learning but helps to generalize the problem. If the q-table has been fed with enough different states it should be possible to reconstruct almost every new wire configuration with a sequence of these states. In one 5x5 matrix the white Ns represent free space, whereas the red Ws show the wire path. Initially the position of the loop and the first local goal have to be added to the matrix. The two electrodes of the end effector are illustrated by two blue Ls, and local goals by a green G. The loop is always positioned so that its centre, also called tool centre point (TCP), is always congruent with the centre of the 5x5 matrix. The local goals give the direction to the Q-Learning along the wire. A local goal counts as reached if it is on top of the TCP. They are then replaced in 3x3 window around the TCP in the direction of the global goal.

As in Meyes et al. [11], we define six possible actions which can be executed to explore the environment. More precisely, the q-learning can choose the actions move one pixel right, down, left, and up or to rotate clockwise and anticlockwise by 45 degrees. To actually transmit these actions to the robot a transformation matrix of pixel coordinates to robot flange coordinates has to be calculated. This procedure is described in detail in (chapter VII). Fig. 5 shows an example of how the actions move down, and right are realised in the 5x5 matrix. From matrix b to matrix c the process of reaching and replacing a local goal can also be observed. According to the description in chapter II every action releases a reward to the q-learning. The exact values of the possible rewards are depending on the reached new state and are set out in the so-called reward-function. Here we defined the following four reward values:

- The system gets punished with a value of -50 for every action when reaching a valid state. This is to later minimize the total number of actions
- When having a collision between the loop and the wire the system is punished by -100.
- Moving out of boundaries also leads to a negative reward of -100.
- If a local goal is reached the q-learning receives a reward of +100

The aforementioned moving out of bounds appear in two situations. First one is moving the loop out of the environment restricted by the pixel path array. The second case is, when the local goal is moving out of the 5x5 matrix after an action.

VI. Q-LEARNING

To implement the Q-learning a q-function and a Q-table have to be built. The Q-function is adopted in the same way as described in chapter II. In the Bellmann equation we set the learning rate α to 0.3. As a consequence, the system considers new information but only adapts slowly to these. When choosing this value, it is assumed that the system will face a lot of identical states in form of a 5x5 matrix and that the optimal action for these is every time the same. Therefore, a rather small learning rate won't overwrite the current knowledge to fast when observing the environment and also compensates weaknesses in the reward function. The discount factor γ is defined to 0.9 what means that the system strives for a long-term strategy. We decided on this strategy because in the end we want the robot to play the wire loop game with a minimum number of actions.

As mentioned before the Q-learning algorithm also needs a Q-table, which has the function of a look-up table. For every state observed by the system and stored in the Q-table, it contains one Q-value for each possible action the system can perform. In terms of the wire loop game the Q-table is build up as a python dictionary. Keys represent all states (5x5 matrix) observed by the system and refer to the q values. To generate a unique key for every state we put the representative numbers for the four possible configurations of the matrix entries row-wise together. This would be enough if the wire is never bend backwards (Fig. 6 a) or has an angle of 90 degrees to the straight line between start and end point of the wire (Fig. 6 b). For these cases it could be possible that due to the minimization to a 5x5 matrix one state could occur twice in one wire configuration, but the robot has to move in opposite directions for both cases. To solve this problem, we added an id number for the last movement, the robot made to reach the respective state, to the key.

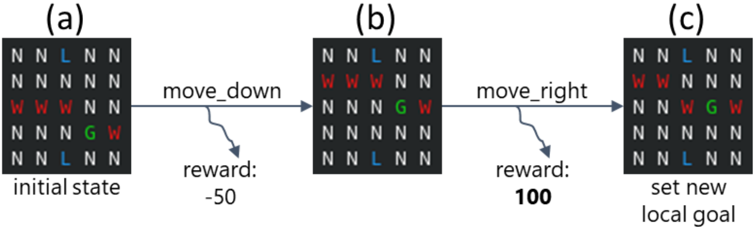


Fig. 5 Illustration of the modifications in the virtual environment after performing the actions move_down and move_right and reaching a local goal

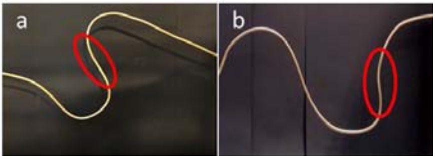
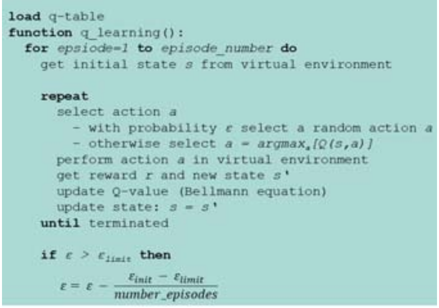


Fig. 6 Wire configurations with difficult passages

Next step is to integrate Q-function and Q-table in a working algorithm, what is done as shown in the pseudo-code representation below (Fig. 7). Q-learning is an iterative process and the q-values converge approximately to the true q-values when enough iterations are run through [16]. To guarantee an adequate number of learning episodes we set up a simple for-loop. Afterwards we extract the initial state out of the pixel path array, in which we want to perform our first action. What follows is an infinite loop, which only gets terminated if game fails e.g. when touching the wire or moving out of the bounds of the pixel path array. Then a new episode will be started. Inside this loop the algorithm selects an action depending on two parameters. With the probability ε a random action is chosen. If this is not the case the next action is determined out of the q-table. For unknown states random q-values are initialized and written into the q-table, which also results in a random action.

In principle the parameter ε is required to solve the exploration-exploitation dilemma described in [17]. To gain progress the q-learning algorithm needs to explore its environment, to find an optimal solution for the wire loop game or in general for the problem to which it is applied. On the other hand, a certain degree of exploitation is needed to reach regions which are less explored then others and to validate already learned q-values. The so-called ε-greedy exploration ensures that a minimum amount of exploration no matter how well the q-values are already approximated to the true q-values. With increasing number of episodes ε gets decreased more and more so that the algorithm skips areas faster which are already explored sufficient.

After the selection of an action, it is performed in the virtual environment, which then returns the reward and the new state. Subsequently the q-values are updated with the help of the Bellmann equation. Finally, the state is updated to the new state *s'*.



```
load q-table
function q_learning():
  for epsiode=1 to episode_number do
    get initial state s from virtual environment

    repeat
      select action a
        - with probability $\varepsilon$ select a random action a
        - otherwise select $a = argmax_a[Q(s,a)]$
      perform action a in virtual environment
      get reward r and new state s'
      update Q-value (Bellmann equation)
      update state: s = s'
    until terminated

    if $\varepsilon > \varepsilon_{limit}$ then
      $\varepsilon = \varepsilon - \frac{\varepsilon_{init} - \varepsilon_{limit}}{number\_episodes}$
```

Fig. 7 Pseudo code of the Q-learning algorithm

One problem when first starting the q-learning with the described code, was that no matter how many episodes the system has been trained, it was not able to reach the global goal. Reason for this is that after an episode was terminated the algorithm initialized the starting position as the first state. As a solution, a smaller area of interest is defined, in which the algorithm learns the wire. Every time a number of local goals x are reached the state with the last local goal is saved as new initial state. When one episode is terminated the next one will start there instead of jumping to the very start. The RL System is now able to learn a complete new wire without any knowledge (entries in the q-table) in about 10 minutes.

To regularly check the learning progress of the q-learning algorithm, it is embedded in further a function. This allows to execute the algorithm in two different configurations. In the first one, the exploration factor ε is set to zero, and the for-loop is limited to one episode. New actions will only be selected based on the q-values in the q-table. If the system reaches the global goal, it is assumed that the q-values are already approximated very well to the true q-values and all actions done in this episode can be written to a list and sent to the robot to play the game in the real world. If not, this checking-phase gets followed by a training-phase. Here ε is set to its actual value of 0.9 to allow the algorithm to explore its environment by doing random actions. After training a fixed number of episodes again the so made progress gets checked.

VII. Robot control

After the Reinforcement Learning Brain is done processing the simulated actions have to be transferred to the robot. For this part of the process we decided to use the ROS-industrial software. This is an open source environment for programming robot applications. It bundles a lot of different functions and processes necessary for complex robot systems. It contains support (drivers etc.) for many robots from different manufacturers, including the here used UR10e. In this way ROS enables a real time connection and allows to control the robot via remote control. Also integrated in ROS are some environment simulation tools like RViz and Gazebo as well as some motion planning tools like MoveIt. These tools enable the implementation of safety features such as safety plans and collision avoidance and can also calculate a motion plan out of the action list that can be sent to the robot.

The motion planning works with cartesian coordinates, whereas the q-learning is acting in the pixel space or rather in pixel coordinates. To map these two coordinate systems, we set up a transformation matrix ${}^{Ro}T_{Im}$. Facing only a two-dimensional problem the matrix has a size of 2x3 instead of 4x4 when working with homogeneous coordinates in the three-dimensional space. To calculate the transformations matrix several points in the work plane off the wire have been measured in image coordinates (${}^{Im}P_i = (column,\ row,\ 1)^T$) (Fig. 8 a) and robot coordinates (${}^{Ro}P_i = (y,\ z)^T$) (Fig. 8 b). With ${}^{Ro}A = ({}^{Ro}P_1,\ \ldots,\ {}^{Ro}P_n)$ and ${}^{Im}B = ({}^{Im}P_1,\ \ldots,\ {}^{Im}P_n)$ the transformation matrix ${}^{Ro}T_{Im}$ can be calculated by converting equation (1) in equation (2)

$$ {}^{Ro}A = {}^{Ro}T_{Im} \times {}^{Im}B \qquad (1) $$

$$ {}^{Ro}T_{Im} = {}^{Ro}A \times ({}^{Im}B)^{-1} \qquad (2) $$

Since ${}^{Im}B$ is not square the inverse matrix has to calculated using the pseudo inverse. For a resolution of 640x480 pixels for the initial image of the wire, the transformation matrix calculates the distance between to pixels as around 1.5mm. The whole transformation from pixel to cartesian coordinates is subject to a mean error of 1,3mm.

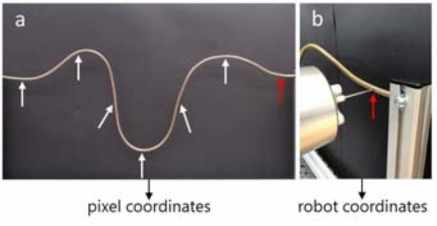


Fig. 8 Determination of the points for the transformation matrix

VIII. Application

In the following, motion planning with reinforcement learning is illustrated using one wire configuration as an example. Therefore, Fig. 9 shows a series of pictures of how the UR10e navigates along this wire configuration and successfully plays the wire loop game. In contrast to the shown wire configuration, the robot is also able to solve configurations with difficult sections, where the wire is bent backwards or runs vertically. Because of the chosen dictionary key in the Q-table (chapter VI) the system encounters no problems when solving these sections.

Most of the time the q-learning does not need any further training phase to play new wire configurations. This is made possible due to offline training sessions with artificially made wire images. To change the curves of the wire by hand and then take pictures of the real wire setup would have been very time consuming and inefficient. Therefore, we implemented the following automatic method to generate training data.

In a cartesian coordinate system with the same dimensions as an original image of the wire, we placed 14 fixed data points (Fig. 10 a). The first and last three points were constantly placed at the same position, whereas the eight ones in the middle are randomly placed and drawn from a uniform distribution. Through all points we then generated a cubic spline and transferred the created path to an image as shown in Fig. 10 b, which has the identical structure as the images at the end of the image processing (chapter IV). Based on thousands of these wire paths the system has been trained to move from left to right and from right to left to cover all possible states for backwards bended wire sections.

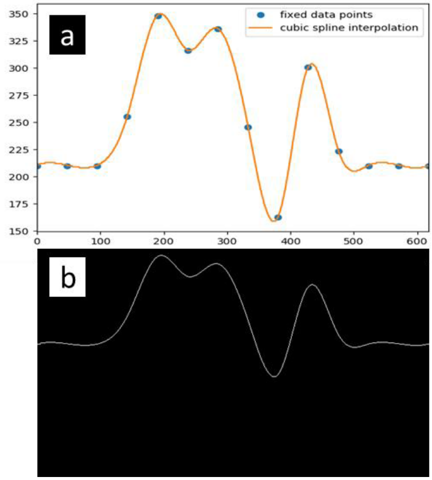


Fig. 10 Cubic spline interpolation and the resulting pixel path

IX. Evaluation

In principle this demonstrator shows a very robust and reliable behaviour. Nearly every tested wire configuration got immediately solved by the Q-learning system or only one training phase of 50.000 episodes (approx. 30s) had to be done. Nevertheless, the system contains some weaknesses which could not all be eliminated during the period of the developed project.

Image processing improvement: Although the black cardboard is installed in the background, the image processing was affected by changing lighting conditions. After the threshold function some areas of the black background also appeared as white pixels in the image in bright light, whereas under poor lightning some parts of the wire occur as black pixels. This Problem was effectively solved and satisfying results are achieved every time.

Local Movement problem: The robot navigating through narrow curves is another focus point. While the simulation has no difficulty in dealing with these tight curves, executing the calculated action list leads to collisions between loop and wire in the real setup. This can be attributed to two weaknesses in the virtual environment:

- It exists some variance in the proportions of the wire and the loop between the virtual environment (5x5 matrix) and the real setup

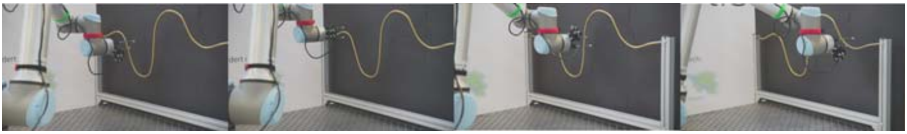


Fig. 9 Different states of the robot moving along the wire

- Second, because of the local constraints of the virtual environment the q-learning algorithm can‘t reach a local goal with a rotation and therefore never gets a direct positive reward for performing a rotation. As a consequence, the simulation only rotates its loop when it is necessary. This often leads to exceedingly small distances between loop and wire in the real setup. Together with the first issue the robot fails to play the wire loop game in narrow curves.

General movement problem: While playing the wire loop game the robot shows a very discontinuous movement. This scenario appears due to the used motion type. The robot executes the actions with a PTP (Point to Point) - movement. With this type of motion, the robot will generally slow down and stop when reaching a new point. PTP allows the blending of the motion path, but this could not totally eliminate the stuttering discontinuous movement because of the small distances between the single points and frequent changes of the movement direction. A possible solution for this problem could be spline interpolation through the waypoints.

X. Summary

In this paper an adaptive motion planning method using the reinforcement learning algorithm called Q-learning is proposed. Image processing, an enhanced learning development framework, ROS virtual environment and control system are integrated into an intelligent system. Based on the concept of Markov Decision Processes the system gains knowledge and experiences by performing actions in the virtual environment and receiving reward depending on the reached states. This knowledge consists of so-called Q-values which are mapped to state-action pairs and are stored in a look-up table. Due to the minimalistic structure of the virtual environment Q-learning is able to generalize the motion task. In the demonstrated use case of the wire loop game, the usability of the intelligent system is presented, which allows the robot to move along different wire configurations without any collisions.

In the future work the discussed weaknesses and limitations in the developed system like the discontinuous movement and the problems with narrow curves will be addressed. Additionally, based on the existing reinforcement learning frameworks, adaption to the third dimension and more complex real-time learning environments will be further developed. The ability to adapt to new products, processes and environments in production and assembly lines could reduce cost as well as programming and implementation time. Especially small and medium-sized enterprises, which produce customer specific products have to come up with high flexible assets. Tough and physically harmful tasks such as welding, gluing will be handed to the robot with reinforcement learning methods to adapt the non-linear path operations.

XI. Acknowledgment

The authors would like to thank the DFKI in person of Dr. Tim Schwartz and Andreas Luxenburger as well as Md Jonybul Islam, Matthias Jost and Christian Kapp for their contribution and generous support during the realization of this project in the context of the CoCoRo-seminar [18] and also Xiaomei Xu for her support during the writing process of this paper.

XII. References

[1] H. Dubbel, K.-H. Grote, and J. Feldhusen, *Taschenbuch für den Maschinenbau,* 23rd ed. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2011.

[2] M. Haun, *Handbuch Robotik: Programmieren und Einsatz intelligenter Roboter,* 2nd ed. Berlin, Heidelberg: Springer Vieweg, 2013.

[3] Patrick Dammann, *Einführung in das Reinforcement Learning.* [Online]. Available: https://hci.iwr.uni-heidelberg.de/system/files/private/downloads/541645681/dammann_reinfocement-learning-report.pdf (accessed: May 27 2020).

[4] B. Lacerda, D. Parker, and N. Hawes, “Optimal and Dynamic Planning for Markov Decision Processes with Co-Safe LTL Specifications,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems: (IROS 2014) ; Chicago, Illinois, USA, 14 - 18 September 2014*, Piscataway, NJ: IEEE, 2014, pp. 1511–1516. Accessed: May 27 2020. [Online]. Available: https://ieeexplore.ieee.org/document/6942756

[5] J. Burlet, O. Aycard, and T. Fraichard, “Robust Motion Planning using Markov Decision Processes and Quadtree Decomposition,” in *2004 IEEE International Conference on Robotics and Automation, 2004: Proceedings : ICRA '04 : 26 April-1 May 2004*, Piscataway, NJ: IEEE, 2004, pp. 2820–2825. Accessed: May 27 2020. [Online]. Available: https://ieeexplore.ieee.org/document/1307488

[6] M. Duguleana and G. Mogan, “Neural networks based reinforcement learning for mobile robots obstacle avoidance,” *Expert Systems With Applications*, no. 62, pp. 104–115, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417416303001

[7] R. Meyes, C. Scheiderer, and T. Meisen, “Continuous Motion Planning for Industrial Robots on Direct Senory Input,” in *CIRP Conference on Manufacturing Systems*, pp. 291–296. Accessed: May 27 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827118301707

[8] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, “Towards Vision-Based Deep Reinforcement Learning for Robotic Motion Control,” Nov. 2015. [Online]. Available: http://arxiv.org/pdf/1511.03791v2

[9] S. James and E. Johns, “3D Simulation for Robot Arm Control with Deep Q-Learning,” Sep. 2016. [Online]. Available: http://arxiv.org/pdf/1609.03759v2

[10] C.-K. Lin, “A reinforcement learning adaptive fuzzy controller for robots,” in *Fuzzy Sets and Systems*, pp. 339–352. Accessed: May 27 2020. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0165011402002993

[11] R. Meyes *et al., Motion Planning for Industrial Robots using Reinforcement Learning*. Aachen: Universitätsbibliothek der RWTH Aachen, 2017.

[12] S. J. Russell and P. Norvig, *Artificial intelligence: A modern approach*. Boston, Columbus, Indianapolis: Pearson, 2016.

[13] Logitech, *C920 HD Pro Webcam.* Accessed: May 27 2020. [Online]. Available: https://www.logitech.com/de-de/product/hd-pro-webcam-c920

[14] MoveIt, *company logo.* Accessed: May 27 2020. [Online]. Available: https://moveit.ros.org/blog/

[15] FAUDE Automatisierungstechnik GmbH, *Leichtbauroboter UR10.* Accessed: May 27 2020. [Online]. Available: https://faude.de/wp-content/uploads/2019/01/UR10_Leichtbauroboter_MRK-Systeme.png

[16] F. S. Melo, “Convergence of Q-learning: A simple proof,”

[17] L. Rejeb, Z. Guessoum, and R. M'Hallah, “The exploration-exploitation dilemma for adaptive agents,” *In Proceedings of the Fifth European Workshop on Adaptive Agents and Multi-Agent System*, 2005. [Online]. Available: https://www.researchgate.net/publication/250426352_The_exploration-exploitation_dilemma_for_adaptive_agents

[18] DFKI - Deutsches Forschungszentrum für Künstliche Intelligenz GmbH and ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gGmbH, *Cognitive Collaborative Robotics (CoCoRo): Seminar at Saarland University*. Course of study: Computer Science. [Online]. Available: https://cms.sic.saarland/cocooro/