Interreg

EUROPEAN UNION

Grande Région | Großregion

Robotix-Academy

Fonds européen de développement régional | Europäischer Fonds für regionale Entwicklung

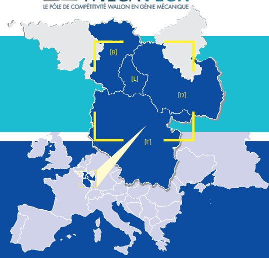
Axe prioritaire | Prioritätsachse 4
Compétitivité et attractivité
Wettbewerbsfähigkeit und Attraktivität

Robotix-Academy Conference for Industrial Robotics (RACIR) 2019

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, Robin Pellois, Matthias Vette-Steinkamp (Hrsg.)



www.robotix.academy

Berichte aus der Robotik

**Rainer Müller, Peter Plapper,
Olivier Brüls, Wolfgang Gerke, Gabriel Abba,
Robin Pellois, Matthias Vette-Steinkamp (Hrsg.)**

Robotix-Academy Conference for Industrial Robotics (RACIR) 2019

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 978-3-8440-7337-9
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)

Preface:

Robotix-Academy Conference for Industrial Robotics (RACIR) is held in University of Liège, Belgium, during June 05, 2019.

The venue for RACIR 2019 is the Campus in Liège. The University of Liège doesn't have a "long" history: many universities in Europe date back to the middle ages. Nevertheless, ULiège also has ties to that period. And for two centuries, its rich and abundant history is also that of the Europe after the Treaty of Vienna, then with the history of Belgium and its scientific, social and cultural, economic, and industrial destiny.

Today, ULiège relies on 200 years of creation and transmission of knowledge to be unfurled between international openness and regional engagement: University of Liège has an internationally relevant research university with 25,000 students and PhD students, more than 800 foreign students and exchange and cooperation agreements with 900 partner institutions over the world.

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 interface and 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

Machine-To-Machine (M2M) Communication of a Mobile Robotic Platform in Machine Tending Applications

Christopher Schneider and Michael Klos
Department of Product Management and Business Development
YASKAWA Europe GmbH
Allershausen, Bavaria, 85391, Germany
(christopher.schneider & michael.klos) @yaskawa.eu.com

Mohamad Bdiwi and Matthias Putz
Department of Robotics
Fraunhofer IWU
Machine Tools and Forming Technology
Chemnitz, Saxony, 09126, Germany
(mohamad.bdiwi & matthias.putz) @iwu.fraunhofer.de

Abstract **- In this paper, different communication possibilities of a mobile robotic platform in the manufacturing context are observed. Based on the use case of machine tending, the required information exchange is assessed regarding real-time priorities and data amount. Under the consideration of technical communication and current protocols, standards and architectures, a proposal for communication interfaces as well as a possible architecture is given.**

Index Terms – Machine to Machine Communication, Mobile Robots, Machine Tending, Internet of Things, Communication Architecture

I. Introduction

In 1953, the first automated guided vehicles (AGV's) were introduced by Barrett-Cravens in Northbrook, Illinois. Equipped with simple lane tracking technologies and bumpers as kind of sensors, these vehicles were bound to their correspondent tracks [26]. Due to the lack of integration, communication and standardization, the technology was unviable. Later, in the 1980's, the electronic and automation sector developed rapidly because of the third industrial revolution. By then, different interfaces and communication technologies were coming online. Today, the fourth industrial revolution brings in new possibilities of internal and external communication, especially through the developments in future projects such as Internet of Things (IoT) and Industrie 4.0 [21]. Equipped with current and emerging network technologies, companies are able to develop AGV's for manufacturing and integrate them in the existing infrastructures. As an evolution of AGV's, mobile robots are currently developed by different manufacturers [25]. For the successful development of mobile robots, several challenges need to be met first. Besides the definition of use cases, the integration of the mobile robot in various dynamic production environments, especially regarding external communication, must be analysed and assessed in detail

II. Current Communication Standards and Protocols

The protocol *Message Queue Telemetry Transport (MQTT)* attracts worldwide attention to become the standard communication protocol for Machine-To-Machine Communication [7] [12]. This protocol is usually built upon a network architecture, such as TCP/ IP, to provide reliable connection capabilities [22] [24]. *MQTT* works with the publish-and-subscribe principle and the information is provided in the XML format [3] [6]. First, a topic is defined toward a *MQTT* broker. Then, a device can publish data within this topic. Other topic-related devices get notified about the publication immediately [31]. If the subscribed device is in sleep mode, it gets notified when switching back to the active mode [7]. There are three important entities within a *MQTT* network: clients, brokers and topics, which can be seen in figure 1. Clients are all devices using a *MQTT* library to interconnect with a broker. In fact, clients can be divided into publishers and subscribers. While the publisher sends a specific message to the broker, the subscribers receive this message [1]. Brokers are responsible for authentication and authorization of the clients, receiving and filtering of messages and distributing the message to all subscribers [23]. The message broker operates as communicational middleware in order to orchestrate the single communication flows. Furthermore, there are *MQTT* brokers available, that support direct broker-to-broker communication, called broker-bridging [18]. The use of topics allows clustering, message distribution, filtering and routing [1].

Fig. 1 Publish-and-subscribe principle of the MQTT protocol (own graphic, according to [23]

MQTT provides three mechanisms of data delivery validating, a mechanism to track the current client connection and three security levels: none, user and password and the use of TLS/ SSL certificates. To reduce complexity, this protocol uses very basic headers. Via several ports or a web socket, communication is enabled. Usually, a two-level architecture with *MQTT* clients and an *MQTT* broker is used [7]. *MQTT-SN* (figure 2) is a modification of the original *MQTT* protocol, that is optimised for wireless communication and wireless sensor networks to face the challenges of low-bandwidth, high link failures, short message length, higher failure rates of wireless networks and limited processing as well as storage onboard-hardware of mobile devices [22]. This variant requires a three-level-architecture [3].

Like MQTT, *XMPP* can work with the publish-and-subscribe principle and is built upon a TCP/IP architecture. [7]. The overlay communication is executed over IP, supporting both standards (IPv4 and IPv6) [13]. Fundamentally, this protocol utilizes different XML technologies and can be extended by so called XEP's (protocol extensions) [15]. Due to the direct-connected client-server XML streams, near real-time communication can be attained [13]. In addition, this protocol does not require a protocol gateway or middleware for networking and simplifies device-connections [15]. The *XMPP* protocol technology is free, open, easily usable and standardized in RFC 3920 and RFC 3921. Due to its flexibility, various functions and applications can be deployed, such as network management, collaboration, file sharing or cloud computing [30]. Within the network, the single clients are identified under the use of a jabber ID (JID), which can act as publisher or subscriber [13]. Like MQTT, *XMPP* can be driven by the publish-and-subscribe principle, where several *XMPP* entities create topics (or also called nodes) and publish information respectively [2]. That information is shared with all the nodes, that have subscribed to the topic and are authorized to access this information [17] [27].

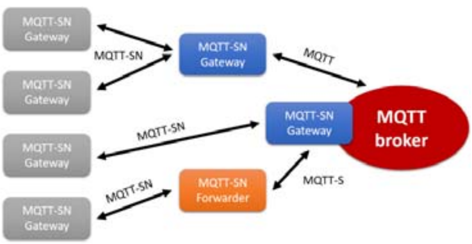


Fig. 2 Exemplary MQTT-SN architecture (own graphic, according to [22])

The functions subscribing, unsubscribing and publishing, can be defined by different messages (request, success or error). Furthermore, the creation, configuration, management and deletion of topics is possible. For the identification of the current connection status, connectivity information about every single entity can be provided (either connected or disconnected). Pubsub mechanisms are often used in sensor systems, where different physical devices produce huge amounts of information. Within such networks, sensors and actuators can act as publisher and subscriber simultaneously. Consequently, these entities can receive commands, read configuration files (subscriber) or share current sensor data with the network (publisher). In order to provide understandable information, the data is collected by aggregators. Aggregators, which are subscribers themselves, collect data from different topics and analyse them with the result of structured information, that can be used more efficiently. In fact, this information bundles are published in separate topics [13].

Based on several web technologies, *REST* utilizes HTTP as transfer protocol, Universal Resource Identifier (URI) for identification, XML for representation and Multipurpose Internet Mail Extensions (MIMO) as content identifier. In the *REST* approach, an XML file describes a specific content, which is placed on a correspondent web page linked with a website [14]. In the field of M2M communication, *RESTful* web services are often deployed due to their stateless service. Consequently, unreliable connections can be stabilised. Furthermore, connected devices do not require extra memory in order to manage connection states. Via *RESTful* API's (Application Programming Interfaces), the single applications can be adjusted individually. But when it comes to networks with extended requirements, where high data rates combined with high battery and CPU power consumption occur, *REST* is not suitable due to the big HTTP headers and huge XML and JSON data packets. Consequently, a more lightweight protocol is required: *CoAP* [7].

As already mentioned, IoT and M2M applications are defined by resource constraints and a high number of low-power devices. Therefore, a lightweight protocol is required as an alternative to HTTP. *CoAP* is based on the REST architecture and runs over UDP [7]. This protocol transfers information directly between client and servers, has a low overhead and easily translates to HTTP [3] [14]. The information is provided in the formats XML or JSON [6]. With this protocol, two models can be utilized: request-response (like in HTTP) or publish-and subscribe. While MQTT works with topics (nodes), *CoAP* utilizes URI as nodes. In comparison to MQTT, *CoAP* uses different reliability mechanisms. In contrast to MQTT (which utilizes the reliable TCP), *CoAP* is built upon UDP. However, specific mechanisms are deployed at *CoAP* in order to guarantee a reliable data exchange. Therefore, *CoAP* messages are differentiated in “confirmable messages” and “unconfirmable messages”, which either require an acknowledgement or not. Furthermore, there is only one QoS level at *CoAP*. [23].

OPC Unified Architecture is a service-oriented architecture (SOA), where a service provider receive requests, calculates them and send the solution back (response). Due to its standardized and generic services, *OPC UA* is compatible, interoperable and platform-independent [8]. It is built in four layers: the abstract UA model specification at the bottom, the service binding on the second level and the extensions and modifications on top of the pyramid. This architecture provides interoperability on a semantic level by offering the exchange of several complex information models. Figure 3 shows the information model, where controls on the field level can relate to overall information systems. Therefore, functionalities for transport, meta model and services are required. Transport allows the data exchange between various *OPC-UA* applications under the use of different, application-specific protocols. UA TCP (which is based on TCP/IP) guarantees speed and throughput, while HTTP and SoAP is firewall-friendly [10]. In the meta-model, different rules and basic elements for the information model are defined. Different services realise the interfaces between servers as information providers and clients as information users [16]. The information model itself is built layer-based, while every higher type uses basic rules. That means, that clients with restricted rule knowledge can edit complex information models, even if these clients are not aware of the relations within the model [19]. Furthermore, *OPC UA* offers an integrated addressing space, where production data, alerts, events, historical data and tasks can be included [16]. Consequently, only one interface is required for navigating the different addressing spaces [19]. According to different sources, *OPC UA* is going to be IoT standard [5] [9] [11] [20].

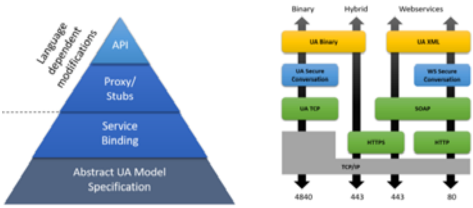


Fig. 3 OPC UA Architectures (own graphic, according to [19])

III. IoT and M2M Architectures

To accomplish the vision of IoT, a vertical approach in systems architecture, so called “silos”, is currently emerging. Each application has its own infrastructure, which leads to redundancy and high costs [3]. In the future, this vertical architecture is going to be replaced by a horizontal one with an overall operational platform for task managing [6]. According to [3], such a network architecture can be distinguished in three layers and phases: the collection phase, the transmission phase and the processing, managing and utilization phase [3]. As already mentioned before, the physical nodes, that means physical devices (RFID, sensors), communicate via short-range communication. For that reason, various protocols can be used, to guarantee efficient networking. Protocols can be distinguished into low power networking protocols (ZigBee, ZWave, Bluetooth), traditional networking protocols (Ethernet, WIFI) and IoT networking protocols (CoAP, MQTT, XMPP). However, the heterogeneity of the available protocols and the need for protocol-independent IoT architectures require solutions for interoperability. Due to the computational resources of gateway nodes, gateways are used to connect sink nodes (also called base or destination nodes) with IoT services, which is called Gateway as Service (SGS) [6]. In fact, the gateway manages the data transmissions between the physical devices and the cloud, which can be considered as Semantic Service Oriented Architecture (SSOA). For this architecture, the base nodes are interconnected either in a hierarchy or a mesh network and connect to the gateway under the use of the previous mentioned protocols (CoAP, XMPP, MQTT). In fact, the data, transmitted to the gateway, is not annotated. In this architecture, the physical sensor nodes relate to simple clients, that support a different protocol each (CoAP, MQTT, XMPP). Via different formats, such as JSON and/ or XML, the client information can be transmitted to the multiprotocol proxy. Therefore, every protocol-specific information stream has its own channel. Due to the different architectures of CoAP, MQTT and XMPP, the multiprotocol proxy is required for an appropriate translation. While MQTT uses the publish/ subscribe architecture, CoAP utilizes either the request/ response or resource/ observer architecture. Based on the publish/subscribe principle (also called pubsub), XMPP understands resources as nodes instead of topics. Due to these specifications, different formats (XML and/ or JSON) are required. As it can be seen in figure 4, the message store and the topic router are exchanging data with the multiprotocol proxy, which guarantees the translation process.

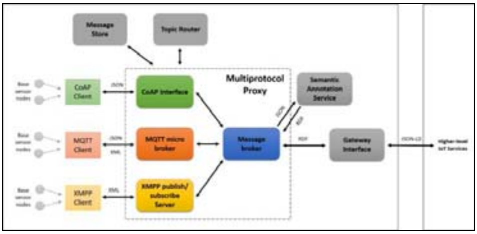


Fig. 4 Model of a Gateway as Service (SGS) architecture including a multiprotocol proxy (own graphic, according to [6])

The topic router offers the creation and management of different topics and the assignment of specific sensor states to these topics. In fact, the alignment of CoAP information with these topics is possible as well. In the message store, the messages from the different clients can be stored and forwarded to the correspondent topics. However, after the multi-channel messages has been translated within the multiprotocol proxy, the message broker transmits them to the semantic annotation service in the JSON format. After processing, the message broker receives the annotated information in the RDF format. This format is required for data transmission to the gateway interface. Then, the gateway transforms this RDF format in a specific-annotated JSON format, which allows the support of RESTful protocols. That is required in order to connect the considered system with cloud services and other SGS (Gateway as Services) systems [6].

M2M communication utilizes general information and communication technologies (ICT) as well as Big Data. Regarding Big Data in M2M, five main requirements are demanded: real-time-processing, scalability, ubiquity, reliability and heterogeneity [4] [7]. According to [7], three different M2M architectures can be applied: the three-level architecture with non-IP end devices, the two-level architecture with IP-enabled end devices and the two-level architecture with non-IP end devices. In the first model (three-level architecture with non-IP end devices), there is no IP assigned to the single devices. Usually, this architecture is deployed when using low-cost end devices without any intelligence or integrated network access. In this case, capillary networks with several gateways are used. In the first layer, either a point-to-point connection via a gateway (i.e. IEEE 802.15.4, M-Bus) or a mesh/ routed connection via a neighbor relay or gateway (i.e. ZigBee, Z-Wave) is established. Based on the first layer, a gateway in the second layer establish an IP-enabled connection using common technologies such as WIFI, Ethernet and cellular. Common protocols are UDP, TCP and HTTP, especially for M2M data exchange the protocols MQTT and CoAP are utilized. Finally, the third layer connects to service providers to manage the single devices in terms of exchanging data and device interaction. The second model is the two-level architecture with IP-enabled end devices. These devices, based in the first physical layer, can connect autonomously to the second layer by using WIFI, Ethernet or cellular. Consequently, a gateway is not required. Due to the ability of autonomous connection, the device itself must have an integrated memory and intelligence. These hardware requirements are a prerequisite in order to apply the appropriate protocol, such as HTTP or MQTT. The last model is the two-level architecture with non-IP end devices. Under the use of an IP-based backhaul, simplification and broader network covering can be achieved [7].

IV. Use Case Description

As main use case, the automated workpiece exchange (Machine Tending) has been predefined for this study. Therefore, a mobile robot platform with an industrial robot manipulator attached to it, is assumed. The process can be subdivided into the following procedures:

- Machine Door Status Check & Door Opening
- Workpiece Status and Weight Check
- Workpiece Space Status Check
- Fixture Decomposition
- Pick workpiece within the machine
- Next Workpiece Identification
- Workpiece gripping and handling
- Place workpiece into the machine
- Applying fixture specifications
- Closing the machine door

The following figure illustrates the use case.

Fig. 5 Use Case of Machine Tending with a mobile robotic platform (YASKAWA graphic)

V. Exchange Information Analysis

In this chapter, information that must be exchanged in mobile-robot systems is analyzed. Firstly, information in general are described, according to different sources. The basis for this chapter forms [29]. In VDI/VDMA 5100 and VDI 2510, various information for material flow systems of automated guided vehicles (AGV) has been collected [27] [28]. Based on the vision of the Internet of Things, further information is reasoned: registration/ deregistration of software agents at system components, service requests by software agents of the transport units, order negotiations between module-agents, notifications of the module-agents about offer and establishment/ cancellation at software services [29]. Wirth combines the information, proposed by VDI and VDMA, with the information requirements of the Internet of Things paradigms to create information classes. Hence, the information can be clustered in:

- visualization data
- topology/ layout
- organization/ control
- order negotiations
- transport job data
- driving job data
- reservations
- load cycle control
- identification data, sensor data
- collision avoidance
- status/ error notifications
- and switch jobs.

Corresponding to these classes, several sub data can be reasoned. In addition to this information classification, Wirth clustered this information according to their real-time requirements. Therefore, the criteria time targets, maximal data amount, average demand frequency and number of endpoints were used to evaluate the real-time requirements. In order to evaluate the technical performance level, the criteria pair time targets and maximal data amount are utilized. Figure 6 shows the relation of the single data classes to these two criteria [29]. Based on this information analysis, further information must be assessed, that are specifically designed for the field of mobile robotics. By process modeling of the use case with EPC's (event-driven process chains), the exchanging entities and the required information has been analyzed. Fundamentally, the information classes are clustered into "identifiers", "states", "interaction", "description data", "robot arm data", "technical motion execution details", "mobile platform data" and "complex data".

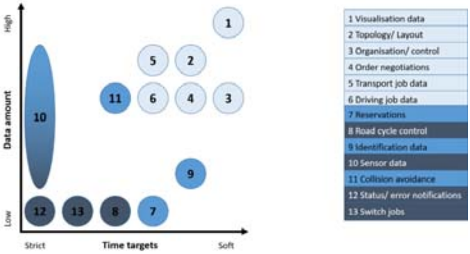


Fig. 6 Data amount and real-time requirements of different classes in general (own graphic, according to [29])

Due to the numeric character of identifiers, the data amount is low. Furthermore, the task activations linked to these numbers do not have to be executed immediately, therefore real-time submission is not required. Secondly, status data, job states and the battery status are needed. In a pubsub architecture, these states are used to write the values into the topic (blackboard), only when the values are changing. Hence, a lean communication network can be created. The Boolean character of states allow low data amounts. Furthermore, states do not require mandatory real-time features. Interactions are much more complex. This information consists of requests or commands, which enlarges their data amount. While ERP requests do not have to be transmitted immediately, command data require high real-time priorities. The safety character of these interactions forces the information exchange to be executed in real-time. Description data is usually very lightweight, because it consists of simple information, such as limits or dimensions. Furthermore, the transmission priority is usually low except of timers due to the required timer accuracy. All mathematical and informatic data, that is used for the robot manipulator, are naturally high. This results from the complex frames, vectors, plains and Jacobi matrices. However, the real-time priority of robot joint data and robot frame is low. In the technical motion specification execution details, specific data according to fixtures, tools, gripping, handling and rework for the robotic manipulator are given. Due to the vast information required, the data amount is high. In contrast, real-time requirements are low, because the motion itself can be executed with a small delay without endangering safety or slowing the processes down tremendously. For the mobile platform, several information classes are required. Position data are used for the mobile platform as well as for the robotic manipulator, which are quite simple, due to their constellation of x, y and z values. Furthermore, position data can be transmitted with a small delay. Map data represent all factory elements as well as tracks, safety zones and other 2D or 3D elements. Hence, the data load is huge. Usually, factory layouts do not change very often, which is why map data is mostly static. The map itself is downloaded by the mobile robot regularly, which does not require real-time features. However, live mapping is possible, which requires such features. The live changes of elements within the map implies flexible facilities, like other mobile robots. In order to avoid collisions, routing information can be provided. Therefore, the data amount as well as the real-time priorities are high. However, routes are typically calculated in advance, so that the fleet manager can calculate collision-free tracks for every mobile robot. In this case, real-time transmission is necessary. But when irregularities in the driving process of a mobile robot occur, i.e. by unpredictable obstacles, this information must be reported to the fleet manager. Then, an alternative route is calculated, which impacts the other routes as well. Consequently, real-time features for these cases are required. Obstacles, as well as the positions of other robots, can be perceived by the sensors, attached to the mobile robot. Under the use of the sensor data, relational position data can be reasoned. In contrast to normal position data, the positions are indicated as incremental values and relations. Hence, the data are much more complex. In fact, this information must be transmitted in real-time in order to enable the robot to converge to other objects precisely. Before a trip can start, the mobile platform needs specific information about the control of their motors (driving parameters), such as acceleration, velocity, reaction time and respective braking. From the diversity of these information streams the complexity can be sensed. These parameters are usually requested, buffered and applied before the trip starts. Therefore, real-time transmission is not important. Changes in the drive process, such as speed limits or safety specifications, can be delivered in real-time at the respective spots. While speed limits imply simple values (low data amount), safety specifications, such as sensor calibration characteristics, are much more complex. In fact, sensor data itself are complex as well. By combining several sensor states, comprehensive conclusions can be made. In order to react to environment conditions, sensor data must be transmitted in real-time. Another class of complex information is visual data. When scanning 3D objects for quality reasons, the scanned data must be transferred into a 3D object for further analysis of quality attribute. Hence, comparing calculations of the actual and the optimal 3D model are required, which increases the data amount. Regarding data amount and real-time priorities, the previously described information classes were put into the following diagram, which is analogue to figure 7.

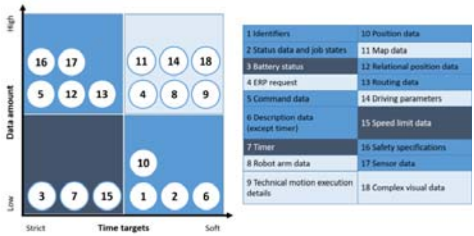


Fig. 7 Data amount and real-time requirements of the information, that has been identified in the use case analysis (own graphic)

According to Wirth, eight requirements for communication systems for mobile robots in general need to be met, which are appropriate for mobile robots itself as well. Summarized, these demands are: allocation of information as needed, wireless communication technology, communication in real-time under consideration of latencies, interoperability including open standards, coexistence with other communication systems, self-regulation regarding signal strength and package failure rate, network security as well as energy supply via a battery. In addition to these basic requirements, further ones can be reasoned from the dissertation of Wirth. For mobile robots, the following are relevant: discovery of communication partners, data consistency and data redundancy [29]. Besides general requirements, the communication interfaces must be considered as well. Based on the blackboard system, that has been proposed by Wirth, the author proposes an exemplary, more use case-tailored system in figure 8. Hence, further requirements can be deduced. Firstly, the mobile robot must be enabled to communicate on the field level with other machines, facilities and devices directly (M2M communication via OPC UA).

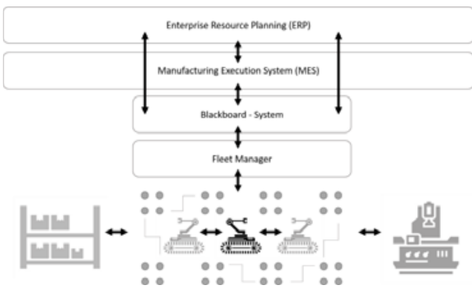


Fig. 8 Exemplary communication architecture (own graphic, according to [29])

Secondly, it is necessary to work with a blackboard information system, which can be achieved by different communication principles, such as publish-and-subscribe. Due to the diversity of communication protocols and the lack of a protocol standard, which is predominantly used, protocol interoperability is important. For this purpose, the principle of a multiprotocol proxy has been explained previously, which can handle different protocols. In fact, the different technologies, like MQTT, XMPP, CoAP and OPC UA must be actively supported as a prerequisite. Note, that if the expert's predictions come true and OPC UA becomes the communication standard in the future, active support of this protocol and architecture is going to be important. Lastly, the mobile robot must interact with the ERP and MES system. Therefore, it is mandatory required, that requests from these information systems can be understood and that their own requests can be executed in an understandable way. This can be achieved either via direct communication or via the use of a fleet manager. This middleware system is suitable, due to the fact, that it can collect requests from both sides, translate them in a way, that is understandable for the corresponding system and allocates them afterwards. For the middleware itself, two different possibilities are offered. Either this middleware is integrated in the mobile platform or the middleware is externalized. In the first case, the decentralization of information processing is advantageous. On the other side, the higher complexity and the extra costs of the platform are a disadvantage. By externalizing the middleware, computing resources can be outsourced, which means that this external system sends the required commands directly to the robot. On one hand, costs for the platform can be spared. On the other, the system reliability of a centralized system is critical. Besides external interfaces, the internal communication is required as well. Firstly, the mobile platform must interact directly with the robotic manipulator. Therefore, the robot memory must communicate with the manipulator and the mobile platform likewise, via IO interfaces. In fact, a safety circuit (Servo ON/OFF) between platform and robot is usually deployed regarding safety regulations. That means, that the robotic manipulator can only move, when the platform stands still. Usually, the robot is controlled by its control system or a PLC. In addition, the gripper system needs an interface via a Bus interface. When it comes to the usage of a collaborative robot, further sensor technology is required in order to guarantee safe human-robot-interaction. For such considerations, the author refers to different regulations, such as ISO TS 15066. Furthermore, several sensors, that are attached to the robot and to the mobile platform, must communicate with each other in order to build a comprehensive device.

VI. Results

In this paper, different communication possibilities of a mobile robotic platform in the manufacturing environment were explored and developed. By observing current and emerging communication protocols, standards and architectures, it became clear, that there is no dominant overall communication standard right now. On the field level, OPC UA as communication standard is required. In order to support different communication protocols and to guarantee interoperability of the system, protocol translation mechanisms were presented. As an example, a multiprotocol-proxy was explained in detail. Furthermore, different M2M and IoT architectures were described, which built the basis for a proposal of a possible architecture. Based on the use case, the required information was gathered and clustered in different classes under consideration of data amount and real-time priorities. Due to the degree of novelty of mobile robots and the subsequent lack of comprehensive information and literature, this paper only lays down a foundation for future research. Therefore, further investigations, experiments and practical tests are required in order to create a stable, comprehensive and functional communication architecture.

References

[1] Agarwal, N.; Paul, S.; Gujar, P.; Gite, V. (2016): Internet of Things (IoT) based switchbox using MQTT Protocol. In IJRET: International Journal of Research in Engineering and Technology, pp. 405–411.

[2] Bauschert, T. (2013): Advances in Communication Networking. 1st edition. Heidelberg New York Dordrecht London: Springer Verlag.

[3] Borgia, Eleonora (2014): The Internet of Things vision: Key features, applications and open issues. In Journal of Computer Communications.

[4] Chilamkurti, N.; Zeadally, S.; Chaouchi, H. (2013): Next-Generation Wireless Technologies: 4G and Beyond: Springer Science & Business Media.

[5] Delsing, J. (2017): IoT Automation: Arrowhead Framework: CRC Press.

[6] Desai, P.; Sheth, A.; Anantharam, P. (2015): Semantic Gateway as a Service architecture for IoT Interoperability. In Proceedings of the 2015 IEEE International Conference on Mobile Services, pp. 313–319.

[7] Dohler, M.; Antón-Haro, C. (2015): Machine-to-machine (M2M) Communications: Architecture, Performance and Applications. 1st edition. Cambridge: Elsevier Ltd.

[8] Eruvankai, S.; Muthukrishnan, M.; Mysore, A. K. (2017): Accelerating IIOT Adoption with OPC UA. In Internetworking Indonesia Journal 9 (1). Available online at http://www.internetworkingindonesia.org/Issues/Vol9-No1-2017/iij_vol9_no1_2017_eruvankai.pdf

[9] Glück, M. (2016): FAQ Industrie 4.0. 1st ed. Düsseldorf: Symposion Publishing GmbH.

[10] Grüner, S.; Pfrommer, J.; Palm, F. (2015): A RESTful Extension of OPC UA. In IEEE World Conference onFactory Communication

Systems (WFCS). Available online at http://ieeexplore.ieee.org/document/7160557/

[11] Huang, D.; Jo, H.; Lee, H.; Kang, H.; Bevilacqua (Eds.) (2009): Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence. 5th International Conference on Intelligent Computing. Ulsan, South Corea, 16.19. September 2009: Springer. Available online at https://books.google.de/books?id=2cZqCQAAQBAJ&dq=opc+ua+dominant+protocol&source=gbs_navlinks_s

[12] Hunkeler, U.; Truong, H. L.; Stanford-Clark, A. (2008): MQTT-S – A publish /subscribe protocol for Wireless Sensor Networks. In 3rd International Conference on Communication Systems Software and Middleware and Workshops, pp. 791–798.

[13] Iivari, A.; VäisÄnen, T.; Alaya, M. B; Riipinen, T.; Monteil, T. (2014): Harnessing XMPP for Machine-to-Machine Communications & Pervasive Applications. In Journal of Communications Software and Systems 10 (3).

[14] ITU-T Focus Group on M2M Service Layer (2014): M2M service layer: APIs and protocols overview. In Telecommunication Standardization sector of ITU.

[15] Klauck, R.; Kirsche, M. (2012): Chatty Things - Making the Internet of Things Readily Usable for the Masses with XMPP. In 8th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing, Collaborate.com.

[16] Leitner, S.; Mahnke, W. (2006): OPC UA - Service-oriented Architecture for Industrial Applications. In Softwaretechnik-Trends. Available online at https://www.semanticscholar.org/paper/OPC-UA-Service-oriented-Architecture-for-Industria-Leitner-Mahnke/0ccb58f9a3a9df31ec16c9993285e3e7d7d46aff

[17] Mandal, A.; Baldin, I.; Xin, Y.; Ruth, P.; Heerman, C. (2014): Enabling Persistent Queries for Cross-Aggregate Performance Monitoring. In IEEE Communications Magazine

[18] Novinskiy, A. (2016): A View of WSN-facilitating Application's Design and a Cloud Infrastructure in Academic Environment and Research. In Proceedings of the Eleventh International Network Conference (INC).

[19] OPC Foundation (2013): OPC Unified Architecture - Wegbereiter der Industrie 4.0. Available online at https://www.iosb.fraunhofer.de/servlet/is/21752/OPC-UA-Wegbereiter-der-I40.pdf?command=downloadContent&filename=OPC-UA-Wegbereiter-der-I40.pdf

[20] Palm, F.; Grüner, S.; Pfrommer, J.; Graube, M.; Urbas, L. (2015): Open Source as Enabler for OPC UA in Industrial Automation. In IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA). Available online at http://ieeexplore.ieee.org/document/7301562/

[21] Reinheimer, S. (2017): Industrie 4.0: Herausforderungen, Konzepte und Praxisbeispiele: Springer Verlag.

[22] Stanford-Clark, A.; Truong, H. (2013): MQTT For Sensor Networks (MQTT-SN): Protocol Specification Version 1.2. In International Business Machines Corporation (IBM).

[23] Tang, K.; Wang, Y.; Liu, H.; Sheng, Y.; Wang, X.; Wei, Z. (2013): Design and Implementation of Push Notification System Based on the MQTT Protocol. In Atlantis Press

[24] Thangavel, D.; Ma, X.; Valera, A.; Tan, H.; Tan, C. (2014): Performance Evaluation of MQTT and CoAP via a Common Middleware. In 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)

[25] Tzafestas, S. (2013): Introduction to Mobile Robot Control: Elsevier.

[26] Ullrich, G. (2013): Fahrerlose Transportsysteme: Eine Fibel - mit Praxisanwendungen - zur Technik - für die Planung. 2nd edition: Springer Verlag.

[27] Richtlinie, 10/2005: Fahrerlose Transportsysteme (FTS). Available online at https://www.vdi.de/nc/richtlinie/vdi_2510-fahrerlose_transportsysteme_fts/

[28] Entwurf/ Draft Richtlinie 5100, 03/2008: Systemarchitektur für die Intralogistik (SAIL): Beispiele zur Modellierung. Available online at https://www.vdi.de/nc/richtlinie/entwurf_vdivdma_5100_blatt_2-systemarchitektur_fuer_die_intralogistik_sail_beispiele_zur_modellierung_/

[29] Wirth, P. (2013): Kommunikationskonzept für selbststeuernde Fahrzeugkollektive in der Intralogistik. Technische Universität München (TUM), München. Fakultät für Maschinenwesen, Lehrstuhl für Materialfluss und Logistik. Available online at http://www.fml.mw.tum.de/fml/images/Publikationen/Tenerowicz-Wirth.pdf

[30] XMPP: An Overview of XMPP. XMPP. Available online at https://xmpp.org/about/technology-overview.html

[31] Xu, Y.; Mahendran, V.; Sridhar, R. (2016): Towards SDN-based Fog Computing: MQTT Broker Virtualization for Effective and Reliable Delivery. In COMSNETS 2016 - Workshop on Wild and Crazy Ideas on the interplay between IoT and Big Data

Robotic assistants in factory routines – the ethical implications

Sophie Klecker, Bassem Hichri and Peter Plapper

Faculty of Science, Technology and Communication
University of Luxembourg
Luxembourg, Luxembourg

sophie.klecker@uni.lu

***Abstract* - This paper is concerned with the problems which arise when humans are working alongside robotic assistants. The main question which appears is how to define the difference between humans and robots in terms of characteristics, similarities or differences and how to consequently treat humans and robots in the factory routine.**
Based on a literature analysis, a common ground for the treatment of human and robotic workforce in the manufacturing industry is established. Subsequently, a framework for their cooperation is deduced and an implementation of the solution suggested.

Index Terms – machine ethics; robot ethics; HRI; industrial robotics.

I. Introduction

Technological progress related to robotics and artificial intelligence is a fact and the impact on the manufacturing industry and its labour cannot be denied. Contrary to the panic fomented by mass media evoking the new and imminent threat of science fiction-like scenarios of humanoid robots taking over the lead and oppressing human workers, automation is not a new trend. The replacement of human workforce is a trend which has been ongoing over the past century [1]. Who would nowadays want to resume jobs like boiler man to fire up a ship's engine, gandy dancer or bindery worker? Despite of technology's contribution to the improvement of jobs in terms of ergonomics and safety, there exists a pluralism in society. While some argue for the unrestricted exploitation of technological opportunities, others are in favour of banning all automation in industry and maintaining a status-quo of the current situation. As neither of these extreme scenarios is likely to happen, experts and society alike should be prepared for the introduction of robots in manufacturing engineering and think beforehand about the implications. Robot ethics is concerned with the global environmental, ethical as well as societal impacts of the further emergence of the robotics industry [2-4]. The situation envisioned here is a partial factory automation as is either current state-of-the-art or feasible in a short term future. Human Robot Interaction, HRI, is considered.

This work is concerned with the problems which arise when humans are working alongside robotic assistants. The main question which appears is how to define the difference between humans and robots in terms of characteristics, similarities or differences and how to consequently treat humans and robots in the factory routine.

The goal of the presented analysis is to find a consensus among all stakeholders with their diverging opinions and to agree on a global framework for the temporal and local coexistence of human and robotic workforce in an industrial environment. The collaboration of humans and robots should indeed take place in a safe structure with values and rules everyone involved can identify with. The contribution of this work is the suggestion of a well-grounded framework for the safe, harmonious and efficient cooperation of humans and robots in a factory.

The rest of this paper is structured as follows: this introduction localizing the topic, specifying the ethical question addressed as well as the contribution of the work is followed by the main part. This part is subdivided into two major sections. The first one is of a theoretical nature. Based on a literature study as well as on discussions with stakeholders, a common ground for the treatment of human and robotic workforce in the manufacturing industry is established. Subsequently, a framework for their cooperation deduced. The following part describes the implementation of the solution suggested in the previous part, i.e. the practical setup of the suggested framework. The paper ends with a final chapter which discusses and summarizes the results.

II. Analysis

This chapter of the paper applies the previously described methodology in order to address the ethical problem defined in the introduction. First, the theoretical discussion of the question based on a literature study is presented. Then the gap between theory and practice is bridged by suggesting an implementation of the developed solution.

A. Ethical Theories

The following analysis discusses the differences between humans and robots in the light of the related question whether robotic assistants should be granted rights and responsibility to the same extent as their human co-workers. Further, state-of-the-art in the branches of ethics concerned with the actors in a partially automated industry is investigated. In a final part, a framework for the efficient and safe cooperation of human and robotic workers is derived.

In a first step, it is investigated how humans have been conceptually defined in order to establish similarities as well as differences with robots. This is achieved through a twofold analysis.

- First, the distinguishing characteristics that have been attributed to humans by the most influencing authors from antiquity to this day have been investigated. In this context, the conceptual and factual definitions of beings that should be granted dignity and responsibility are analysed in the framework of different ethical theories, i.a. Utilitarianism or Consequentialism, Libertarianism and Personalism or Existentialism.
- Second, a brief presentation of the discussed concepts in a religious framework completes the mainly secular discussion.

The first author to be mentioned in this context is the ancient Greek philosopher Aristotle. Among the unique natural properties that define humans, he focussed on rationality, i.e. the fact that humans are endowed with the capacity to reason. The ability to think and take decisions based on logic distinguishes humans from other creatures and objects. In the 15th century, Pico della Mirandola [5], published his thoughts on the distinguishing nature of humans. The Italian Renaissance philosopher qualified humans as free, vulnerable and imperfect. He insisted on the concept of self-development, the fact that humans are free from deterministic laws. According to Pico della Mirandola, humans are not restricted by or bound to laws from nature or the outside, instead they are free to choose their acts and behaviours.

The English philosopher of the 17th century John Locke is seen as one of the founders of Libertarianism. As the name suggests, liberty and freedom are at the core of the theory. As far as the characteristics of the human are concerned, it is believed to be free to develop itself autonomously starting from scratch through personal experiences [6]. Jean-Jacques Rousseau, inspired by Voltaire, elaborated the concept of freedom [7]. With his ideas of humans having the freedom to choose and free will, he influenced the Enlightenment in Europe as well as the French Revolution in the 18th century. In Utilitarianism, influenced by the Scottish philosopher David Hume, the English jurist and philosopher Jeremy Bentham as well as his student John Stuart Mill, the focus is put on sensation. The ability to feel the difference between pleasure and pain characterizes a being worthy of dignity. This qualification is in line with the theory's axiom, the greatest happiness principle. According to Utilitarianism, the aim of right behaviour should be the greatest possible happiness of the largest possible amount of beings [8]. In the same time period, Immanuel Kant [9] took up the capacities-based definition from Aristotle. The focus lay on the potential of the human as a rational agency to set ends through logic and reason. The German philosopher however extended the definition with the concept of morality. Humans are not only rational but also moral agencies, i.e. they know the distinction between good and bad and can adapt their acts accordingly. In the 20th century, the ideas of Personalism and Existentialism became popular, i.a. thanks to their supporters: the French philosopher Emmanuel Mounier, the Swiss Denis de Rougemont and the Russian philosopher Nikolai Berdyaev. This theory focuses on the uniqueness, self-consciousness as well as freedom and free will of persons [10]. Based on the philosophy of Aristotle, in the 20th and 21st centuries, the German Roman Catholic philosopher Robert Spaemann [11] deepened the discussion on morality. He sees the human not only as a moral individual but also as a moral member of a community, i.e. a participant in a group of mutual recognition and respect. In this sense, a human is not only defined through his individual thoughts and acts but also through his behaviour towards his peers. More recently in the 21st century, both concepts, free will and morality are confirmed as human qualities [12]. Further, Pico della Mirandola's ideas were taken up as the concept of self-transcendence, i.e. the human's ability to be or become different from its naturally given form [13].

Concepts similar to the ones identified in secular literature can be retrieved from the different religious doctrines. In Christianity some form of dignity is intrinsic. It can be characterized as the human's soul which according to their belief appears during natural conception [14]. Next to the inherent form of dignity accorded to humans, the Islamic belief also includes the concept of rationality. Their ability to think and decide with reason is one of the foundations of God's trust in humans [15]. In Buddhist as well as in Hinduist traditions, a strong focus is put on the existential freedom and autonomy of individuals [16] which culminates in the possible salvation from the cycle of eternal rebirth for living beings. The Jewish doctrine teaches that humans are different from the rest of creation because of their free will, their freedom to choose. Their morality, i.e. them knowing the difference between good and evil, leads to the responsibility that is entrusted to each and every individual [17].

Summarizing, it can be noted that, despite the prevalent ethical pluralism on the definition of humans or creatures worthy of being granted dignity, some elements are recurrent and allow the establishment of a consensus on a difference between human and robotic workers in semi-automated plants.

The recurring ideas in most secular and religious theories are rationality, morality and freedom. These are the three qualifications that distinguish humans, i.e. beings worthy of dignity and of irreplaceable value from objects, i.e. beings that can be exchanged and that a quantifiable price can be attached to.

Robots can be qualified as rational agencies. Indeed, state-of-the-art artificial intelligence algorithms are able to take logical, rational decisions based on general rules and inputs from sensors. The behaviour of robots: evaluating current situations based on environmental information acquired through sensors and taking decisions on the next acts based on learned rules, emotions and experience is similar to the behaviour of humans and can be qualified as rational behaviour [18-20].

Though robots are able to follow moral guidelines once they have been implemented in their program, they are not able to derive ethical principles by themselves. In this sense, robots are no moral agencies. The ethical principles and resulting moral or amoral behaviour of robots is not their responsibility, but rather the responsibility of the programmers, moral agencies themselves, who developed and implemented the ethical guidelines [21].

The most striking difference between human and robotic workers in this context is related to their freedom and free will. Although they have the capacity to evaluate situations, take decisions autonomously, learn, gain experience and adapt their behaviour accordingly, robots are not free. Robots are designed, developed, produced and programmed by humans. In this sense, robots do not have any inherent free will, but are dependent on an input from outside, they only act like they have been programmed to by an outside factor, i.e. they are trapped in heteronomy. As also freedom is a conditio sine qua non for morality, robots are no moral agents.

Two conclusions can be drawn from the above analysis. First, human and robotic workers have to be treated differently according to their respective characteristics and qualifications. The framework for their collaboration therefore has to address these differences. Second, an additional actor needs to be taken into account next to human workers and their robotic assistants in partially automated factories. This third actor is the robot developer and programmer. Indeed, as pointed out before, the human programmer is responsible for the ethical principles the robot is following as well as for the resulting behaviour.

Two different branches of ethics are concerned with this third actor, the human robot developer.

B. Robot Ethics

On the one hand, the interdisciplinary field of robot ethics is concerned with the impacts of robots on society, i.e. it deals with ethical questions related to the emergence of the robotic industry [2-4]. Although the main areas of concern of the discipline are military, social and medical robots, other types of robots should not be neglected [22]. For industrial robots and robotic assistants designed to cooperate with human workers in factories which are considered here, some guidelines should be issued for developers. The establishment of such guidelines for the design and development process of industrial robotic systems requires the collective effort of engineers, programmers, industrial managers, politicians, lawyers, ergonomists, health and safety specialists, economists and sociologists. Topics which need to be addressed in this context include but are not limited to:

- The changes brought about by the introduction of robots in the factory, i.e. changes in the work environment as well as in the operation of the plant. These impacts should be investigated from societal, financial, administrative as well as juridical viewpoints. I.a. the image of the company, the opinion of the employees, amendments in the administrative, legal status and in the internal regulations, short-term as well as long-term financial implications and risks for accidents or failures should be considered.
- The human-centric development. All research efforts should be done for the good of the human workers, i.e. the ultimate goal should not be to outperform and replace human workers but rather to help them by taking over functions which involve non-ergonomic work, hazardous environments or dumb and repetitive tasks.
- Safety. The safe operation of the robots with respect to the laws has to be guaranteed at all times. This issue becomes even more critical when HRI (Human-robot interaction) is involved.
- HRI, human-robot interaction. As soon as robots are no longer operating behind fences, but cooperating in time and space with humans, more factors need to be taken into account during the development process. These factors reach from human acceptance, division of labour over safety features to ergonomics of the interaction interface [23-26].

C. Machine Ethics

Machine ethics on the other hand, is concerned with the ethical behaviour of machines or robots. Rather than investigating the ethical consequences of robotic assistants for society and targeting the design process by human developers, machine ethics investigates the ethical behaviour of the robots and targets the ethical guidelines which are implemented.

The majority of machine ethics research is based on the three laws of robotics which the science-fiction author Isaac Asimov introduced in his novels mid-20th century [27]. The three laws are as follows:

'1) A robot may not injure a human being or, through inaction, allow a human being to come to harm.

2) A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

3) A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.' [28]

One application to cite is the work of the Anderson-couple, a philosopher and her husband, a computer scientist. They showed that even situations which at first sight might seem trivial involve complex ethical questions which in case of automation need to be treated in the framework of machine ethics. The banal example the couple has been investigating is the humanoid robot Nao reminding patients to take their medicine. The ethical questions which arise for this use case are among others: how to proceed when the patients are not obeying, when to intervene, which measures to apply to make them take their medicine, how often to remind them in order to find a balance between being useful, respectful of the patient's autonomy and annoying, disrespecting the free will of the patient [21]. The work of the Andersons illustrates that it might be desirable that multidisciplinary teams think the to be automated scenarios through beforehand. All options and eventualities should be envisaged and prepared. The advantages of such a preparatory work are a more controlled environment which inspires confidence to the humans involved. The downside would be a less flexible automation, i.e. a system which is unable to react and adapt to undesigned situations [27]. Additionally, it is probably impossible to foresee all eventualities and to program the robot accordingly. An alternative programming method would then be to teach the robot how to take decisions on what is good and what is wrong starting from a set of rules and to act correspondingly. Whereas this scenario is flexible and adaptive, the situation becomes less controllable and might seem less trustworthy to humans [27, 29]. What differentiates most between robots and humans in this regard is how they follow rules. While robots always strictly follow rules like they have been programmed to, humans are more likely to abandon logic or rules, mainly when emotions are involved. The trolley dilemma [27] points out the implications of both behavioral patterns. After concluding that machine ethics do are relevant and do require some research, the more specific questions on which ethical doctrines to use for the creation of ethical machines and whether the latter should be created at all in the first place, arise. [30] In his study, Tonkens illustrates this point with the help of Kantian theory. For the sake of consistency, the author argues that Kantian AMAs, artificial moral agents, should not be created at all. If, however it is agreed that robots need some ethical ground rules, it has to be investigated which ethical doctrine can be used and how to overcome the consistency-issues between the ethical theory and the creation of moral machines. These concerns are known as limitations of machine ethics based on ethics [30, 31]. Next to these constraints, limitations based on computational nature restrict the advancement of machine ethics. The programming and implementation of an ethical doctrine still requires some research.

III. SUGGESTED SOLUTION

The purely theoretical discussion of the ethical problem is followed by a more applied part where the gap is bridged between the theory and the practice. Directly related to the use case of a partially automated factory, some guidelines in the format of a framework for the cooperation of humans and robots are elaborated. The previous analysis has shown that all different secular and religious theories agree that their free will distinguishes human workers from their robotic counterparts. Their freedom with the resulting responsibility is the main characteristic of persons. In the context of Human Robot Interaction on the shop floor, this signifies that human workers have a different, higher rank in terms of rights and responsibilities than robots. As robots are not autonomous and therefore do not possess free will. This fact has two implications:

1) Robots cannot be considered persons worth of specific rights, i.e. they are subordinate to humans in this regard.
2) Robots are not responsible for their behavior. It is rather the human programmers and developers who need to take the responsibilities for the robots' acts.

The first point implies that robotic assistants are not granted the same rights than human workers. The second point implies that robots cannot be hold responsible for their actions as they are not free to choose what they do. As robots behave only like they have been programmed to, it is the human programmer who chose this behavior and who consequently is responsible for it. In this same line of thought, as far as safety and protection are concerned, the health and well-being of the human is most important. The outline of these regulations is consistent with the denomination 'robotic assistant'. Indeed, the goal of HRI is to assist and help the human worker. In this scenario, the human is the master and the robot its slave. The justification for the establishment of these guidelines for the cooperation of humans and robots in factories can be found in the confidence standards inspire in new, unknown situations. An official framework provides transparency, confidence, control and trust [32].

A. *Implementation*

The practical implementation of the proposed framework is the next step after the theoretical development of the possible solution to the analysed ethical problem. This is likely to be a slow process as it involves a wide range of stakeholders as well as the interference with and amendment of administrative procedures which might be cross-sector and/or cross-border. This paragraph therefore presents a 3steps-approach to the final implementation of the framework:

1) Awareness raise,
2) Dissipation of the suggested solution-concept,
3) Implementation on different levels.

This 3steps-approach is also illustrated in Fig.1. The first step involves identifying the stakeholders, i.e. the people concerned by a partial factory automation, and raising awareness of the identified ethical problem among them. The involved parties span different sectors from academia to industry. In the former, researchers from humanities as well as researchers from natural and applied sciences, disciplines addressing technological problems of automation are directly and indirectly concerned with ethics. Industrial stakeholders include both robot manufacturers and factory managers. Among the employees directly concerned by factory automation, programmers and manual workers are to be cited. Last but not least, general society should also be informed about the ethical problem at stake as well as about the solution process. Raising awareness can be done by direct discussions with stakeholders and through media: social media posts, written and oral reports for the broad public. The second step is dedicated to the dissipation of the developed framework. Following up on step 1, this can be included in discussions and broad public-articles. In addition, more scientific papers can be published in academic conferences and journals fostering a discussion among peers. The submission of a project proposal is another possibility to acquire attention, interest and funding for the issue at stake. Here, a collaboration between academia and industry where the manual workers are involved, is conceivable. The third and final step is concerned with the implementation of the framework and regulations to harmonize the cooperation of human and robots in the factories. This is happening step-by-step on different levels, i.e. on strategical as well as operational levels. After a proof of concept, i.e. a scientific work performed in a research setting, factory- and company-internal regulations can be put in place. The ultimate goal would be the ratification on a national or even international level.

Fig. 1: 3steps-approach for the practical implementation of the suggested solution.

IV. Discussion and Conclusion

This paper addressed the ethical questions resulting from the emergence of robots and artificial intelligence as well as from their introduction in today's society. The discussion of the effects of the emergence of the robotics industry implies the discussion of future issues in contrast to past or current ones. This future can be subdivided in three categories: now, near and far. The first one, now, considers the application of current state-of-the-art technology. It is concerned with the effects of the implementation in society of technologies which have already been developed and validated in a research environment. While the first scenario analyses taking known innovations from laboratories to industry, both, the second and third categories deal with technologies which have not been developed yet, neither in a research nor in an industrial environment. The difference between both categories is the amount of speculation involved. The second category, near, predicts future technological developments based on current state-of-the-art research and it can be considered as an informed prediction of the next results of the robotics community. The developments at stake are likely to happen in the near future and the category is therefore named near. The third category on the other hand deals with the possible technological developments of the further future. These predictions and their implications are much more speculative, only slightly based on facts and consequently less certain. In an attempt to eliminate speculation to the highest possible extent, in this paper only scenarios of the two first categories are considered, i.e. the paper deals with the ethical implications of the implementation of state-of-the-art and related technologies which are currently under development. In the case of ground-breaking developments in the field, the question would have to be reinvestigated.

A second limitation of this paper concerns the analysed sector. Indeed, only automation in manufacturing industry is considered. Although industrial robotics initiates less ethical questions than military or social robotics, it has been shown that partial automation of factories leads to ethical questions which require a detailed analysis and the suggestion of a thought-through solution.

This paper discussed in how far human and robotic workers are alike or unlike in a partially automated factory-context. It tried to establish a consensus in the ethical pluralism on the differences between human workers and their robotic assistants and to deduce a framework for their cooperation on common grounds. This theoretical analysis of the ethical problem and the suggestion of a solution was completed with a 3steps-approach for its practical implementation. The latter is only constrained by the personal sphere of influence.

Summarizing, it can be said that automation of manufacturing processes is a highly probable future scenario. However, at the moment and in the near future, we should only see partially automated factories with HRI, Human Robot Interaction. Indeed, although researchers are introducing smart or intelligent robot systems, smart here does not mean autonomous. Think in this respect about smartphones, despite being highly performant, smartphones are still far from being intelligent in the sense of autonomous. The introduction of intelligent autonomous robotic assistants on the shop floor is still speculative and therefore out of the scope of this paper. In a first phase, automation and HRI in manufacturing industry rather has the potential to bring back production to high-wage western countries through increasing productivity and lowering the need for manpower. In this sense, the introduction of robotics can be compared to the Industrial Revolution. If automation is well-led and frameworked, it can be as beneficial for the human workers as the Industrial Revolution has been by reducing working hours per day, working days per week and dispensing humans from some physically demanding, tedious and dangerous jobs. To make sure the introduction of robotic assistants is beneficial to the human workers and society as a whole, however, it is necessary to think ahead of time about the implications of automation, i.a. the ethical ones and to establish a framework for the optimal implementation and operation of robotic assistants alongside humans in factories.

Acknowledgment

This work was performed in the framework of the 'Building a Global Network for Sustainable Responsibility – the Lucerne Summer University: Ethics in a Global Context' under the patronage of UNESCO at the Institute of Social Ethics ISE of the University of Lucerne, Switzerland (04.-08.06.2018).

References

[1] I. D. Wyatt, D. E. Hecker, 2006, 'Occupational changes during the 20th century', Monthly Labor Review, 35.

[2] G. Veruggio, F. Operto, 2008, '64 Roboethics: Social and Ethical Implications of Robotics' in B. Siciliano, O. Khatib, 'Springer Handbook of Robotics', Berlin-Heidelberg: Springer-Verlag, pp. 1499-1524.

[3] P. Lin, K. Abney, G. Bekey, 2011, 'Robot ethics: Mapping the issues for a mechanized world', Artificial Intelligence, 175, pp. 942-949.

[4] M. Scheutz, 2013, 'What is Robot Ethics?', IEEE Robotics & Automation Magazine.

[5] P. della Mirandola, 1956, 'Oration on the dignity of man', trans: A. R. Caponigri, Chicago: Gateway.

[6] E. Mack, 2013, 'John Locke', New York: Bloomsbury Academic & Professional. Accessed July 20, 2018. ProQuest Ebook Central.

[7] J.-J. Rousseau, 1819, 'Discours', Paris: Madame veuve Perronneau.

[8] F. Rosen, 2017, 'Jeremy Bentham', Milton: Taylor & Francis Group. Accessed July 19, 2018. ProQuest Ebook Central.

[9] I. Kant, 1974, 'Grundlegung zur Metaphysik der Sitten', Werkausgabe Weischedel Band 7, Frankfurt am Main: Suhrkamp.

[10] B. P. Bowne, 1908, 'Personalism', Boston: Houghton, Mifflin.

[11] R. Spaemann, 2007, 'Personen: Versuche über den Unterschied zwischen etwas und jemand', Stuttgart: Klett-Cotta.

[12] D. Sulmasy, 2013, 'The Varieties of Human Dignity', Med. Health Care and Philos., 16, pp. 937-944.

[13] D. Heyd, 2003, 'Human Nature: An Oxymoron?', Journal of Medicine and Philosophy, 28-2, pp. 151-169.

[14] L. *Mlsna, 2010, 'Stem Cell Based Treatments and Novel Considerations for Conscience Clause Legislation'* Indiana Health Law Review. *United States:* Indiana University Robert H. McKinney School of Law. *8 (2): pp. 471–496.*

[15] K. Abou El Fadl, 2017, 'Qur'anic Ethics and Islamic Law', Journal of Islamic Ethics, 1, pp. 7-28.

[16] M. Düwell, J. E. Braarvig, R. Brownsword, D. Mieth, 2014, '*The Cambridge Handbook of Human Dignity',* Cambridge University Press.

[17] E. Berkovits, 1973, 'Faith after the Holocaust', KTAV Publishing House.

[18] M. Kapotoglu, C. Koc, S. Sariel, 2015, 'Robots Avoid Potential Failures through Experience-based Probabilistic Planning', 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2015), pp. 111-120.

[19] H. Yi, 2015, 'A Sliding Mode Control Using Brain Limbic System Control Strategy for a Robotic Manipulator', Int J Adv Robot Syst. 12:158.

[20] *M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Van de Wiele, V. Mnih, N. Heess, T. Springenberg, 2018,* 'Learning by Playing-Solving Sparse Reward Tasks from Scratch', arXiv:1802.10567v1 [cs.LG]

[21] M. Anderson, S. L. Anderson, 2011, 'Machine Ehics', Cambridge University Press.

[22] IEEE Robotics & Automation Society, Technical Committee for Robot Ethics. (http://www.ieee-ras.org/robot-ethics accessed 20.07.2018)

[23] A. M. Zanchettin, L. Bascetta, P. Rocco, 2013, 'Acceptability of robotic manipulators in shared working environments through human-like redundancy resolution', Applied Ergonomics, 44, pp. 982-989.

[24] *A. Montebelli, F. Steinmetz, V. Kyrki, 2015,* 'On Handing Down Our Tools to Robots: Single Phase Kinesthetic Teaching for Dynamic In-Contact Tasks', Proceedings-IEEE International Conference on Robotics and Automation, pp. 5628-5634.

[25] *M. P. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, O. Madsen, 2016,* 'Robot skills for manufacturing: From concept to industrial deployment', Robotics and Computer-Integrated Manufacturing, 37, pp. 282-291.

[26] L. Rozo, J. Silvério, S. Calinon, D. G. Caldwell, 2016, 'Learning Controllers for Reactive and Proactive Behaviors in Human-Robot Collaboration', Frontiers in Robotics and AI, 3, pp. 1-11.

[27] B. Deng, 2015, 'The Robot's Dilemma-Working out how to build ethical robots is one of the thorniest challenges in artificial intelligence.', Nature, 523, pp. 24-26.

[28] I. Asimov, 1951, 'I, Robot', New York: Gnome Press.

[29] W. Wallach, C. Allen, 2010, 'Moral Machines-Teaching Robots Right from Wrong', Oxford University Press.

[30] R. Tonkens, 2009, 'A Challenge for Machine Ethics', Minds & Machines, 19, pp. 421-438.

[31] M. Brundage, 2014, 'Limitations and risks of machine ethics', Journal of Experimental & Theoretical Artificial Intelligence, 26-3, pp. 355-372.

[32] J. J. Bryson, A. Winfield, 2017, 'Standardizing Ethical Design for Artificial Intelligence and Autonomous Systems', Computer, 50-5, pp. 116-119.

Programming by demonstration using fiducial markers

Deom Nathan[1] , Olivier Brüls[1] and Thierry Jacques[2]
[1]Department of Aerospace and Mechanical Engineering, Univeristy of Liège, Belgium
[2]Citius Engineering, Belgium
{npdeom,o.bruls}@uliege.be

Abstract— **The programming by demonstration is a method that allows one to register a trajectory and to reproduce it by a robot. It could be used to speed up the programming of robots using the practical skills of workers without the time consuming code development part of classical robotic applications. This article exposes a methodology based on fiducial markers and only one classical camera which is an inexpensive solution. The fiducial markers are based on binary symbols generally used for augmented reality applications. It also presents the experimental setup created at the laboratory of human motion analysis and used to analyze the repeatability and the precision of this solution. The primary results show a promising technology and following developments will be done.**

I. INTRODUCTION

Programming by demonstration is a research field that aims to transfer the human skills to robots. Instead of programming, the end-users teach the desired behaviour to the robot. This method is composed of two phases: the teaching and the reproduction. During the teaching phase, the user shows the desired action to the robot. Afterwards, the robot programming can be automatically determined. This methodology allows one decreasing the programming time and taking advantage of the technical and adaptive skills of the end-user.

The teaching phase requires the registration of the motion. Several methods can be used [1]. For a collaborative robot, hand-guiding, which is also named lead-through programming, can be considered. However, for a classical industrial robots, the interactions with the human are restricted so other techniques should be used, such as human motion analysis methods. The measurement system could be

- a mechanical system with position encoders [5],
- a magnetic system using triangularization (e.g., ABB Simplified Robot Programming),
- an inertial system using inertial measurement units [4], [7],
- an optical system which uses cameras.

The optical systems are generally based on markers tracking, however, with the recent increase in computation power and artificial intelligence, some markerless methods also appear.

Once the teaching phase completed, the raw data should be processed to generate the robot programming. The processing could be more or less advanced depending on the needs of the final application. If required, the trajectory should be closely continuously followed as in welding [9], painting [8] or teleoperation [7]. In other cases, like pick-and-place, it is sufficient to reproduce the general behaviour [10]. Consequently, the starting point and the end point are the only interesting information of the recording.

The fiducial markers allow camera pose estimation and are generally used for augmented reality and robot localization. Several fiducial marker systems have been proposed, [2], [3]. They differ by the generation method of the marker codes but the camera pose estimation are similar, using perspective by n points problem described below. The pose (translation and rotation) of a marker is computed in the camera frame.

II. METHOD

The programming by demonstration for painting or welding application requires the tool pose measurement during the whole operation. In this work, the proposed technique is based on the computation of the tool pose on which fiducial markers are fixed using only one camera.

A. *Fiducial markers used : ArUco*

The fiducial markers selected for the project come from the ArUco library, since it is implemented in OpenCV, Open Source Computer Vision [11], which is an open source library for image and video analysis. Consequently, the ArUco library offers image processing algorithms to detect and identify the markers.

The ArUco markers are square fiducial markers composed of a binary matrix (white and black) and a black border, as it can be seen in Fig. 1. A set of markers composes a dictionary which is defined by the size of the markers sides and the number of markers. The matrices of each marker are different since they represent binary codifications selected to maximize variations between the markers, in order to easily identify the marker.

To compute the position of one marker in the camera frame, the process requires the camera parameters (distortion coefficients and camera matrix), the 2D positions of the four corners in the image and the 3D positions of the corner in the marker frame. Using these values, the Perspective by n Points (PnP) problem can be solved with n equal four which has theoretically a unique solution if the points are co-planar [12].

In a similar way, if several markers are placed on the tool, the pose of this one can be obtained using the 3D positions of the corners in the tool frame and the 2D positions of corners obtained from visible markers. The usage of more

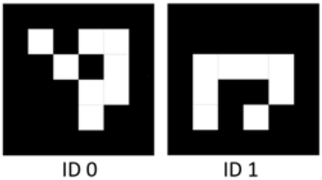


Fig. 1: Examples of the ArUco markers using the predefined dictionary DICT_4x4_50, corresponding the fifty binary codified matrix of four by four

than one marker permits the detection with more orientations, since the pose can be estimated as soon as one marker is detected. The markers used to solve the problem are the one that are identified using the binary code. Then appropriate 3D positions can be used for solving the PnP problem.

The flowchart shown in Fig. 2 present the methodology used to compute the tool pose in the camera frame. The possible markers are extracted from the image. They are represented by the four corners positions. After the detection, the identification step provides the corresponding number of a marker in a specific dictionary. Using the model of the tool, the 3D positions corresponding to the corners seen are selected. The PnP problem can then be solved to obtain an estimation of the tool pose.

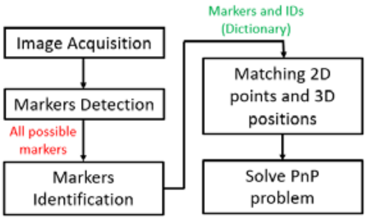


Fig. 2: Flowchart to compute the position of the tools in the camera frame, using several ArUco marker defined by a dictionary

B. *Motion analysis and robot reproduction*

In a programming by demonstration application, the correspondence between the motion of the tool in the frame of the camera and the motion of the robot in his base frame should be known. Moreover, if the programming by demonstration should be used for industrial robots, the interaction between the human and the robot must be limited. Consequently, the cell of the robot and the recording cell should be separated as shown in Fig. 3. A fixed marker is used to determine the working frame of the recording cell, and the camera can be move at a suitable position to record the motion. Making a correspondence between the fixed marker and the robot frame leads to the desired path for the robot.

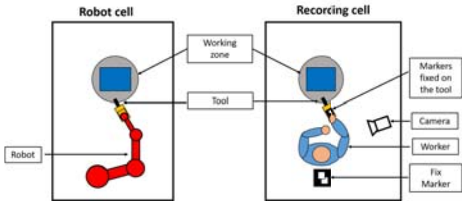


Fig. 3: Comparison between the recording cell and the robot cell

III. EXPERIMENTS

A. *Experimental setup description*

In order to evaluate the system, an experimental setup is created at the Laboratory of Human Motion Analysis of the University of Liege[1]. Four 3D optoelectronic systems, *CX1* CODAmotion, based on active markers are used and considered as a gold standard reference system. In the following, the position of an object measured with these systems is considered as the ground truth. The markers are fixed on 3 elements: the camera, the fixed marker and the tool. It provides the position of the fixed marker and of the tool in the camera frame from which an estimation of the precision and the repeatability could be computed. The experimental setup is illustrated in Fig. 4

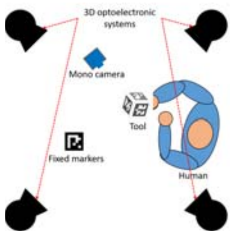


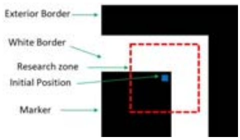
Fig. 4: Experimental setup developed at the Laboratory of Human Motion Analysis of the University of Liege and used to estimate the precision and the repeatability

B. *Error sources*

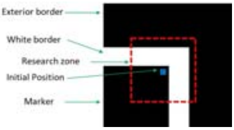
There are two main types of errors. They could be due to the distortion or to the numerical treatment of the images. The distortion means that the image is not a perfect perspective projection due to the usage of lenses and a calibration procedure is required which involves camera internal parameters[13]. However, some errors could be remain and the calibration may not be perfect. The second type of errors are due to the numerical resolution of the PnP problem and the determination of the corner positions.

[1] http://labos.ulg.ac.be/lamh/

1) The subpixel corner detection: The corner detection is improved using a subpixel method which should give a better estimation of the corner positions. The image is a discretization of the scene, so the initial guest for a corner position corresponds to a pixel. However, the real position of the corner is more precise using a subpixel method because the position is estimated using the pixels around the original guest to refine it. Nevertheless, the subpixel position is still an estimation which is degraded if the marker size decreases while keeping it at the same distance and if the distance increases while the marker size stays the same. Moreover, the marker should be surrounded by a white border to simplify its detection but if the border is to small it could induced some problems during the subpixel detection. The corner detected could be the external corner and not the corner of the marker. This situation is ambiguous, as it can be seen in Fig. 5. It can lead to important errors in some particular cases.



(a) Correct situation



(b) Ambiguous situation

Fig. 5: Two situations that could appear using the subpixel method

Generally several markers are detected in the images, so the errors of corner detection are compensated. However, if only one marker is detected, it can lead to strong orientation error. Even if the PnP problem has theoretically a unique solution, two orientations give similar images when they are projected. So a small error of corner detection can lead to the wrong orientation as it can be seen in Fig. 6.

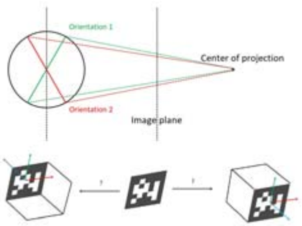


Fig. 6: The Z-flipping problem representation

In conclusion, to avoid the error due to the corner detection, several improvements can be used: the markers should be as big as possible with a large white border, the markers should be close to the camera, several markers should be detectable on an image or the image resolution should be increased. Also, the camera calibration should be done carefully with a camera that has small distortion.

C. Static measurements and quantification

Some static measurements have been done with the experimental setup. Twenty-five configurations have been measured with orientation and position variations and 100 images have been taken for each configuration. It allows evaluating the precision and the repeatability. The position error is computed using the formula $\| T - \hat{T} \|$ where T is the position vector obtained with the Codamotion system and $\hat{T}$ is obtained with the fiducial markers. For the rotational error, this formula can be used $\theta = acos\big((Trace(R^T\hat{R}) - 1)/2\big)$ where R is the rotational vector obtained using the Codamotion system and $\hat{R}$ is obtained with the fiducial markers.

The repeatability of the measurements is smaller than one millimetre if there is no error of corner detection. When some errors appear, it is in the range of 2-3 millimetres. However the precision is not as good as expected since it is in the range of the centimetre and around two degrees of errors. However, it seems clear that there is some calibration error since the precision of the tool and the fixed marker in the camera frame are worse than the precision of the tool in the frame of the fixed marker. In the three cases, the errors are in the range of the centimeter.

A more accurate procedure should be defined to study this phenomenon in static and in dynamic configurations. It will be done in a future paper.

IV. CONCLUSION

The programming by demonstration aims at bringing the practical knowledge of workers to the robot without the long development required by classical programming method. Consequently, it could be used to program robots used with small batch size, since the program could be changed easily. It requires a motion recording phase which can be done using an optical system. However, the optoelectronic systems are generally costly since they require several cameras and a synchronization system. The methodology presented in this article is based on a single 2D camera with square markers which is a cost-effective solution.

The first experimental measurements are promising. However, to validate the methodology, a study with dynamic motion should be done. Afterwards, a practical case should be realized using a robot to reproduce a recorded trajectory. It could require an adaptation since some motion done by the human could not be reproducible by the robot.

ACKNOWLEDGEMENT

The first author would like to acknowledge the Belgian Fund for Research training in Industry and Agriculture for its financial support (FRIA grant).

REFERENCES

[1] A. Billard, S. Calinon, and S. Schaal, "Springer handbook of robotics," Choice Rev. Online, vol. 46, no. 06, pp. 46-3272-46–3272, 2013.

[2] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognit., vol. 47, no. 6, pp. 2280–2292, 2014.

[3] J. Wang and E. Olson, "AprilTag 2 : Efficient and robust fiducial detection," pp. 2–7.

[4] R. Pellois, L. Joris, and O. Br, "Robot control based on human motion analysis with IMU measurements."

[5] "Robot Learning from Demonstration in Robotic Assembly: A Survey," Robotics, vol. 7, no. 2, p. 17, 2018.

[6] J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning Attractor Landscapes for Learning Motor Primitives," Adv. Neural Inf. Process. Syst. 15, pp. 1547–1554, 2002.

[7] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," Mechatronics, vol. 55, no. March, pp. 248–266, 2018.

[8] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognit., vol. 47, no. 6, pp. 2280–2292, 2014.

[9] F. Röhrbein, G. Veiga, and N. Ciro, "Gearing Up and Accelerating Cross-Fertilization between Academic and Industrial Robotics Research in Europe: Technology Transfer Experiments from the ECHORD Project ABC," Springer Tracts Adv. Robot., vol. 94, no. January, 2014.

[10] P. Pastor, H. Hoffmann, T. Asfour and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," 2009 IEEE International Conference on Robotics and Automation, Kobe, 2009, pp. 763-768. doi: 10.1109/ROBOT.2009.5152385

[11] Bradski, G.," The OpenCV Library," Dr. Dobb's Journal of Software Tools, 200

[12] Y. Zheng, Y. Kuang, S. Sugimoto, and K. Åstr, "Revisiting the P n P Problem : A Fast , General and Optimal Solution," 2013.

[13] Z. Zhang. "A Flexible New Technique for Camera Calibration." IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.

Modular concept for assistance functions and extension of the working area of an HRC system

Prof. Dr.-Ing. Rainer Müller, Dr.-Ing. Matthias Vette-Steinkamp, Dipl.-Ing. Daniela Schmidt* and M.Sc. Fabian Adler*

ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gemeinnützige GmbH
D-66121 Saarbrücken, Gewerbepark Eschberger Weg 46 , Gebäude 9, Germany
d.schmidt@zema.de, f.adler@zema.de

***Abstract* - At ZeMA – Zentrum für Mechatronik und Automatisierungstechnik – the group assembly processes and automation work on industry-related research domains in the fields of robotics, planning and tolerance management.**
The scope of this work is to develop a prototype of a semi-automated physical human-robot cooperation (pHRC) station with a modular control concept which standardizes human robotic interfaces. The collaborating characteristics of the robot are conceptually transferred to a linear axis for workspace extension. This enables skill-based task sharing between human and robot in an extended workspace. Safety is considered through guaranteeing the stopping of the robot through the capacitive sensor skin when people or obstacles approach in range. The signal of the sensor skin is transmitted in parallel to the linear axis to set it into emergency stop status. The HRC station, presented in this paper, serves as a basis for further researches in the fields of pHRC workspace extension and a modular control concept. For first investigations, a drilling process is implemented. In this demonstration, a handheld cordless drilling machine is retrofitted to be "Industry 4.0-Ready". Additional assistance systems, such as "guide by spotlight" for example, are modular in design and integrated via standardized interfaces based on MQTT. For these reasons a simple integration, even in existing systems, is possible. The assistance systems support the exchange of information with humans, increase the acceptance between robots and humans. Due to the intuitive implementation of the human-machine-interaction (HMI), the resources are shifted from not seeing the robot as an "instrument" but as a "partner" within the framework of human-robot-cooperation.

Index Terms* – *Industry 4.0, HRC, extended Workspace, Retrofit, modular control concept

I. INTRODUCTION

Human-Robot-Cooperation (HRC) is especially interesting for semi-automation in assembly processes. The hype about HRC began with the market launch of the UR in the year 2012 and is primarily the result to the low acquisition costs, intuitive programming and new safety features of the system. Economic and workplace ergonomic improvements are realized by new working types (Fig. 1). The introduction of the regulation TS15066 enable new cooperation and security areas, which allows direct physical, fenceless cooperation between humans and robots and combining the individual capabilities of humans with the repeatability and endurance of robots. [1] In order to be able to process large components cooperatively or to integrate the HRC system into the assembly line, this paper will present a concept for integrating a linear axis into the collaborative workspace.

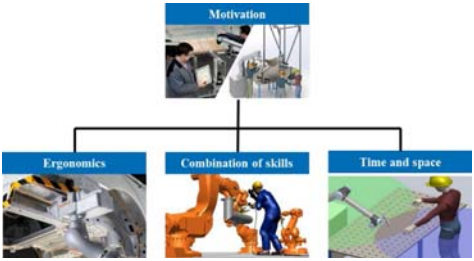


Fig. 1 Motivation for HRC use [2]

The additional axis initially requiresan unpredictability of the extended robot system, which reduces the human trust and situational awareness of the system. Therefore, the system will be enhanced with a plug & play, Industry 4.0 based modular interaction between human and machine that can be quickly integrated.

Industry 4.0 and industrial Internet of Things (iIoT) allow the connection and integration of devices with different hardware, functionality and software in one system. The scheme for the system connectivity deviates from the classical automation pyramid. The integrated devices and services in a system communicate more and more with each other. To prevent industrial change from leading to an increase in the complexity of the introduction of such systems modular cyber-physical system (CPS) must be formed, which can be integrated with plug & play.

II. STATE OF THE ART

In production, four cooperation solutions between humans and robots can be classified in addition to the conventional use of robot systems with protective fences (Fig. 2).

Autarkic operations describe production without a safety fence in which the human and the robot are separated locally. In "synchronized mode", temporally separated sequences enable alternating work in the same workspace. Collaborating robots must be used in cooperative and collaborative work sharing. In this case, robots and humans work simultaneously and in the same workspace, whereby a common task is performed in collaborative mode. [3]

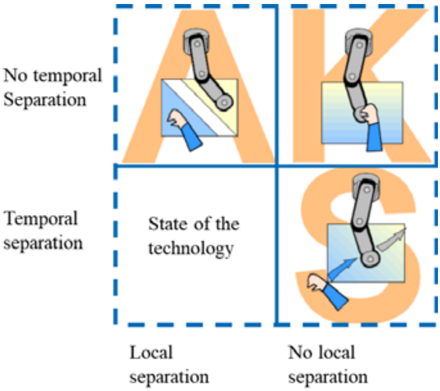


Fig. 2 Four stages of human-robot- collaboration [3]

A. Human-Robot-Collaboration

In the implementation of an HRC application, the acceptance of the worker must be guaranteed. This is achieved by involving the worker at an early stage in the decision-making process and by an intuitive, transparent interaction between robot and operator. [2] In order to guarantee the safety function of the system, the collaborating mode differs between the following four methods: [4]

- Safety evaluated monitored stop
- Hand guidance
- Velocity and distance monitoring
- Power and force limitation

In the following, the safeguarding of the cooperation area with speed and distance monitoring is discussed more precisely. For example, the position of the person, speed and distance in relation to the robot can be recorded via external sensors such as light barriers, safety mats, etc. Research is currently studying methods (camera-based 3D sensors [5] [6], ultrasound and radar) for collision prevention. [7]

Employers' liability insurance association (BG) certified and already in industrial use is an HRC solution which is capable of contact detection and prediction via a sensor skin. [3] This robot is equipped with sensors that generates a capacitive field. The principle is based on changing the electrical capacitance of a capacitor system in response to electrically conductive material or a dielectric in the immediate vicinity. This capacitive change leads to a stop of the robot movement as soon as the switching distance is reached. In addition, the sensor skin has an elastic pad that absorbs most of the collision energy. If the human moves out of the switching distance, the movement of the robot is continued again. [8]

Collaborating robot systems are subject to the European community (EC) Machinery Directive and must be equipped with an EC Declaration of Conformity and a CE mark. The robot system consists of robots, tools, workpieces and fixtures, which together form a machine according to the EC Machinery Directive. [9] [10]

The robot and all connected safety-related functions must be triggered when the safety distance becomes too low. In order to achieve a performance level d Cat. 3 [11], redundant monitoring of electrical outputs is required for safety-relevant functions.
Robots and axis must have the following characteristics in order to be able to guarantee safety technology for speed and distance monitoring:

- Safety-rated monitored velocity
- Safety rated monitored stop
- Safety-rated software for axis and space limitation
- The distance monitoring system shall comply with the requirements of [9] [10]
- Safety-related functions connected to the robot system must trigger (switch off all dangerous tools)

B. Awareness for Human-Robot-Collaboration

The model according to Endsley can be used to determine the proper awareness of a situation. It requires humans to perceive the objects in the environment, understand the meaning of the objects, and predict how the environment and the future state of the object will change for a sufficient time interval. Based on this information, humans make process-specific decisions, execution plans, and actions. [12]

Therefore, it is important that the human operator is involved in the semi-automated process with meaningful, complete and well thought-out tasks and that there is communication between human and robot so that loss of situational awareness and trust in the robot does not occur. [13] [14]

Due to the progress of the available technology today, the automated system can change the mode during operation due to external influences or safety functions. To avoid operator errors, the operator needs to know when and how the mode can be influenced and what the functions are in each mode. This is called mode-awareness and must be preserved in system automation. [15]

C. Communication Protocols

Generally defined, interaction is the transmission or exchange of information. In computer science, communication consists of a physical and software component. The components are subdivided into technology, topology, underlying communication protocol and information flow (central/decentral). [16] With the introduction of iIoT and Industry 4.0, Application Layer IoT Protocols (MQTT, AMQP, CoAP) are useful for applications or machine communication. AMQP - Advanced Message Queuing Protocol - is an open-standard message protocol that manages the queue, routing and alignment of messages. CoAP - Constrained Application Protocol - is a web transfer protocol designed specifically for devices with limited resources (small memory, short battery). Official OASIS standard is the machine-to-machine connectivity protocol MQTT - Message Queuing Telemetry Transport - and distinguishes itself from AMQP and CoAP in terms of the Message Oriented Approach (MOA). MQTT uses TCP as transport layer and TLS/SSL as security layer. [17]

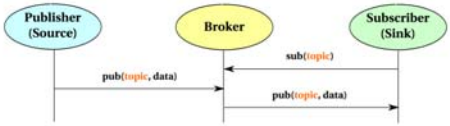


Fig. 3 Publish & subscribe pattern [18]

The basic functionality of MQTT is to publish/subscribe to a pattern (Fig. 3). For communication between the devices (publisher, subscriber) a broker is required, which manages the incoming messages from a publisher and passes them on to a subscriber if required. For assignment, the devices publish with a specific topic, which are defined as strings, buffers or JSON objects. For structuring purposes, any number of topic hierarchy levels can be created using forward slashes. To receive the message with another device, the device is subscribed to the broker with the topic of the published message. [19] [20] MQTT offers basic end-to-end Quality of Services (QoS). [21] The QoS level indicates how reliably the messages are to be delivered to the subscriber. At the lowest setting there is no guarantee that the message will arrive. The highest QoS level guarantees the arrival of the message. [18]

By decoupling all communication participants, the simplicity of use and the resource-saving properties, MQTT offers the possibility as a communication basis for many new modern machine systems and is the standard protocol for the iIoT. [22]

Other well-known communication protocols are for example OPC-UA - Open Platform Communications-Unified Architecture - and DDS - Data Distribution Service. OPC-UA is an IEC standard and is a manufacturer-, operating system- and programming language independent communication protocol. A Service Oriented Architecture (SOA) is used. A device makes a request to a service provider, whereupon the service provider sends the response back to the requesting device. The request & response behaviour is standardized with OPC-UA and enables platform independence.[23] DDS is a based on publish-subscribe concept that supports deterministic resource management and is real-time capable. It is a middleware for data-centred communication in highly dynamic distributed systems. [24]

iIoT applications can be implemented with Node-RED, a graphical development tool based on a modular principle. By linking function blocks, applications can be programmed, controlled and visualized. The function blocks cover the most common technologies. [25]

III. Concept and Construction of the HRC system

The concept of the HRC system is used on the hardware side as a basis for research of the working area-extension while ensuring HRC capability and on the software side for analysing the modularization of an HRC robot system. Due to the modularization of the software, it is possible to easily integrate and commission new components such as assistance functions to support the operator during processes. The iIoT standard MQTT is used for this purpose.

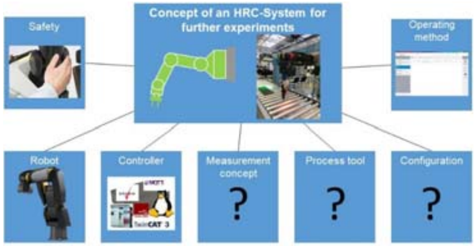


Fig. 4 Components for a (semi-) automated system

An application-oriented robot system can be divided into seven blocks (Fig. 4). The HRC robot system developed in this paper serves as a research basis for various problems. The basic components are a safety concept to make moving components HRC safe, a HRC-capable robot, a control concept and an operating method. Measurement concept, process tool and configuration must be adapted for the problem that has to be solved.

A. *Hardware concept*

To extend the working area, the collaborating robot is mounted on a linear axis. As shown in Fig. 5, a working area is created which is extended by the length of the linear axis. The ideal robot motion in the operating area can only be achieved if the height of the linear axis, end effector and robot kinematics are well related to each other. This must be taken into account in the design of the system in order to guarantee optimum machining of the workpieces and accessibility of the robot.

The safety concept is to be implemented as described below. If the robot stops due to the triggering of the safety skin, the trigger signal should also be transmitted to the axis and end effector and their movements have to stop. Depending on the end effector, it must be decided how the stop signal is to be processed and how a safe state can be established. The operator should be informed about the reason for the stop. The skin-triggered stop remains until the sensor skin is released again and the operator has safely moved away from the switching distance. Afterwards the robot, axis and end effector continue to execute their program.

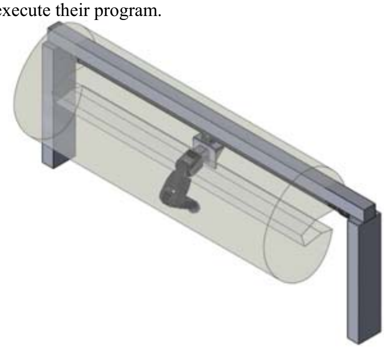


Fig. 5 Expanded working area of the robot on a linear axis

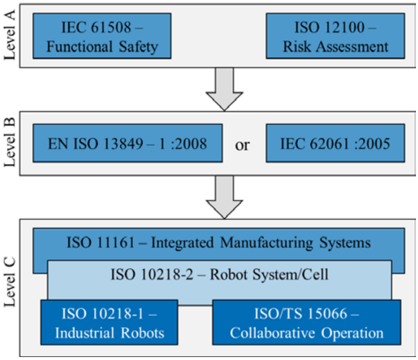


Fig. 6 Overview of the different safety ISO standards [2]

If a safety function of the system is triggered, e.g. triggering of the collision element, error on the axis, end effector or robot, the entire system should change into a safe mode. The system components are superimposed by the emergency stop of the system. [26] Furthermore, it must be ensured that nobody enters the close range of the axis, that there is no danger from the moving guide carriage and moving parts of the linear axis.

All relevant safety standards must be observed and applied in order to use the system industrially (Fig. 6). The basic safety standards "Level a" deal with safety aspects applied to machines and describe the risk analysis and functional safety of a system. "Level b" standards (safety generic standards) cover a safety aspect or a type of protective devices such as the creation of a risk graph, or the design of the emergency stop functionality. Special safety requirements for specific kind of machines are defined in "Level c". [27] This includes DIN EN ISO 10218-2 [10] which describes the safety requirements and refers to the corresponding standard for calculating the safety distances depending on the protective device. For example, for separating protective devices on EN ISO 13857 or for non-contact protective devices on DIN EN ISO 13855. ISO/TS 15066 defines the requirements for safe collaborating operation. This technical specification defines the biomechanical load limits for power- and force-limited operation. For speed and distance monitoring, the safety distance between moving robot and human must not be exceeded.[4]

B. Hardware implementation

The HRC system is based on the individual components linear axis and a Bosch APAS robot.

An ITEM linear unit of 4.5 m length with toothed belt drive is used. The toothed drive belt is form-fit to the motor driven pulley. The toothed belt drive is therefore highly dynamic and has a short cycle time. This is driven by a permanent magnet excited three-phase synchronous servo motor. [28] The control is done by an ITEM controller. This is enabled via the I/O interface. The positions, parameters for starting and accelerating the linear/lifting unit, are implemented via an EtherCAT interface. [29]

The APAS (Automatic Production Assistant) is composed of the industrial robot Fanuc LR Mate 200 iD/7L and the capacitive sensor skin (switch cabinet, touch panel, sensor skin unit). The APAS has a range of 911 mm and a wrist capacity of 2 kg as well as a total weight of 27 kg.

If the robot is in collaborative operation, the sensor skin is active. The minimal safe switching distance from the sensor skin is 50 mm. The maximum speed is reduced to 0.5 m/s. The end effector is connected via the collision element. This is a compensating element which triggers a safety-rated monitored stop when force is applied. [30] The **s**afety **sk**in and the control **u**nit (SSKU) must not exceed the cable length of 2.5 m due to their sensitivity to interference signals. The SSKU is therefore mounted next to the robot on the linear axis.

The reaction time between the detection of the human being in the "close range" and the standstill of the machine must be determined for the safety consideration. The safety distance is the result of:[4]

- the position uncertainty of the robot, the linear axis and the operator
- penetration distance (distance at which the worker is in the detection zone but has not yet been detected)
- breaking distance of the robot/ axis
- robot/ axis response time (robot speed integrated over stopping time)
- position change through operator (estimated 1.6 m/s in direction of robot movement if not monitored)

According to the manufacturer, the detection of the operator, signal transmission and processing to stop the robot should be performed safely 50 mm before collision. Therefore, the axis must not have a stopping distance of more than s(t) = 50 mm. At a speed of v_1 = 500 mm/s, the boundary conditions such as minimum stopping acceleration and stopping time can be calculated as follows (Fig. 7):

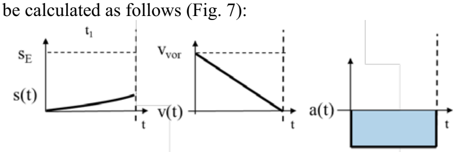


Fig. 7 Deceleration ramp path for trapezoidal velocity curve [29]

For the deceleration curve (1):

$$s(t) = \frac{1}{2} x a \Delta t^2 + v\Delta t + s_o \qquad (1)$$

results with:

$$t = \frac{v_1}{a_1} \qquad (2)$$

the distance which is required for braking as a function of time:

$$s(t) = \frac{1}{2}\Delta v_{vor} t = \frac{1}{2}\frac{{v_1}^2}{a_1} \qquad (3)$$

with final state s_E, initial velocity v_1, braking acceleration a_1 and braking time t. [31]

Minimum brake acceleration for reaching standstill after 50mm results from (3) with a = $2500\frac{mm}{s^2}$ and resulting braking time t = 0,2s.

If the axis cannot reach the braking acceleration or the response time is too high, the speed of movement of the axis can be reduced, for example.

B. *Control concept*

In addition to the software used, the communication properties technology, topology, underlying communication protocol and information flow are decisive for implementing the modularization of a system. Implemented technologies are EtherCAT and TCP/IP. EtherCAT is used for real-time-critical processes. For the remaining wireless processes, the participants get connected over TCP/IP. The underlying topology is a star-shaped network. This means that the messages are converge at the server and are distributed from there (Fig. 8). This results in a central flow of information. MQTT can be used as the communication protocol for such a setup and integration of sensors/actuators. The server represents the broker and the message flow. The sensors are publishers. The actuators are subscribers. The PLC is both publisher and subscriber and currently commands all participants in the system. Since the data flow converges at the server, it serves as a human-machine-interface for controlling the actuators and querying or providing the sensor data.

The HRC-capable robot and the safety controller of the robot are controlled by the PLC via EtherCAT. TwinCAT 3 is used as the programming environment for the robot. Opcon Plus from Bosch is used to generate the interface between the robot controller and the PLC. At the same time, Opcon Plus provides a self-configurable human-machine-interface (HMI) with which the robot can move and teach in point lists.

The modules are integrated into the system via MQTT. As the interfaces, Raspberry Pis and Node-RED are used for configuration. This makes it possible to integrate devices that are not MQTT-capable. Node-RED enables each device to work and respond independently, as well as modularization and integration of the devices in the system. Actuators have MQTT in-/outputs and sensors have only outputs. Node-RED dashboard function blocks make it possible to display information or interaction via a web-based dashboard. This can be used on each device separately or bundled on the server to the human-machine interface, as long as it is in the same network.

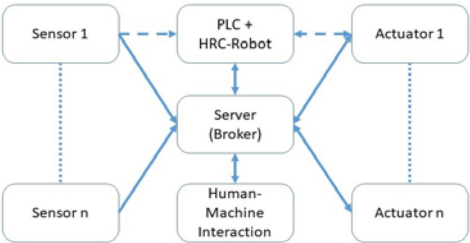


Fig. 8 Control concept for a modular design

IV. Use case

An application example is used to test the selected implementation steps (Fig. 9). The basic setup consists of the previously introduced HRC robot (2), which is fixed to a linear axis (3). Thus, the robot work space is extended along the roller conveyor. When the sensor skin is triggered, this is indicated in yellow by the status lamp (6). Error messages from the robot are indicated red of the status light. The end effector is a self-designed cordless screwdriver (4), which is controlled by a Raspberry Pi. A kind of "retrofitting" is accomplished. The torque can be adjusted on the software side and the torque limitation is adjusted mechanically. Two moving heads (1), which project the working areas of the robot and human are also controlled via a Raspberry Pi, serve as an assistance function to increase the operator's confidence and understanding of the system. The Raspberry Pis are controlled by the PLC of the robot. They are controlled via MQTT protocol. By calibrating the moving heads, the spotlights can be moved through Cartesian coordinates in space. To address the moving heads, an array of the form [x,y,z,n] is entered into the Raspberry Pi, where "x, y, z" are the Cartesian coordinates of the workspace and "n" stands for moving head one or two. On the Raspberry Pi the transformation of the coordinates into bytes takes place. Also, cordless screwdrivers and moving heads can be addressed via a web-based dashboard (5) that communicates via MQTT and was created using Node-RED.

V. Summary and Outlook

The presented state is the base for continuing research in the field of workspace extension of an HRC system without losing the HRC capability. Also the modularization of systems in the Industry 4.0 and iIoT context were created. Therefore a HRC robot was mounted on a linear axis. Through an initial conceptual safety consideration and implementation, the linear axis, like the robot, stops on contact with humans and resumes movement only after the contact has been released and acknowledged. With the first implementations, a modular control concept was implemented using MQTT. Node-RED offers an accessible implementation with possibilities for the creation of human-machine-interaction through interfaces.

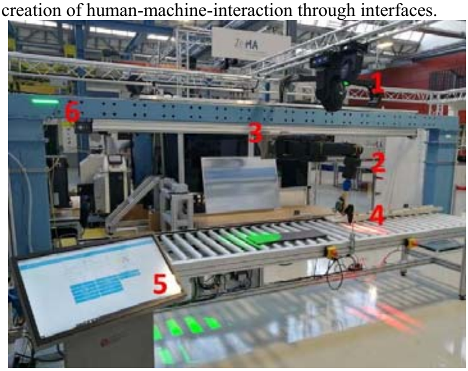


Fig. 9 Use case for first validation

For initial tests, a cordless screwdriver and two moving heads were installed to display the workspace for the robot and humans under the aspect of "retrofitting". The assistance functions were controlled via MQTT using two Raspberry Pis in Node-RED and allowed feedback from the system to increase transparency for the operator.

During the development of the first motion sequences, it was noticed that the selection of the working area is important to avoid the self-triggering of the capacitive sensor skin of the robot. On the hardware side, it must also be observed that some signals, such as those of the capacitive sensor skin, are very susceptible to interference and thus cannot be extended at will (max. 2.5m), so the SSKU (sensor skin unit) had to be installed on the linear axis, for example. This may also be a challenge with USB for camera connection.

Time-critical movements/functions implemented via MQTT show limitations in the speed of the data transfer. This must be taken into account in the programming and in the further consideration of the control concept. In addition, the safety concept has been indicated up to now and must be applied in the following, for example, for the end effector and the close range of the linear axis.

After successful commissioning, the next steps are the transfer to real application use cases in the field of HRC workspace extension, increasing the acceptance/trust of the operator for the HRC system, safety study (validation of the given data against real data) and control concept study. For the implementation, the measurement concept, process tool and configuration areas (Fig. 4) are not defined and must be adapted to the application use case. In addition, further assistance functions such as the control of smart cameras via MQTT shall be integrated into the control concept. The individual participants shall be controlled via a server using a state machine. Accordingly, the server would be responsible for the complete control and each participating device in the system could function as an independent unit.

VI. References

[1] DGUV: FB HM Informationsblätter

[2] Prof. Dr.-Ing. Rainer Müller: Robotix Lecture. Development of HRC Robot Systems. Universität Luxembourg 2019

[3] Müller, R., Franke, J., Henrich, D., Kuhlenkötter, B., Raatz, A. u. Verl, A. (Hrsg.): Handbuch Mensch-Roboter-Kollaboration. München: Hanser 2019

[4] 2017. *DIN ISO/TS 15066:2017-04, Roboter und Robotikgeräte_- Kollaborierende Roboter (ISO/TS_15066:2016)*

[5] iff Fraunhofer: Optische Arbeitsraumüberwachung. ECHORD-EXECELL, 2016. https://www.iff.fraunhofer.de/de/geschaeftsbereiche/robotersysteme/echord-execell.html

[6] iff Fraunhofer: Sichere Arbeitsraumüberwachung. Projekts: ViERforES, Magdeburg 2016. https://www.iff.fraunhofer.de/de/geschaeftsbereiche/robotersysteme/vierfores.html

[7] Sebastian Reitelshöfer: MRK - Flexible Zusammenarbeit oder schutzzaunlose Vollautomatisierung. MRK Fachtagung Nürnberg. Nürnberg 2019

[8] Laible, M.: Mechanische Größen, elektrisch gemessen. Grundlagen und Beispiele zur technischen Ausführung ; mit 5 Tabellen. Kontakt & Studium, Bd. 45. Renningen: Expert-Verl. 2009

[9] 2012. *DIN EN ISO 10218-1 : 2012-01, Industrieroboter_- Sicherheitsanforderungen_- Teil_1: Roboter (ISO_10218-1:2011); Deutsche Fassung EN_ISO_10218-1:2011. 2012-01, Industrieroboter_- Sicherheitsanforderungen_- Teil_1: Roboter (ISO_10218-1:2011); Deutsche Fassung EN_ISO_10218-1:2011*

[10] 2012. *DIN EN ISO 10218-2 : 2012-06, Industrieroboter_- Sicherheitsanforderungen_- Teil_2: Robotersysteme und Integration (ISO_10218-2:2011); Deutsche Fassung EN_ISO_10218-2:2011. 2012-06, Industrieroboter_- Sicherheitsanforderungen_- Teil_2: Robotersysteme und Integration (ISO_10218-2:2011); Deutsche Fassung EN_ISO_10218-2:2011*

[11] 2016. *DIN EN ISO 13849-1 : 2016-06, Sicherheit von Maschinen_- Sicherheitsbezogene Teile von Steuerungen_- Teil_1: Allgemeine Gestaltungsleitsätze (ISO_13849-1:2015); Deutsche Fassung EN_ISO_13849-1:2015. 2016-06, Sicherheit von Maschinen_- Sicherheitsbezogene Teile von Steuerungen_- Teil_1: Allgemeine Gestaltungsleitsätze (ISO_13849-1:2015); Deutsche Fassung EN_ISO_13849-1:2015*

[12] Endsley, M. R. u. Garland, D. J. (Hrsg.): Situation awareness analysis and measurement. Boca Raton: CRC Press 2008

[13] Kaber, D. B. u. Endsley, M. R.: The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. Theoretical Issues in Ergonomics Science 5 (2004) 2, S. 113–153

[14] Endsley, M. R. u. Kaber, D. B.: Level of automation effects on performance, situation awareness and workload in a dynamic control task. Ergonomics 42 (1999) 3, S. 462–492

[15] Sarter, N. B. u. Woods, D. D.: How in the World Did We Ever Get into That Mode? Mode Error and Awareness in Supervisory Control. Human Factors: The Journal of the Human Factors and Ergonomics Society 37 (1995) 1, S. 5–19

[16] König, H.: Protocol Engineering. Prinzip, Beschreibung und Entwicklung von Kommunikationsprotokollen. Leitfäden der Informatik. Wiesbaden: Vieweg+Teubner Verlag 2003

[17] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, Jesus Alonso-Zarate: A Survey on Application Layer Protocols for IoT. transaction on IoT and Cloud Computing, 2015. https://de.scribd.com/document/356485459/A-Survey-on-Application-Layer-Protocols-for-IoT, abgerufen am: 27.05.2019

[18] Urs Hunkeler & Hong Linh Truong, Andy Stanford-Clark: MQTT-S – A Publish/Subscribe Protocol ForWireless Sensor Networks. https://sites.cs.ucsb.edu/~rich/class/cs293b-cloud/papers/mqtt-s.pdf

[19] A. Luoto and K. Systä: IoT Application Deployment Using Request-Response Pattern with MQTT. In: Garrigós, I. u. Wimmer, M. (Hrsg.): Current trends in web engineering. ICWE 2017 International Workshops, Liquid Multi-Device Software and EnWoT, practi-O-web, NLPIT, SoWeMine ; Rome, Italy, June 5-8, 2017 ; revised selected papers. Lecture Notes in Computer Science, Bd. 10544. Cham: Springer 2018

[20] Garrigós, I. u. Wimmer, M. (Hrsg.): Current trends in web engineering. ICWE 2017 International Workshops, Liquid Multi-Device Software and EnWoT, practi-O-web, NLPIT, SoWeMine ; Rome, Italy, June 5-8, 2017 ; revised selected papers. Lecture Notes in Computer Science, Bd. 10544. Cham: Springer 2018

[21] Dazhi Chen and Pramod K. Varshney: QoS Support in Wireless Sensor Networks: ASurvey. in International Conference on Wireless Networks, Department of EECS, Syracuse University 2004

[22] Dominik Obermaier * / Sebastian Gerstl: Was ist MQTT?, 19.06.18. https://www.embedded-software-engineering.de/was-ist-mqtt-a-725485/

[23] Thomas J. Burke: OPC Unified Architecture Wegbereiter der 4. industriellen (R)Evolution. https://www.iosb.fraunhofer.de/servlet/is/21752/OPC-UA-Wegbereiter-der-I40.pdf?command=downloadContent&filename=OPC-UA-Wegbereiter-der-I40.pdf

[24] OpenDDS: DDS Overview. Overview of DDS - DCPS. http://opendds.org/about/dds_overview.html

[25] Bartmann, E.: IoT-Programmierung mit Node-RED. Aachen: Elektor 2018

[26] 2016. *DIN EN ISO 13850:2016-05, Sicherheit von Maschinen_- Not-Halt-Funktion_- Gestaltungsleitsätze (ISO_13850:2015); Deutsche Fassung EN_ISO_13850:2015*

[27] 2011. *DIN EN ISO 12100:2011-03, Sicherheit von Maschinen_- Allgemeine Gestaltungsleitsätze_- Risikobeurteilung und Risikominderung (ISO_12100:2010); Deutsche Fassung EN_ISO_12100:2010*
[28] ITEM: Datenblatt Motor SE 60-150-3-60-R. https://product.item24.de/produkte/produktkatalog/produktdetails/products/motoren-ohne-bremse-1001374081/motor-se-60-150-3-60-r-weissaluminium-aehnlich-ral-9006-66602/
[29] ITEM: Produkthandbuch item Servo Positioning Controller C3.05. http://downloads.item-motionsoft.de/Documentation/C-Series/Manuals/C3-Series%20Handbuch%20DE.pdf
[30] Robert Bosch Manufacturing Solutions GmbH: APAS assistant Produktübersicht, 2018. https://www.bosch-apas.com/media/de/apas/downloads/apas_assistant_technische_unterlage_de.pdf
[31] Prof. Dr.-Ing. Rainer Müller: Montagesystemtechnik. Vorlesung 8, UDS. Saarbrücken 2019

Contact Information
ZeMA – Zentrum für Mechatronik und Automatisierungstechnik gemeinnützige GmbH und
Universität des Saarlandes, Lehrstuhl für Montagesysteme
Prof. Dr.-Ing. Rainer Müller
Fabian Adler M.Sc.
f.adler @zema.de
Phone +49 (0) 68185787-522
Dipl.-Ing. Daniela Schmidt.
d.schmidt@zema.de
Phone +49 (0) 68185787-544

Acknowledgments
The results are part of the project “COTEMACO- Increased NWE COmpetitiveness Through Efficient MAn & Machine COllaboration”, funded by the European Union’s Horizon 2020 research and innovation program at ZeMA. The programme guides SMEs through the process of adopting collaborative robotic (cobot) and shop floor digitalisation technologies, from the exploration of technological opportunities to the detailed definition of a business plan.

Study of the haptic interface of a collaborative parallel robot

Zeya WANG, Jean-François ANTOINE and Gabriel ABBA

Department of Design Manufacturing Control
Université de Lorraine, 1 route d'Ars Laquenexy 57078 Metz Cedex, France
ARTS ET METIERS ParisTech, 4 Rue Augustin Fresnel, 57078 Metz, France
Corresponding author: zeya.wang4@etu.univ-lorraine.fr

***Abstract* - This paper describes all the work carried out for the study of the haptic interface for a collaborative robot. The proposed structure connects two parallel mechanisms in series: first a modified Delta and second a hybrid Agile Eye mechanism. The global structure is taken to achieve 6 degrees of freedom of movement. By doing simulations, the main geometrical parameters of the interface were determined. Then, a 3D model of the interface was designed in order to implement the interface through 3D printing.**

Keywords: Collaborative robot, 6 degrees haptic interface, parallel mechanism, control of cable robots.

I. INTRODUCTION

For the development of a collaborative cable-driven parallel robot, a four cable-driven platform has been designed and is manufactured.

In order to realize the human control over this collaborative robot, a haptic interface is necessary. The operator moves the handle of the interface to order the movement of the robot, this displacement should be measured by the sensors embedded in the interface in order to use it as the input for the speed control of the robot.

Research on haptic interfaces began in the 1950s when a master-slave system was proposed by Goertz [1]. Since then, many haptic interfaces have been developed. Phantom [2] developed by Massie and Salisbury is a 3-DOF (degrees of freedom) haptic device.

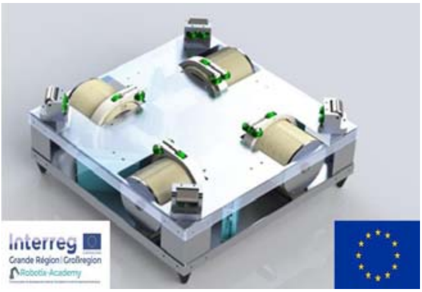


Fig. 1 The future cable-driven platform

II. CHOICE OF THE STRUCTURE

The design requirements for a mechanism of this targeted interface are listed as follows:

1. 6-DOF movement capacity: the interface must be able to move in 6 DOF corresponding to the movement of the controlled robot.
2. Compact design: The overall structure of the interface must be compact to fit the existing system.
3. Large workspace: if its runs of each DOF are large, it makes a larger displacement of the sensor axes, which will limit the necessary resolution of the sensor.

To meet requirements 1 and 2, the parallel mechanism will be a good candidate. However, requirement 3 will not be satisfied by a parallel mechanism only, since the rotation space of the parallel mechanism is generally very limited. So we solve the problem by applying two parallel mechanisms connected in series, separately for the translation movement and the rotational movement.

For the two parallel mechanisms, our idea is to fix the 3-DOF rotation mechanism on the moving platform of the 3-DOF translation mechanism to achieve the 6 DOF. Thus, it is necessary to take the right structures for these two mechanisms.

A. The structure for the 3-DOF translation mechanism

We are inspired by the structure of the Delta robot which is able to make the fast translation movement in the 3 DOF.

The Delta robot was invented by Clavel [3], it has excellent performance: high work speed and accuracy, low power consumption.

As the traditional Delta uses ball joints to connect the forearm to the arm at one end and the moving platform at the other, its workspace is relatively small. We decide to take a modified Delta robot structure proposed by Tsai [4].

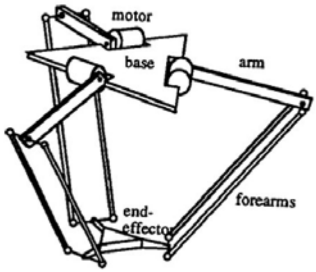


Fig. 2 Robot Delta [5]

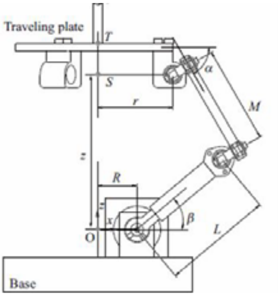


Fig. 3 Kinematic chain of “modified Delta” [4]

The modified mechanism uses revolute joints to replace all the ball joints in the Delta: two revolute joints whose axes of rotation are perpendicular to each other to replace one ball joint. We call this mechanism a "modified Delta". With this modification of the mechanism, we have a larger moving range for the joints respectively between arm and forearms and between forearms and moving platform.

B. The structure for the 3-DOF rotation mechanism

For the mechanism that realizes the 3 rotating DOF, we are inspired by the structure called "Agile Eye" [6].

Agile Eye is a 3-RRR spherical parallel mechanism. Due to its low inertia in motion and its inherent stiffness, the mechanism is able to reach very high angular speed.

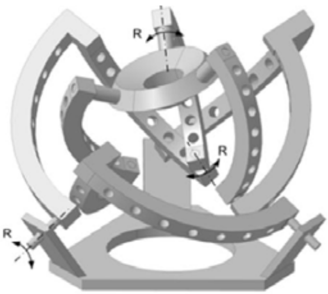


Fig. 4 Agile Eye [7]

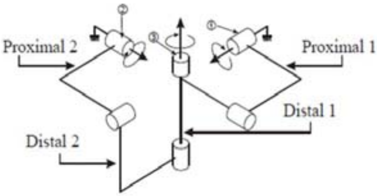


Fig. 5 Kinematic diagram of “hybrid Agile Eye”

However, the Agile Eye has a small workspace in rotation around the Z axis (yaw), its mobility is limited by singular configurations and internal collisions. Therefore, we decide to take a simpler mechanism whose modification is based on the Agile Eye.

This mechanism is composed of 2-DOF Agile Eye and a joint operated by rotation of the handle, it provides unlimited movements for the yaw rotation. We call it “hybrid Agile Eye” [8].

III. SIMULATION

For determining the geometric parameters of the system and its balance position, we have simulated it under Matlab.

A. The geometric model

In order to simulate the system, it is necessary to create the geometric model of the mechanism.

As the 3RRR “hybrid Agile Eye” mechanism makes the 3 angles of rotation measured directly, we focus on the geometric model of the "modified Delta".

In the figure 6, we consider the configuration to be ternary symmetry, formed of 3 identical kinematic chains arranged in a period of 120 ° (like Delta in Fig.2).

P: the point at the center of the moving platform;

O: the point at the center of the fixed base;

Ra: distance between the center of the fixed base and the rotation axis of the arm La;

Rb: distance between the center of the moving platform and the rotation axis of the forearm Lb;

La: length of the arm;

Lb: length of the forearm;

αi: angle between the i-th arm (i = 1, 2,3) and the plane of the fixed base.

The manipulation of the operator will cause the translation of the platform, the three rotary encoders (green in Fig.6) are installed on the base to measure the 3 angles αi. With the values of these 3 angles, we must get the exact displacement of the platform in order to control the robot’s movement. So for the geometric model of the mechanism, the inputs are α1, α2 and α3, and the outputs to be calculated are the coordinates of the point P in the fixed reference frame O.

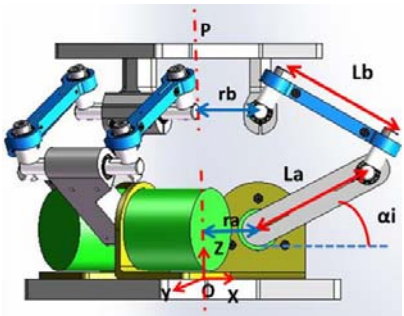


Fig. 6 Geometric parameters of “modified Delta”

```
function [ xyz ] = MGD( a )
lb=60;%bielle liée à la plateforme
la=60; %bielle liée au codeur
r=0;
b1=0;
b2=120;
b3=-120;

D1=-(lb^2)+la^2+r^2+2*r*la*cosd(a(1,1));
D2=-(lb^2)+la^2+r^2+2*r*la*cosd(a(1,2));
D3=-(lb^2)+la^2+r^2+2*r*la*cosd(a(1,3));

E1=2*(r+la*cosd(a(1,1)))*cosd(b1);
E2=2*(r+la*cosd(a(1,2)))*cosd(b2);
E3=2*(r+la*cosd(a(1,3)))*cosd(b3);

F1=2*(r+la*cosd(a(1,1)))*sind(b1);
F2=2*(r+la*cosd(a(1,2)))*sind(b2);
F3=2*(r+la*cosd(a(1,3)))*sind(b3);

G1=-2*la*sind(a(1,1));
G2=-2*la*sind(a(1,2));
G3=-2*la*sind(a(1,3));

H1=E1*G2-E1*G3-E2*G1+E2*G3+E3*G1-E3*G2;
H2=-(E1*F2-E1*F3-E2*F1+E2*F3+E3*F1-E3*F2);
H3=-(E1*D2-E1*D3-E2*D1+E2*D3+E3*D1-E3*D2);
H4=F1*D2-F1*D3-F2*D1+F2*D3+F3*D1-F3*D2;
H5=-(F1*G2-F1*G3-F2*G1+F2*G3+F3*G1-F3*G2);

L=1+(H5^2+H1^2)/H2^2;
M=(2*(H5*H4+H1*H3)/H2^2)-((H5*E1+H1*F1)/H2)-G1;
N=((H4^2+H3^2)/H2^2)-((H4*E1+H3*F1)/H2)+D1;

xyz(1,3)=-(-M-sqrt(M^2-4*L*N))/(2*L);
xyz(1,1)=(-xyz(1,3)*H5/H2)+H4/H2;
xyz(1,2)=(-xyz(1,3)*H1/H2)+H3/H2;

end
```

Fig. 7 Programs for the geometric model

In order to create this geometrical model, we have used the geometric model of Delta robot for reference. Then we adapted it in our case and did the corresponding programming.

B. Determination of the geometric parameters' values

We have to take the good geometric parameters for realizing physically the mechanism. The determination of these parameters is aimed at achieving a compact structure and the suitable workspace of the interface.

To create the 3D workspace of the system, the vector a = [α1 α2 α3] is taken as the input of the geometric model function to calculate the Cartesian coordinates of the moving point P. By using different values of αi within a certain range, all of the corresponding points P can be plotted in the 3D space to simulate the workspace of the mechanism.

At first, the value of r = Ra – Rb and the range of αi are kept constant to show the influence of La and Lb to the workspace. By plotting different workspace with different La and Lb (to avoid the singular points in the workspace, the value of Lb/La should be controlled between 1 and 2 [3]), the combination La=Lb=60mm is chosen because of the regular workspace obtained.

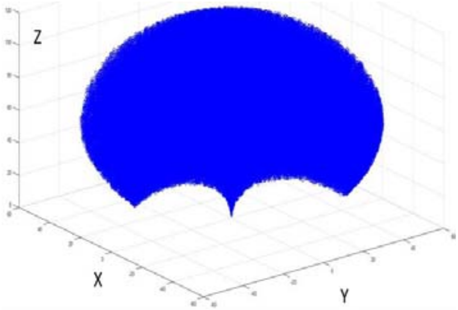


Fig. 8 Workspace for La=Lb=60mm

Similarly, with the La=Lb=60mm and the range of αi unchanged, the value of r is changed to plot different workspace (for the same reason of avoiding singular points, r should be kept positive).

On these 4 figures (Fig. 9) it can be clearly seen that the points are gradually concentrated in the lower part when the value of r increases. In addition, with the increase of r, a "breaking" zone expands in the workspace. Because we want the points to be distributed evenly and the workspace to be continuous, obviously the situation of r = 0 meets best the requirements.

In the same way, various ranges of αi are tested to get a reasonable workspace. To avoid the singular position, the maximum of αi cannot exceed 90°. The workspace is plotted by using αi from -30 to 90.

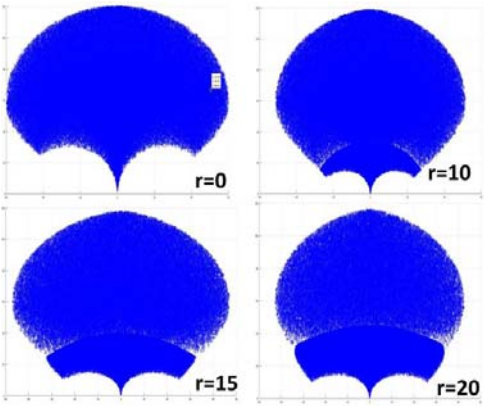


Fig. 9 Workspace in YZ plane with different values of r

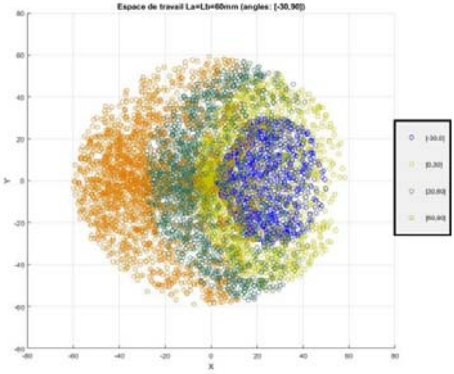


Fig. 10 Workspace for different ranges of αi

It's observed that the range [-30, 0] (blue in Fig.10) generates a space which is entirely covered by other ranges, which means the points coincide with the others. If the same output corresponds to more than one input, the robot control will be unrealizable. Therefore, after rejecting the negative values, the range [0, 90 °] is chosen for αi.

C. Determination of the balance position

The balance position is the origin of the platform's movement, the manipulation of the operator always starts at this position. With all the parameters of the geometric model determined, it is necessary to choose this position for the physical system. In the diagram of the workspace, this position should be placed where concentrates the most points, allowing to have the best sensitivity of the sensors. It means that we should detect the smallest displacement as much as possible at the beginning of the operator's manipulation for a good performance of the control system.

By observing the scatter plot in the XY plane, it can be easily discovered that the points are concentrated in the middle of the circle, that is to say the position x = y = 0. Since it is not easy to identify at which position Z concentrates the most points, we extracted different layers (parallel to the XY plane) of points in the same scatter plot in order to count by programs the number of points in every layer. We have found that the layer $z \in [55, 65]$ contains the most points, and in this layer, it is around z = 60 where the points are most concentrated. So the point (0 0 60) is taken as the balance position for the "modified Delta" mechanism in the interface.

IV. 3D MODEL

A. The design for "modified Delta"

The 3 same kinematic chains assure the dynamic performance of the mechanism. The arm drives directly the rotation of the encoder axis to transfer directly the changes of the angle αi. The two forearms, rotary axis with the 4 shoulder screws form a moving parallelogram which ensures the pure translation of the platform.

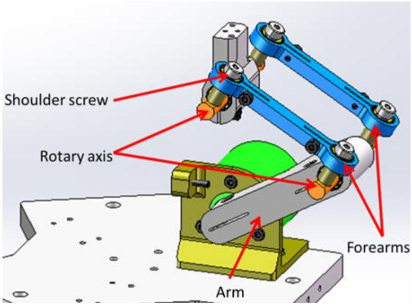


Fig. 11 The kinematic chain design for "modified Delta"

Two shoulder screws are bolted on each end of one rotary axis; the two forearms are connected to the two shoulder screws by revolute joints, their axes of rotation remain perpendicular to that of the rotary axes. All revolute joints are made by ball bearings for higher precision and less friction.

The 3 kinematic chains connect the base and the moving platform to realize the complete mechanism, 3 springs are taken to make the mechanism automatically return to its balance position.

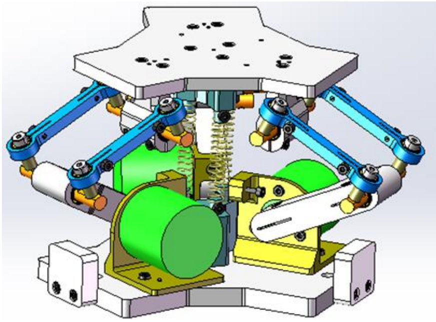


Fig. 12 3D model of "modified Delta"

B. The design for "hybrid Agile Eye"

In the Fig.13, part 1 rotates around the Y axis and drives the corresponding encoder. Part 3 is installed in the same way, it rotates around the X axis.

The part 2 and the part 5 are fastened together. By revolute joints, part 2 rotates relative to 3 around the Y axis. The lower part of 5 is a cylindrical structure that allows 5 to slide in the semicircular groove of the part 1. Consequently, the part 1 drives 5 to rotate around the X axis, and the part 2 driven by 3 rotates around the Y axis, so the assembly of 2 and 5 makes a 2-DOF rotation.

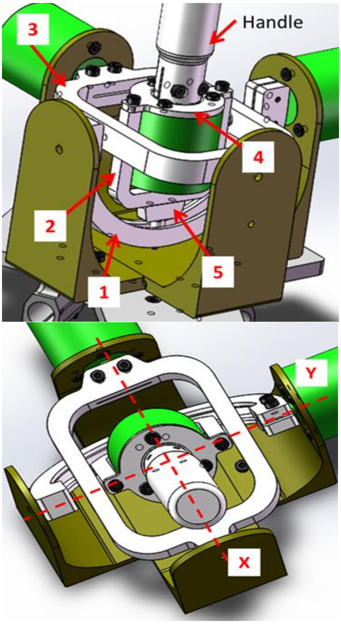


Fig. 13 3D model of "hybrid Agile Eye"

The handle is connected to the axis of the encoder who measures the rotation around the Z axis. The part 4 fixes this encoder on the part 2; therefore the handle is able to realize a 3-DOF rotation.

C. *3D model of the interface*

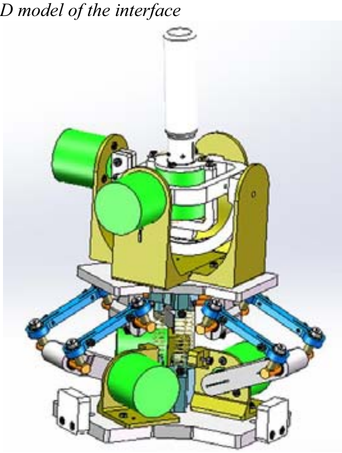


Fig. 14 3D model of the interface

The "hybrid Agile Eye" is fixed on the moving platform of the "modified Delta" to make the complete system of the interface.

V. Conclusions

We have presented the study of a haptic interface for the collaborative cable-driven robot. This study is mainly on the 3 following aspects:

1. The choice of the structure for the interface: connect "hybrid Agile Eye" to the "modified Delta" to constitute the interface for a 6-DOF motion.
2. Simulation of the mechanism "modified Delta": we have determined the geometrical parameters of the mechanism to meet the performance requirements.
3. The design of the interface: the 3D model of the interface is made and this model is ready for 3D printing to test its functionality.

Future research will be directed to test and improve the interface by connecting it to the control system of the robot.

Acknowledgment

We would like to thank the Robotix-Academy, contract number N°002-4-09-001 for funding this work as a part of the project funded by INTERREG V-A Grande Region program.

References

[1] Goertz R. C, "Fundamentals of general-purpose remote manipulators". Nucleonics, Vol. 10, No. 11, pp. 36-42, 1962.

[2] Massie T. H. and J. K. Salisbury, "The PHANToM haptic interfaces: A device for probing virtual objects". ASME Int. Mechanical Engineering Exposition and Congress, pp. 295-302, Chicago, USA, 1994.

[3] Clavel R, "DELTA, a fast robot with parallel geometry". In: 18. Int. Symposium on Industrial Robots, pp 91–100J, Lausanne, Switzerland, 1988.

[4] Tsai LW, "Multi-degree-of-freedom mechanisms for machine tools and like". U.S. Patent No. 5656905, 1995.

[5] Bentaleb, K. Zennadi, and M. A. Belouchrani, "Etalonnage Géométrique du Robot Parallèle "Delta"-Méthodes et Applications". Conférence sur les Sciences de la Mécanique, Oum El Bouaghi, Algeria, 2006.

[6] Gosselin C., E. S. Pierre, and M. Gagne, "On the development of the agile eye" Rob. Aut. Mag., IEEE, vol. 3, no. 4, pp. 29–37, 1996.

[7] Callegari Massimo, Carbonari Luca, Palmieri Giacomo, Palpacelli Matteo, "Parallel Wrists for Enhancing Grasping Performance". Grasping in Robotics, pp 189-219, 2013.

[8] Chablat, Damien and Wenger, Philippe, "A Six Degree-of-Freedom Haptic Device Based on the Orthoglide and a Hybrid Agile Eye". IDETC, Philadelphie, USA, 2007.

Planning for Human-Robot Collaboration using Markov Decision Processes

Khansa Rekik*†, Prof. Jörg Hoffmann†, Prof. Dr.-Ing. Rainer Müller*, Marcel Steinmetz†
and Dr.-Ing. Matthias Vette-Steinkamp*
*ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gemeinnützige GmbH, Saarbrücken, Germany
†Saarland University, Saarbrücken, Germany

***Abstract*—In semi-automated industrial applications, interaction between humans and robots is essential. Such interactions require some level of mutual awareness and coordination. Precisely, while interacting with humans, robots need to be aware of their state and possible future actions in order to collaborate with them and help achieve their goal more efficiently. The focus of this work is the problem of planning for Human-Robot collaboration. First, the robot actions and their dependency on the human's activity are modeled as a Markov Decision Process (MDP). Second, the instances of the model are solved using an off-the-shelf planner. Through analysis of the solutions to the model, results highlight the influence of experimental parameters such as the size of the task and the horizon on the efficiency of the solver. Finally, the deployment of the MDP on a use-case assembly process scenario inspired from an aerospace manufacturing industry is discussed.**

***Index Terms*—Human-Robot collaboration, AI planning, Markov Decision Processes**

I. Introduction

Large scale industries are constantly seeking new technologies to increase overall efficiency in the production line. To that end, attempts are continuously being made to automate production processes. One such an attempt is introducing robots to automate repetitive and non-ergonomic tasks. However, introducing fully automated solutions can, in some cases, be inefficient compared to manual or semi-automated solutions. In such cases, one can use hybrid solutions involving both humans and robots. For instance, one can use Human-Robot Collaboration (HRC) based solutions. HRC solutions embody challenges present in division of responsibilities, action disambiguation etc [1].

Identifying the intended action by the human is important for the robot to disambiguate what the former is trying to achieve, hence to flexibly support him. However this disambiguation step can be significantly simplified in scenarios where the worker workflow is known to the robot, for instance, the ordering of steps in the workflow, the probabilities of transitioning from one activity to the other, etc.

In the context of establishing a human aware collaboration, the robot task can be viewed as "a Artificial Intelligence planning task"; it is the task of selecting a goal-leading course of actions based on observations and a model of the world and action behaviour [2], [3]. In the aforementioned scenarios' category the robot has to transition across states through submitting actions until reaching the common goal. A state is affected by multiple factors namely the observations about the human worker and model of the human's workflow, the model of the environment, as well as the model of its own actions. In other terms, based on the human work-flow as well as any environmental factor that affect the decision making process, the robot can infer which actions to pursue in order to achieve the shared task.

Such dynamic can be modeled with different planning frameworks (deterministic planning, stochastic planning etc.). The challenge is to find trade-offs between accurately modeling the problem and the planning complexity [4]. Planning, as presented in [2], [4], assumes certainty about the initial state and the actions' effects. This assumption is relaxed for probabilistic planning frameworks, such as *Markov Decision Processes (MDP) planning*, where transitions are probabilistic. According to [5], one can use MDPs for simulating a rational human's acting towards a certain intention. The result of solving these models would be integrated in the robot's model and used as a reference for the observations gathered by the robot. Similarly, the work-flow of both agents can be modeled by means of MDP [6].

As an initial case study, we consider the riveting process in the aircraft assembly for a use-case scenario as it is one important use case among many others. Due to its complexity, this process can be semi-automatized in a way where the human can perform the hammering while a robot counter-holds from the opposite side. This scenario is modeled as an MDP, since MDPs represent a good compromise between accuracy and complexity. On one hand, MDPs allow to model stochastic workflows. on the other hand, they allow assuming that the human workflow state is known so that no reasoning about sensing and its implications is required. Note that the methods we use are generic, thus potentially reusable for more complex scenarios.

This work is divided into five main parts. First we introduce a use-case scenario from the aircraft industry. Second we give an overview about the MDP planning background. In the third part we present the modeling of the use-case process first as an MDP followed by results of the efficiency tests performed using an off-the-shelf solver. Finally, we discuss the deployment of the model to a real demonstration.

This research was supported by ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gemeinnützige GmbH through the Robotix-Academy project.

II. USE-CASE SCENARIO: RIVETING PROCESS IN AIRCRAFT ASSEMBLY

The use-case scenario implemented in this work is the riveting process used for aircraft assembly. The body of an aircraft consists of different segments, which are adjusted with respect to each other employing handling systems and joined manually with rivets. Riveting requires two workers; one for counter holding and one for hammering. For ergonomical purposes and to improve production conditions and thus efficiency, we aim at contributing to the automatization of this process.

Previous work on this project, described in [7], replaced one of the workers with a robot to perform the counter-holder role during the riveting process. However, the remaining worker has to send orders to the robot of the actions to be performed through an interactive device. A more optimal solution would be like follows; the robot observes the worker, infers the current activity and plans actions to help reach the common goal.

We call *human workflow* the set of activities the worker can perform and the possible transitions between them. We restrict the human possible activities to three: {waiting, riveting, having a break}. Note that transitions are non-deterministic.

Once the robot has inferred the activity of the worker, it can decide on what actions to perform avoiding making the worker wait. The process is divided into three main parts. First the riveting points have to be scanned to determine their absolute positions with respect to the robot's coordinate system. Second, the riveting itself is performed i.e. the robot navigates between point and counter-holds while the human is hammering. Finally, points are inspected to verify their compliance with quality standards.

In the frame of this work, we assume that the scanning step has been accomplished, we focus on riveting and inspecting.

III. BACKGROUND: MDP PLANNING

We consider the formulation of [8] for a Markov Decision Process (MDP), it is represented as a tuple $\mathcal{P} = \langle S, A, T, R \rangle$ where:

- S stands for a set of state variables
- A represents a set of the agent's actions, in analogy with classical planning, one can extend this MDP formulation by adding effects $(eff(a))$ and preconditions $(pre(a))$ to actions
- T is a function of the transition probabilities $P_a(s'|s)$ for $a \in A$, $s', s \in S$, if one is in state s and performs and action a, one gets to a state s' with probability $P_a(s'|s)$,
- $\mathcal{R}$ is a reward function for executing an action a in state s

We define an MDP policy π as a function $\pi : S \rightarrow A$ that maps actions to MDP states, a the policy that maximizes the long-term expected reward is an optimal policy π^*. A reward can be discounted by means of a discount factor γ in $[0, 1]$. The role of the discount factor is to make earlier rewards advantageous. For instance, a reward n steps away is discounted by γ^n.

An MDP Horizon H is the number of actions the system will take during its life time [5]. It gives a foresight in several time steps in the future. Thus one can get a horizon-optimal policy i.e a policy that, for every initial state s_0, results in the maximal expected reward from times 0 to the size of the horizon. Note that MDPs assume the *Markov Property* [9], more explicitly, the effects of an action a taken in a state s_t do not depend on the prior history, they only depend on that state (s). In addition, in a markov decision process the dynamics of the environment are fully observable. In other terms, the state s' resulting from executing a is fully known by the system.

One can evaluate a policy thanks to the value function V. This function calculates the long-term expected reward of a policy π, it can be computed using the *Bellman equation*:

$$V_t^\pi(s) = R(s, \pi_t(s)) + \sum_{s' \in S} T(s, \pi_t(s), s') \cdot \gamma \cdot V_{t-1}^\pi(s') \quad (1)$$

Thus one can compute the optimal policy π^* like follows:

$$\pi^* = \arg\max_{a \in A} \left[R(s, \pi(s)) + \sum_{s' \in S} T(s, \pi_t(s), s') \cdot \gamma \cdot V^*(s') \right] \quad (2)$$

and:

$$V^*(s) = \max_{a \in A} \left[R(s, \pi_t(s)) + \sum_{s' \in S} T(s, \pi_t(s), s') \cdot \gamma \cdot V^*(s') \right] \quad (3)$$

Algorithms for solving MDPs: There are several ways to optimally solve MDPs namely Value-iteration and Policy-iteration algorithms. In value-iteration algorithms, one keeps improving the value function at each iteration until the value-function converges. Whereas in Policy-iteration algorithms one re-defines the policy at each step and computes the value according to the new policy until it converges. Another algorithm for planning under uncertainty is the UCT algorithm [10]. UCT is one of the representatives of Monte-Carlo Tree Search algorithms on which the planner PROST is based. According to [11], PROST implements techniques on top of the UCT skeleton to show its applicability to domain independent probabilistic planning and to adapt it to the stochastic planning context. One such a context is creating strongly connected search space. Moreover, unlike UCT, PROST detects reward locks which makes it more efficient in domains presenting such locks. Furthermore, PROST performs a Q-value initialization step which prevents initial random walks in the search space. The input language used by PROST is the Relational Dynamic Influence Diagram Language (RDDL) [12]. Conventionally, actions in MDPs do not explicitly have preconditions and effects. However, planning domain languages like PPDDL and RDDL specify these actions in a factored manner analogous to classical planning. Thus actions can have preconditions and effects. Winner of the IPPC 2011 and IPPC 2014 competitions, PROST is mainly efficient for MDP planning. To this end, we will use RDDL for the implementation of the MDP model of our use-case task and PROST for running the experiments.

IV. THE PROPOSED MODEL FOR THE RIVETING PROCESS

In this section, the MDP model of the aforementioned riveting process is presented. The human's work-flow is embedded in the robot's model. More precisely, the activity of the human is represented as a state variable that evolves in a probabilistic fashion. The state space is composed of the robot's state variables S_R, the human's state variables S_R and the state variables relative to the riveting points S_{rp} :

$$S = S_R \times S_H \times S_{rp}$$

The robot variables contain the robot position, which is a number between one and the total number of riveting points, and weather it has stopped i.e the task has been finished. Moreover, the rivets variables contain the state of the riveting point: scanned, riveted or inspected. Also if a point has been inspected or riveted in the last time step. The human variables contain the human position and current activity: waiting, riveting or having-a-break.

Analogically to classical planning, in this model actions have preconditions and effects. The possible actions are move next or previous, counter-hold, inspect and stop. Those actions are parametrized by the considered riveting point. Preconditions restrict the applicability of the actions, they can depend on the human state variables. For instance, counter-holding is only possible if the human is waiting in the same position as the robot, this point should be scanned and not previously riveted, also the robot should not be "stopped". Once this action succeeds the state of the point switches from scanned to riveted with a certain probability.

The transition function $P_a(s'|s)$ represents the uncertainty of the output of the action performed by the robot, as well as the uncertainty entailed by the change of the activity of the human.

$$Pr_a(s'|s) = Pr(s_{t+1} = s'|s_t = s, a_t = a) \tag{4}$$

In what follows, we divide a state s into its three different components s_h, s_R and s_{rp} for respectively states variables relatives to the human, the robot and the riveting points. The transitions relative to the position of the robot are not probabilistic as they depend only on the success of the move actions which are chosen to be deterministic. They are independent of the human and the rivets' states. Note that this does not apply for cases where a move-next is performed at the last point or if a move-previous is performed at the first point.

The transitions of the state variable relative to the riveting points are probabilistic. Their transition probabilities depend on the success probabilities of either the counter-hold action and the transition probabilities of the human state or on the success probability of the inspect action.

Analogically the states relative to the human activity evolve in a probabilistic scheme. The model of the human work-flow can be seen as a sub-model of the MDP of the robot where transitions are also probabilistic. Given that the human is more likely to transition from *"watiting"* to *"riveting"* and vice-versa, it is less likely that he takes a break very often and thus the probability of *"having a break"* is lower. Note that transitioning from a *"having a break"* to *"riveting"* is not possible. In other terms, being in break the human can only go to a waiting state. The model of the human, although independent of the robot actions, influences the evolution of the rest of the state variables as well as the applicability of some actions e.g counter-hold.

The reward function is defined in a way that boosts the human workers comfort and penalizes encumbering her plan execution. As described in the following function a positive reward is assigned each time a new point, that has not been riveted previously, is riveted or a point, that has not been inspected, is inspected. A negative reward is however assigned if the robot keeps the human worker waiting or if none of the aforementioned conditions are satisfied.

V. EXPERIMENTS AND PRELIMINARY RESULTS

The MDP model is implemented using RDDL. The flexibility of the language and the broad range of modeling possibilities it offers allows capture the real world setting of the riveting task. However, this flexibility is constrained by the solver that is used to run models encoded in RDDL i.e. PROST. As the latter does not support all possibilities offered by RDDL in terms of modeling, hence, assumptions and simplifications need to be made while modeling the task.

One such a simplification is downgrading the models from their factored form to a ground for in which each factorized variable with a parameter x is transformed to set a variable for which each variable is an instantiation of x.

The results, calculated considering an action time limit of 0.5 s and a total time limit of 60 min, and showed in Fig.1, indicate that, PROST can time out during its heavy parsing phase. To that end, the parsing method has been modified. This has resulted in a significant improvement in performance as shown in the figure.

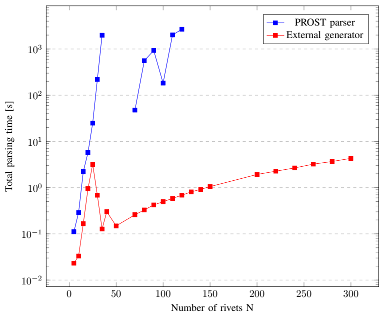


Fig. 1. Comparison of the parsing time between the internal parser of PROST and an external parser given a horizon size H equal to the size of the MDP model N.

The blue curve, representing the parsing time of PROST's conventional parser, shows a variation in the time needed for

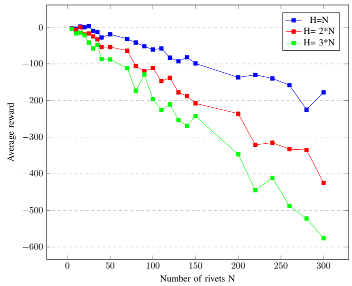


Fig. 2. Change in the average reward for different MDP model sizes N in accordance to change in the horizon size H

parsing with respect to the size of the task i.e. the number of rivets. The whole experiment frequently times out during the parsing namely for tasks with sizes between 40 and 60, and above 120.

As the reward functions is parameter that drives the MDP model, we evaluate the performance of the use-case models solved by PROST in terms of average rewards. In Fig.2, we show the reward values for different horizon sizes with respect to the number of rivets. The rewards have negative values as each action that does not result in a new riveted or inspected point is penalized. In this analysis, it was shown that the solver can solve models with sizes up to 300 riveting points within the time limits imposed. Furthermore, the larger the horizon is, the lower the reward values are for a fixed number of rivets. This is due to the accumulated uncertainty. Note that, having a lower reward for a larger horizon does not presume that a restricted horizon is better. The horizon is not only the foresight but also the number or actions to be performed. More precisely, for a 5 rivets task, if $H = N$ then the process would stop after submitting 5 actions, whereas with $H = 3 * N$ the process would stop after 15 actions. As a result, reward values are lower for larger horizons for there is room for submitting more actions, accumulating more uncertainties and getting more penalized.

VI. Deployment to the Use-Case

The aforementioned model is deployed to a demonstrator of the use-case scenario. The demonstrator consists of the a Universal Robot 10, a lifting unit enabling the transportation of the robot and the sections to be riveted. The section of the demonstrator contains thousands of rivets, thus having a model with hundreds of rivets it's practical for that it helps limit the number of runs launches.

Using a Markov decision process in a real application requires assuming that sensors use for observations are noise free. For the riveting process scenario sensors are used to update the state variables as actions are being executed. Theoretically each state variable should have a way to be validated through the feedback of the sensors. As discussed previously, there are variables relative to the robot, to the human and to the riveting points. In order to observe those variables three external sensors are used in addition to the robot's internal sensors. Detecting the position of the robot is performed by means of the robot's internal encoder. The external sensors are a safety mat for detecting the presence of the human in the workplace, a force-torque sensor to detect if the human is waiting or riveting and also if the rivet has been successfully riveted and a laser scanner to validate the inspection.

Furthermore, bringing the models to practical use requires replacing the RDDLSim server with the server of the real demonstrator. The latter handles the sensor data as well as the deployment of the actions by the robot.

For the tests run on the robot, it is possible to use an input model describing a task with a maximum size of 240 riveting points. At first, PROST is called once, its input is the initial state of the demonstrator and a horizon size of $3 \times N$ for a model of N points. Once an action is submitted, it is sent to the server. An internal clock of the server waits for six seconds which is the time needed for the longest action to be performed. After this period the server reads the new state s' and sends it to the solver. The stopping criterion is the end of the task which is reached once the state variable R-stopped becomes true. However, as the horizon values considered for testing are relatively small, it is unlikely to finish the task within the horizon.

VII. Discussion

In the simulated context, it was shown that a large horizon allows a higher number of actions to be submitted. However, with large horizons, the short term reward is not as maximized as it can be with a restricted horizon. In other words, the size of the horizon influences the decision making in terms of prioritizing actions that give a higher accumulated reward over several decisions instead of maximizing short term rewards. It is important to mention that a different level of flexibility is gained using planning tools in comparison to rule-based or fixed controllers namely the possibility of looking several steps ahead to gather information useful for making the decision.

In the real context, several test have been performed from which one can only detect salient problems, therefore no actual evaluation has been made so far. A further investigation is considered for future work. Nevertheless, some preliminary conclusions and remarks can be done.

On one hand, through the deployment of the model to the real scenario, multiple flows in translating the behavior established in the simulated environment has been detected. For instance, assuming that sensors are noise free has resulted in problems in detecting the current activity of the human (waiting vs. riveting). More precisely, testing the sensor off load shows already a significant variation in the force values returned although no force is applied. This makes it challenging to opt for a specific threshold.

Moreover, using the external parser alleviated the pre-computational but limits the horizon size which makes it almost not possible to finish the whole process.

On the other hand, In comparison with the semi-automated version of the process where the human needs to use an interactive device and guide the robot, our version is flexible. More precisely, one can adapt the time of the counter-holding based on the pace of the worker (beginner or expert).

VIII. Conclusion

This work tackles Human-Robot Collaboration from an MDP planning perspective for a simple industrial scenario. This application represents a step towards deploying non-deterministic planning tools in real life settings. In our case the human model is known to the robot which facilitates deciding on what actions to perform. Whereas if the human has a wider range of activities that are not necessarily explicitly cited in the robot's MDP, the decision making would be more challenging.

In future work, we intend to expand our results to account for more industrial like settings i.e. more uncertainty about the human activity and environmental state.

Acknowledgment

We thank our colleagues from Loria Nancy in France who provided insight and expertise that greatly assisted the research.

References

[1] B. Hayes and B. Scassellati, "Challenges in shared-environment human-robot collaboration," *learning*, vol. 8, no. 9, 2013.

[2] J. Hoffmann, "Everything you always wanted to know about planning," in *Annual Conference on Artificial Intelligence*. Springer, 2011, pp. 1–13.

[3] J. A. Hendler, A. Tate, and M. Drummond, "AI planning: Systems and techniques," *AI magazine*, vol. 11, no. 2, p. 61, 1990.

[4] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: theory and practice*. Elsevier, 2004.

[5] A.-B. Karami, "Modèles décisionnels d'interaction homme-robot," Ph.D. dissertation, Université de Caen, 2011.

[6] B. Bakker, Z. Zivkovic, and B. Krose, "Hierarchical dynamic programming for robot path planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 2756–2761.

[7] R. Mueller, M. Vette-Steinkamp, A. Kanso, and T. Masiak, "Collaboration in a hybrid team of human and robot for improving working conditions in an aircraft riveting process," SAE Technical Paper, Tech. Rep., 2019.

[8] M. Ramírez and H. Geffner, "Goal recognition over pomdps: Inferring the intention of a pomdp agent," in *Twenty-Second International Joint Conference on Artificial Intelligence IJCAI*, 2011.

[9] L. D. Pitt, "A markov property for gaussian processes with a multidimensional parameter," *Archive for Rational Mechanics and Analysis*, vol. 43, no. 5, pp. 367–391, 1971.

[10] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *European conference on machine learning*. Springer, 2006, pp. 282–293.

[11] T. Keller and P. Eyerich, "Prost: Probabilistic planning based on uct." in *International Conference on Automated Planning and Scheduling ICAPS*, 2012.

[12] S. Sanner, "Relational dynamic influence diagram language (rddl): Language description," *Unpublished ms. Australian National University*, p. 32, 2010.

Identification of Friction Model for a Pneumatic Actuator of Robotic Grinding System

Xing Wen*, Mohamed Didi Chaoui, François Léonard, Vianney Papot, Gabriel Abba

Department of Design Manufacturing Control

Université de Lorraine, 1 Route d'Ars Laquenexy, 57078 Metz Cedex 3, France

Arts et Métiers ParisTech, 4 Rue Augustin Fresnel, 57078 Metz, France

*Email : xing.wen5@etu.univ-lorraine.fr

***Abstract* - This work comes from the development process of a robotic grinding system using a pneumatic actuator, which aims to reduce or eliminate instability during the processing. This paper focuses on the identification of friction model for the pneumatic actuator. After depicting the experimental setup and the test method, the friction-velocity maps are obtained through experiments, the model parameters were identified by nonlinear least-squares (NLS) method. A simplified friction model is proposed based on identified parameters. The friction model of the pneumatic actuator is crucial for control precision and grinding force model identification.**

Keywords: Robotic grinding, Pneumatic actuator, Friction model, Parameter identification

I. INTRODUCTION

The emergence of robotic grinding has significantly saved time in production and reduced the overall cost as well as the health and safety risks associated with the metal dust. The robot characteristics like versatility, flexible workspace and high productivity meet the demand of Industry 4.0. Active force control is indispensable for the material removal process because grinding tasks involve force interactions with the environment. The robot with position control has the risk of generating excessive force due to the positioning error, which can damage the workpiece and the tool. The pneumatic actuator is one of the methods to realize the active force control, which is independent of the robot's position controller.

Recent work has shown an idea that a robotic grinding system using a pneumatic actuator which aims to improve the robot processing quality. The previous work [1] has determined the system configuration, and also proposed a grinding force model as well as a process program including the path planning method. However, the dynamics of the cylinder that determines the grinding force remains to study. A pneumatic actuator shows nonlinear behaviors due to the compressibility of air and complex friction force characteristics. Therefore, the study of the friction is necessary to obtain precise control and to calculate the grinding force.

This paper presents a study of friction behavior of the pneumatic actuator, which is carried out through experimental tests. We have firstly depicted the experimental system and the friction test method. Then we have plotted the steady state friction-velocity maps from the experimental data and analyzed the friction behavior by comparing with the friction model. We finally identified the parameters in the model by nonlinear least-squares (NLS) method.

II. EXPERIMENTAL SETUP

As depicted in Fig. 1, an electro-pneumatic system is set up for the experimental test. The setup is composed of two circuit: a pneumatic circuit for control of the end-effector, and an electrical circuit for the acquisition by sensors.

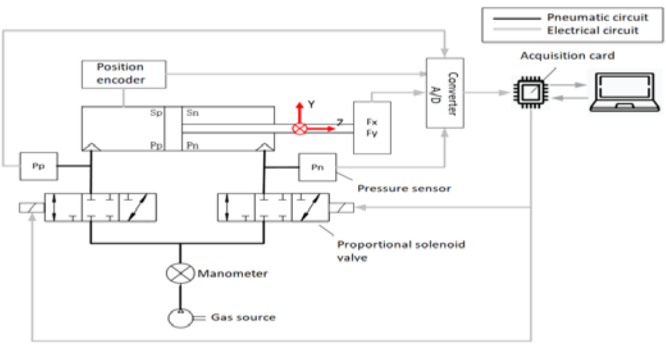


Fig. 1 Schematic diagram of the electro-pneumatic system

The electro-pneumatic system consists of the following elements:

- A gas pipe as the pneumatic power source, the regulator adjusts the supply pressure, which is measured by a manometer placed before the gas flowing to the electric valve.
- Pneumatic double-acting rod cylinder with a position encoder
- Two proportional solenoid valves for flow control with pressure sensor integrated for measuring pressures in two cylinder chambers
- A force sensor measured in the X-Y plane
- Two A/D converters for sensor signal conversion
- An acquisition card
- A laptop for programming, control and data saving

A. *End-effector*

An end-effector (see Fig. 2), is mounted on the robot flange plate through a support. The support integrates several elements of the electro-pneumatic system, including a pneumatic actuator. The piston rod cannot absorb any torque, so an external guide unit is used with the cylinder. A force sensor acting on the XY plane is embedded in a box mounted at the end of the cylinder. An angle grinder "BOSCH GWS 22-230 H" is attached to the box through a mechanical connector. The actuator allows the angle grinder to move along the Z axis.

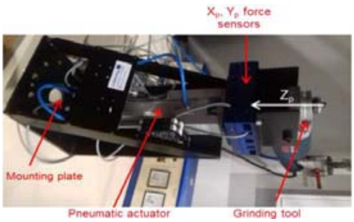


Fig. 2 End-effector for robotic grinding

B. *Sensors*

The system contains 3 types of sensor: pressure sensor, position encoder and force sensor. All sensor references as well as the acquisition elements used are shown in Table 2.

TABLE I ELECTRO-PNEUMATIC EXPERIMENTAL SETUP COMPONENTS

Component	Model	Manufacturer
Electrical valve (pressure sensor)	VPPM-8L-L-1-G14-0L10H-V1P-S1	FESTO
Force sensor	Load Cells 615	Tedea-Huntleigh
Converter	DADE-MVC	FESTO
Strain gauge converter	Z-PC-LINE Z-SG	SENECA
Actuator (Position encoder)	DNCI-40-200-P-A-FENG-MU	FESTO

C. *Acquisition*

All digital measurement signals are converted to analog signals by converter. Then the analog signals are transmitted to the acquisition card, the analog voltage value is between 0-10 V, which is proportional to the actual value of the physical quantities. Bidirectional communication between the acquisition card and the computer is performed by C++ program compiler. The solenoid valves are controlled by computer, but the signal must pass the acquisition card. The recorded measurement data is the sampled discrete signal.

III. MODELING OF THE FRICTION

A suitable friction model can effectively predict the friction behavior and calculate the output force of the cylinder. There are two types of friction model: the static friction model and the dynamic friction model. The static friction model describes friction versus velocity, the shape is relatively simple, but it does not describe dynamic friction effects, while the dynamic friction model describes friction versus velocity and displacement, which is more complex. But this type of model describes more realistically various friction effects.

A. *Stribeck friction model*

The Stribeck friction model (Fig. 3) is widely used in the position control system. It is a static model expressed by [2]:

$$F_f = (F_C + (F_S - F_C)e^{-\left(\frac{|v|}{v_S}\right)^{\delta}})sign(v) + bv \tag{1}$$

where F_S the static dry friction ; F_C the Coulomb friction; v the velocity of the piston ; v_s Stribeck separation speed ; δ a constant, value between 0.5 ~ 2.

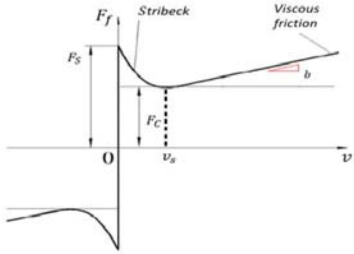


Fig. 3 Stribeck friction model

B. *LuGre friction model*

The LuGre model (see Fig. 4 (a)) [3]takes into account the hysteresis, which avoids the abrupt change of friction across the zone of zero velocity. It is a relatively complete model with good accuracy. The LuGre model considers the contact surface as an elastic bristle with random behavior at the microscopic level (see Fig. 4 (b)), the friction is generated by the deflection of the bristle and it is described by the following equation [4][5]:

$$\begin{cases} F_f = a_0 z + a_1 \dfrac{dz}{dt} + a_2 v \\ \dfrac{dz}{dt} = v - \dfrac{|v|}{g(v)} \\ a_0 g(v) = F_C + (F_S - F_C)e^{-\left(\frac{v}{v_S}\right)^2} \\ F_{SS} = F_C sign(v) + (F_S - F_C)e^{-\left(\frac{v}{v_S}\right)^2} + a_2 v \end{cases} \tag{2}$$

where a_0 rigidity of bristle ; a_1 the damping coefficient of the bristle ; a_2 the viscous coefficient ; z the average bristle deflection ; v the relative sliding velocity ; $g(v)$ describes the Stribeck effect ; F_{SS} the steady state friction.

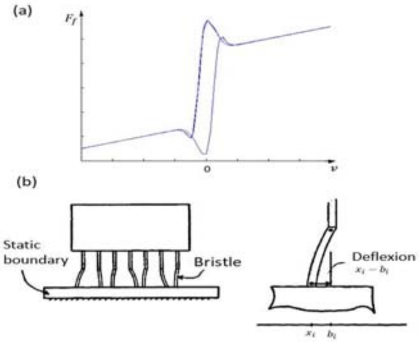


Fig. 4 (a) LuGre friction model [6] (b) Bristle model

There are 6 parameters in total in the model. The identification of the static parameters F_C, F_S, v_S, a_2 is carried out by measuring the relation between the friction and the velocity. The identification of the dynamic parameters a_0, a_1 is complex because of the introduction of an immeasurable quantity z.

C. Summary of the friction model

In general, the dynamic friction model has better continuous characteristics and better description of the nonlinear behavior of the friction, but its parameters are more difficult to identify. In practical engineering applications, the static model is widely used, but the dynamic model meets high precision requirements. The LuGre model is a relatively complete dynamic model, its stable form consist of Stribeck model, and its static parameters can be measured by steady-state tests.

IV. Friction Measurement and Identification

Based on the friction model, the next step is to measure the friction through experimental tests and get the friction-velocity map. In order to identify parameters of the model, NLS method is used.

A. Test method

As depicted in Fig. 5, in the case that the cylinder is placed horizontally, applying Newton's second law, we get the following equation:

$$\begin{cases} F_f = p_1 A_1 - p_2 A_2 - ma \\ a = \dfrac{d^2 x}{dt^2} \end{cases} \tag{3}$$

Therefore the instantaneous dynamic friction of the cylinder can be obtained by measuring the pressure of the two chambers of the cylinder, the velocity and the acceleration of the movement. The mobile mass with grinder needs to measure.

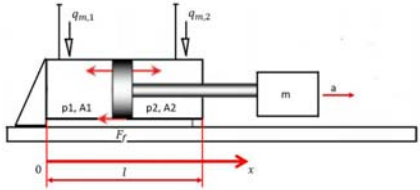


Fig. 5 Schematic diagram of cylinder

The test bench can be divided into passive drive and active drive according to its implementation principle. The active traction is chosen because it allows to measure the friction force directly with the existing system and the operation of the cylinder is closer to the actual conditions, but it is difficult to provide a stable velocity by driving directly with the gas. In addition, it is impossible to exclude the influence of the pressure of the two chambers on the friction.

B. Mass measurement

An electronic balance is used to measure the weight of the piston-rod with the grinder as well as the mounted accessories. Manipulate the attitude of the robot to make the actuator in the vertical direction, no pressure is provided in the chamber, so the mobile part is subject to gravity. The cylinder descends and then climbs vertically with a constant speed thanks to the linear movement of the robot (see Fig. 6). Record the data of the display when the end of the actuator is just touching, and when it is just leaving the electronic balance. The measurements were performed at three different speeds, the mass is determined by calculating its average value (see Table II). The moving mass including piston-rod and grinder is 15.6 kg.

Table II Mass measurement

velocity [mm/s]	weight [kg]						average [kg]
20	17.3			13.8			15.55
2	17.51	13.6	17.58	13.55	17.6	13.6	15.57
6.3	17.4	13.8	17.4	13.9	17.5	13.8	15.63
							15.6

Fig. 6 Mass measurement

C. Friction force measurement

To plot the friction-speed maps, we did a series of cyclic stroke tests: the cylinder starts at a position around a half stroke, the cylinder repeatedly extended and retracted to cycle around the starting position [6]. Velocity and acceleration are respectively the first derivatives and the second derivative of the piston displacement. Since the acquired displacement data is discrete, the derivatives can only be approximated by the finite difference method.

The results presented in Fig. 7 correspond to a particular case: for four cycles around the starting point and without contact with the stops on two-sides. From the diagram we can observe the following phenomenon:

(1) The static dry friction and the Stribeck effect exists just at the starting moment (1-2) according to the first cycle result on the bottom of the figure, then the friction varies nonlinearly with the velocity. The cycle path is 3-4-5-6-7-8-9-3.

(2) There is a hysteresis for the velocity interval [-0.2, 0.2], which means that the Stribeck effect only occurs when the velocity increases. Stable friction can be modeled just by taking into account viscous friction and Coulomb friction.

(3) It is possible to modeling the steady state friction by a piecewise linear function as shown by the yellow curve.

(4) The dry friction is defined in a very low speed zone [-0.02, 0.02].

(5) The other operating cycles with different starting positions show the same behavior.

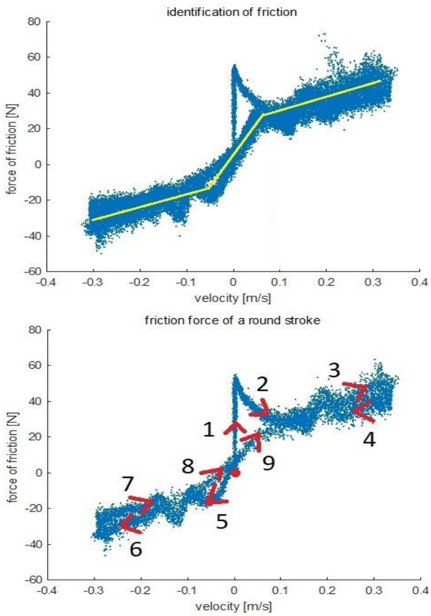


Fig. 7 Friction-velocity map

D. Friction force identification

From (1), the friction force of path 2-3 can be expressed by the following equation:

$$F_f = (F_C + (F_S - F_C)e^{-\left(\frac{|v|}{v_S}\right)})sign(v) + bv \tag{4}$$

Therefore, friction force is a linear function of velocity. There are four parameters to identify, we can solve this problem by NLS method. The objective function is:

$$\min_{\hat{b},\hat{v_S},\hat{F_C},\hat{F_S}} \sum_{i=1}^{n} (F_{fi} - \hat{F}_f(v_i))^2 \tag{5}$$

where F_{fi} the measured value of the discrete point i ; $\hat{F}_f(v_i)$ the theoretical value calculated by (4). Matlab 'curve fitting toolbox' is a handy tool for NLS. Fig. 8 shows the curve fitting results and residual differences.

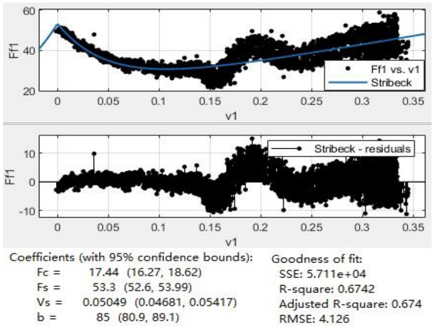


Fig. 8 Model parameters identification

From these parameters, a simplified friction model for steady state is proposed:

$$F_f = \begin{cases} 85v + 17, v \geq 0.05 \\ 425v, -0.02 \leq v < 0.05 \\ 85v - 6.8, v < -0.02 \end{cases} \tag{6}$$

Fig. 9 shows the simulation results of this model.

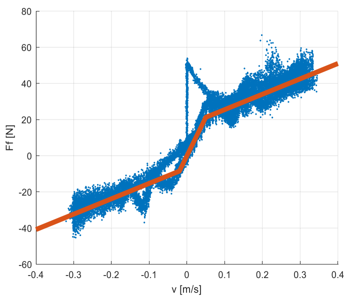


Fig. 9 Simulation of proposed friction model

V. CONCLUSION

In this work, we have proposed a simplified friction model for steady state, which bases on the measurement and NLS identification results. This friction model help us predict the dynamic behavior of the pneumatic actuator. In future work, we will measure the grinding forces and validate the force model that we have proposed. This friction model can be used to calculate the normal grinding force provided by the pneumatic actuator.

ACKNOWLEDGMENT

We would like to thank the Robotix-Academy, contract number N°002-4-09-001 for funding this work as a part of the project funded by INTERREG V-A Grande Region program.

REFERENCES

[1] M. Didi Chaoui, F. Léonard, G. Abba, Path planning for accurate surface shaping using a robotic grinding system, *International Conference on Advances in Mechanical Engineering and Nanotechnology,* 2019.

[2] Y. F. Liu, J. Li, Z. M. Zhang, X. H. Hu, and W. J. Zhang, "Experimental comparison of five friction models on the same test-bed of the micro stick-slip motion system," *Mechanical Sciences,* vol. 6, no. 1, pp. 15-28, 2015.

[3] A. Nasr, C. Mrad, and R. Nasri, "A study of friction vibration absorber: impact of friction modeling on the efficacy of the absorber and friction coefficient optimization," *Mechanics & Industry,* vol. 19, no. 6, p. 602, 2018.

[4] Karl Johan Åström, Carlos Canudas de Wit. Revisiting the LuGre friction model. IEEE Control Systems Magazine, *Institute of Electrical and Electronics Engineers*, 2008, 28 (6), pp.101-114.

[5] X. B. Tran, W. H. Khaing, H. Endo, and H. Yanada, *Effect of friction model on simulation of hydraulic actuator*. 2014, pp. 690-698.

[6] X. Brun, "Linear and nonlinear controls in electropneumatic. Methodologies and applications," Ph.D. dissertation, Laboratoire d'Automatique Industrielle (LAI), Insa Lyon, 1999.

Robotic throwing controller for accelerating a recycling line

Norman Marlier[1,2] , Olivier Brüls[1], Godefroid Dislaire[3] and Gilles Louppe[2]
[1]Department of Aerospace and Mechanical Engineering, University of Liège, Belgium
[2]Department of Electrical Engineering and Computer Science, University of Liège, Belgium
[3]Department of Architecture, Geology, Environment and Civil Engineering, University of Liège, Belgium
{norman.marlier, o.bruls, godefroid.dislaire, g.louppe}@uliege.be

***Abstract*— Recycling is a promising way to prevent the use of raw material and reduce energy consumption, air pollution and waste. However, the process of recycling has to be economically efficient in order to be adopted by industrial manufacturers. One way to achieve this goal is to improve the recycling rate. We propose a novel method to design a machine learning-based controller to improve the efficiency of a recycling line by throwing waste into buckets instead of picking and dropping them. Our proof-of-concept is demonstrated on stones, because of their simple and uniform shapes. The method enables an ABB IRB 340 robot to throw objects to buckets with an empirical success rate of 99%.**

I. INTRODUCTION

Automation and robotics are popular methods to improve the efficiency of production lines. Industrial robots perform quickly and with high accuracy simple tasks to manufacture objects. However, they suffer from several drawbacks. First, they are very task-specific and evolve in a controlled and structured environment. If the environment or the task slightly changes, they need to be reprogrammed. This mean they cannot deal with uncertainty. This lack of flexibility can be quite expensive for industrial manufacturers. Second, they cannot achieve complex tasks due to the implementation of explicit instructions in their programs. This can be problematic for future developments of industrial processes.

Machine learning techniques, especially reinforcement learning [1], are becoming more and more popular in the domain of artificial intelligence and robotics. They can achieve super-human performances on certain tasks, such as video games [2]. With these techniques, robots can interact with an unstructured environment and progressively learn how to act in a given situation. They are not preprogrammed anymore, but their programs evolve over time to perform better and better.

Recycling lines need to deal with the problem of sorting. The recycling line used in this study is as follows. Waste comes from a vibrating platform and is distributed over a conveyor. Waste passes through a set of sensors, a 1D camera, a hyperspectral camera, X-ray sensor and a LIBS, in order to determine the nature of the matter (aluminum, copper, zinc, lead, etc.) and the precise composition of alloys [3]. Moreover, the position on the conveyor and the mass are also determined. Waste is then sorted in different buckets in order to be recycled, which is the main goal of the line. Because of the large variety of waste, robots are employed to sort them. The current sorting method is a pick-and-drop operation. That is often inefficient when dealing with a high speed conveyor (>1m/s).

Contribution Our main contribution is the design of a robotic controller guided by a neural network classifier for throwing waste into buckets in order to save time and improve the recycling rate.

II. METHOD

A. Problem statement

The problem of throwing object into buckets is formalised as the problem of finding the action a^* in state s which maximizes the probability of success of the throw. It is equivalent to minimizing the probability of failure, $P(r = 0|s, a)$, where

$$r = \begin{cases} 1 \text{ if the throw succeeds,} \\ 0 \text{ otherwise.} \end{cases}$$

The problem is an optimization problem defined as follows,

$$a^* = \min_a P(r = 0|s, a). \tag{1}$$

We define the action space $\mathcal{A} = \{(t, y, z) \in \mathbb{R}^3\}$, where the three action variables $\{t, y, z\}$ correspond to the time t to wait before opening the gripper, the horizontal displacement y and the vertical displacement z (Fig. 1).

These variables are bounded as $t \in [60, 110], y \in [0.04, 0.1], z \in [0.04, 0.09]$, where t is in $[ms]$ and y, z are in $[m]$. We also define the state space $\mathcal{S} = \{s \in \mathbb{R}|s = [-0.2, 0.2]\}$, where s is the distance of an object from the center of the conveyor, in $[m]$.

The probability $P(r|s, a)$ is unknown but can be approximated from observed data using machine learning. The resulting approximation $\hat{P}(r|s, a)$ leads to an approximate solution of the action a^*. The final problem is thus

$$\hat{a}^* = \min_a \hat{P}(r = 0|s, a). \tag{2}$$

This problem requires only a one-step prediction which simplifies strongly the computation time and the complexity. The optimization method uses the L-BFGS algorithm [4] to converge quickly to a solution. The optimization procedure is repeated 20 times by starting from different points in order to avoid getting stuck into a local minimum.

We measure the score of our approach by the empirical success rate (ESR), which is the number of successful throws over the total number of throws.

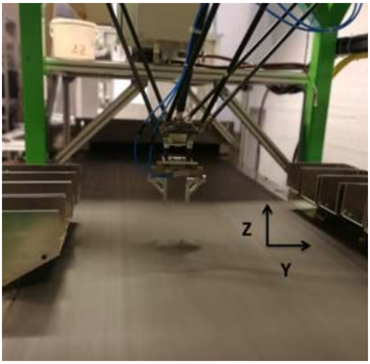


Fig. 1: The ABB robot with it pneumatic gripper on the recycling line. The buckets are located on the sides of the conveyor.

B. Model

The model used to approximate $P(r|s,a)$ is a neural network. Because r is a binary variable, this is a regular binary classification problem. The inputs of our neural network are the state and the actions, (s,a) and its output is r. We train a neural network to fit at best the relation $r = f(s,a)$ to compute $\hat{P}(r=0|s,a)$. The hyper-parameters of the model, *i.e*, the number of hidden layers and the number of neurons by layers, are chosen by cross-validation [5] with the ROC-AUC metric [6] (see Table. I). The final choice is a neural network with two hidden layers with respectively seventy and eighty neurons.

Table I: ROC-AUC obtained by cross-validation (cv=5) with different hyper-parameters.

Number of neurons	Mean score	Std score
10	0.892	0.14
30	0.905	0.12
50	0.904	0.124
70	0.907	0.118
100	0.903	0.12
10, 10	0.891	0.13
40, 30	0.9	0.097
70, 80	**0.91**	**0.081**

The choice of a neural network is also motivated by the time constraint. In fact, our small neural network is approximately an order of magnitude less time-consuming to compute the probability in contrast to ensemble trees methods, such as Extremely randomized trees [7].

The neural network was implemented with the open source library Scikit-Learn [8], as a *MLPClassifier* with Adam optimizer [9].

C. Training procedure

The training procedure is the key to quickly succeed in the task of throwing objects into buckets. The procedure needs to converge to the best action and yet to explore enough situations to generalize well. This is known as the exploration/exploitation dilemma. For practical reasons and its simplicity, the ε-greedy method [1] is chosen to solve this dilemma. This method consists in taking the best action with a probability $1-\varepsilon$ and a random action with a probability ε. The training procedure is defined as follow:

1) At step i, 50 samples are generated by the ε-greedy method and constitute the dataset $\mathcal{D}_i$.
2) 50 more samples are generated without the ε-greedy method to compute the ESR.
3) The neural network is trained by using the dataset $\mathcal{D}_i$.
4) Repeat 1-3 N times.

This process will be repeated ten times and random actions are taken to initialize the training process.

III. EXPERIMENTS

A. Simulation

The use of a simulator is a common practice in the reinforcement learning community [10, 11] because of its advantages: data are freely and quickly available, no expensive damages occur on real hardware. However, a simulator is a simplified version of the real world. A policy learnt in a simulator can fail in the real world.

Our simulator is based on several assumptions:

1) The simulator implements the equations of projectile motion with no air friction.
2) The gripper opening time is bounded between the two experimental values $80[ms]$ and $90[ms]$.
3) Velocity and acceleration are constrained due to the hardware constraints.

The simulator has two purposes. First, one can verify if the task can be achieved or not. Second, it gives an idea of how much data is needed to learn the task. It can be seen in Fig. 2 that the task is nearly perfectly learnt and only about one hundred samples are needed to learn the task. In order to choose the right value for the ε-greedy method in the real setup, four values are tested in the simulator. Because it exists few differences between these values, the value $\varepsilon = 0.1$ was chosen for the real setup.

B. Data acquisition

Learning in the simulator shows that one hundred samples of (s,a,r) are needed to achieve a good ESR, *i.e*, $\sim 90\%$. This is obviously a lower bound, because the simulator does not model the real-world complexity of the task. Because of the setup, acquiring data by hand is a tedious and error prone process. Therefore, the process was automated by using a camera to detect whether the throw is a successful by looking at the bucket. The detection method, called *Background subtraction*, compares the background, which is basically the stationary part of the image (Fig. 3b), and the foreground, which is the change in the image (Fig. 3a). We can detect if a object is in the bucket or not (Fig. 3). The algorithm is implemented in OpenCV [12] and runs on an Odroid X4 plateform.

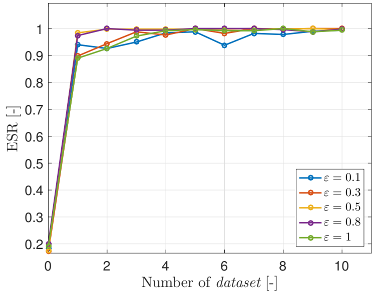
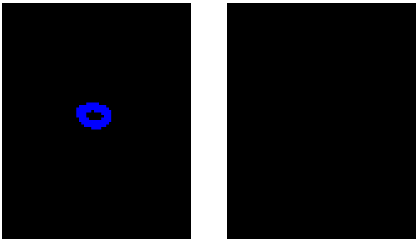


Fig. 2: ESR comparison between several values for ε-greedy policy in the simulator. One dataset corresponds to 50 samples.



(a) With an object. (b) Without.

Fig. 3: Object detection by using background subtraction method.

C. Real-life problem

Compared to the simulator, the neural network will learn more slowly in the real world setup. Indeed, as it can be seen in Fig. 4, the model needs about 250 samples to achieve a good ESR ($\sim 90\%$).

Concerning the probability of success as a function of the distance from the bucket, it can be seen in Fig. 5 that the neural network is confident along the width of the conveyor. The probability quickly drops outside the range of possible throws. But it is not relevant for this application.

A last experiment was conducted in order to compute to final score when the neural network was trained on the entire dataset. The robot throws 34 stones and succeeds in throwing all, achieving an ESR of 100%.

IV. Conclusion

Controllers based on machine learning techniques enable robots to achieve more and more complex tasks without explicit programming, as it is usually the case for industrial applications. Despite a very simple architecture, the neural network performs very well on the throwing task, achieving nearly an ESR of 99% with time constraints.

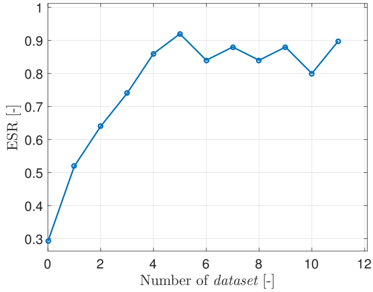


Fig. 4: ESR in real setup for $\varepsilon = 0.1$. One dataset corresponds to 50 samples.

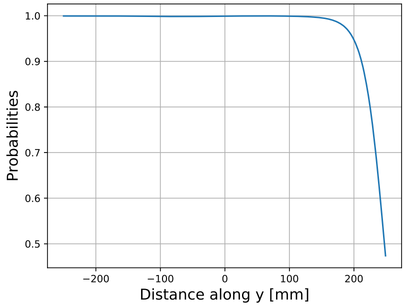


Fig. 5: The probability of success from a distance y of the bucket while taking the best action.

Furthermore, the use of a simulator was very helpful to estimate how much data is needed to learn. The limitations of the simulator were also shown: the dynamic behavior of the pneumatic gripper is very difficult to simulate.

Future work may investigate if adding more steps in the decision-making process and taking into account more sensor information, such as geometrical parameters, can improve the ESR for complex shape objects and thus get a better recycling rate.

Acknowledgement

The first author would like to acknowledge the Belgian Fund for Research training in Industry and Agriculture for its financial support (FRIA grant).

References

[1] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 2011.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[3] Pierre Barnabé, Godefroid Dislaire, Sophie Leroy, and Eric Pirard. Design and calibration of a two-camera (visible to near-infrared and short-wave infrared) hyperspectral acquisition system for the characterization of metallic alloys from the recycling industry. *Journal of Electronic Imaging*, 24(6):061115, 2015.

[4] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[5] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[6] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[7] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[8] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Mark Cutler, Thomas J Walsh, and Jonathan P How. Real-world reinforcement learning via multifidelity simulators. *IEEE Transactions on Robotics*, 31(3):655–671, 2015.

[11] Andrei A Rusu, Mel Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. *arXiv preprint arXiv:1610.04286*, 2016.

[12] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

A survey: Scheduling of Automated Guided Vehicles in Flexible (Re-)Manufacturing Systems

Sebastian Groß M. Eng., Prof. Dr.-Ing. Wolfgang Gerke

Department of Robotics and Remanufacturing
University of Applied Science Trier, Environmental Campus Birkenfeld
Campusallee, 55761 Birkenfeld, GER

w.gerke@umwelt-campus.de

s.gross@umwelt-campus.de

Prof. Dr.-Ing. Peter Plapper

Department of Intelligent Robotics

University of Luxembourg

Rue Richard Coudenhove-Kalergi, L-1359, Luxembourg

peter.plapper@uni.lu

Abstract **– This work gives a state-of-the-art overview regarding the scheduling of Automated Guided Vehicles (AGV) in Flexible (Re-)Manufacturing Systems (FMS/FRMS). Different, general approaches are available to solve this task. In this regard an overview of currently available fleet management software and research in this field as well as the simultaneous scheduling of machines and AGVs is given. Production scheduling and transport scheduling have been vastly studied by many researchers, but most of the works address these both problems separately. However, these two problems are closely linked and influence each other. By looking at them together, it is possible to achieve an improvement in the overall scheduling. The different works are examined regarding the consideration of one available machine for each tasks, or if alternative machines are available and if the approaches are static or dynamic. Especially the appropriability of the presented works regarding the use in remanufacturing systems is examined. Remanufacturing is a process for used products to make them "as good as new or better". Due to unknown condition of the used products many challenges occur during the remanufacturing process which are special to the domain of remanufacturing and not know from manufacturing. The resulting stochastic routing of products and material in particular places special demands on the flexibility and dynamic of the scheduling and control of the remanufacturing system. In order to meet the challenges of a real remanufacturing system, the scheduling algorithm should consider alternative machines as well as be dynamic. None of the approaches examined can meet these requirements, which is why new methods have to be developed.**

Index Terms – remanufacturing, scheduling, fleet management, flexible manufacturing system, simultaneous scheduling of machines and AGVs, automated guided vehicles

I. INTRODUCTION

In the manufacturing industry, the trend is more and more in the direction of a higher number of variants, which is due to the increasing customer demand for the personalization of products. The goal often mentioned here is production with batch size one. This trend requires new concepts for manufacturing systems. An example of this is the Flexible Manufacturing System (FMS) concept. With this system, different machines are available which can carry out different operations. This should make it possible to manufacture different variants of a product simultaneously. An example of such an FMS can be seen in Figure 1.

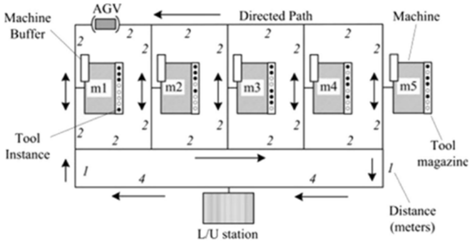


Figure 1: Example of a FMS by [1]

The material transport is not rigidly organized but is flexibly arranged by AGVs. Because of this, not only the scheduling of the machine occupancy plays a central role in these systems, but also the integration and scheduling of the AGVs within the system. The structure of remanufacturing systems is often similar, but not equal to that of FMS, why we will called them Flexible Remanufacturing System (FRMS). Because a large number of products and product variants often have to be processed in a remanufacturing system, the corresponding flexibility of the system is also required here. Due to the different, usage-dependent states of the used products, it may even be the case that two identical products have to take a different route through the remanufacturing system. In addition, the unknown product state can result in route changes even while the product is still running through the system, which places particularly high demands on the dynamic capability of scheduling and controlling machines and AGVs within a FRMS.

This paper is structured as follows: the second part will give a short overview regarding the special challenges for production planning and control in the domain of remanufacturing, the third part will give an overview about traditional methods for the integration and scheduling of AGVs as well as a state-of-the-art regarding currently available fleet management systems for AGVs as well as current research in this field, in the third part a state-of-the-art review regarding research in the field of the simultaneous scheduling of machines and AGVs will be presented. At the end identified research gaps will be explained and further research will be proposed in the conclusion.

II. Challenges in Flexible Remanufacturing Systems

In the manufacturing sector, for example, special solutions for job shop, flexible job shop [2]–[11] or re-entrant manufacturing systems are necessary and examined by researchers [12]–[16]. But none of these solutions perfectly fits to describe remanufacturing systems. This is due to the fact that in the domain of remanufacturing many challenges occur, which are special to remanufacturing and not known from manufacturing [17]–[23]. Some of these challenges are:

- Unknown product condition of the used-products due to different stress during their usage
- Unknown inspection results of the used-products
- Unknown condition of the disassembled parts
- Varying processing times
- Balance between customer demand and return of used-products
- Stochastic routings for materials and products
- Unknown arrival time and quantity of used-products

The product condition can be very different from product to product, depending on the use, which means that every product can take a different route through the remanufacturing process. This makes the production planning and control (PPC) of remanufacturing systems difficult. One reason for this is that the process steps, required remanufacturing the product at hand, are only known after an initial inspection. Even after the initial inspection, the product condition is not always completely known and therefore unexpected events may occur during the remanufacturing process. This leads to stochastic routings of the products which requires a flexible material handling system on the shop-floor like Automated Guided Vehicles (AGVs). AGVs are considered as one of the most important enablers of flexible material handling on the shop-floor [24]. This has some similarities with the already mentioned FMS, which is why we describe such systems as Flexible Remanufacturing Systems (FMRS). The further requirements for FRMS vis-à-vis FMS are described in the following section.

Due to the already described challenges in the domain of remanufacturing the scheduling and control of the remanufacturing process must be adaptive in order to be able to react adequately to the unknown schedule of a product as well as to unexpected internal and external events. Also the stochastic routings of materials and the associated scheduling and control of AGVs needs to be taken into consideration in the PPC. However, the PPC systems currently available often do not adequately reflect the complexity and volatility resulting from the above-mentioned circumstances [25]. In addition, the PPC systems usually do not integrate the management of AGVs. These are managed via external fleet management software. A holistic approach, especially in the area of joint scheduling of machines and AGVs, can result in optimization potentials.

These characteristics, unique to the domain of remanufacturing, require different approach for PPC then in the field of traditional manufacturing systems [22]. Especially the uncertainties and stochastic routings in the domain of remanufacturing need to be taken into consideration for the PPC.

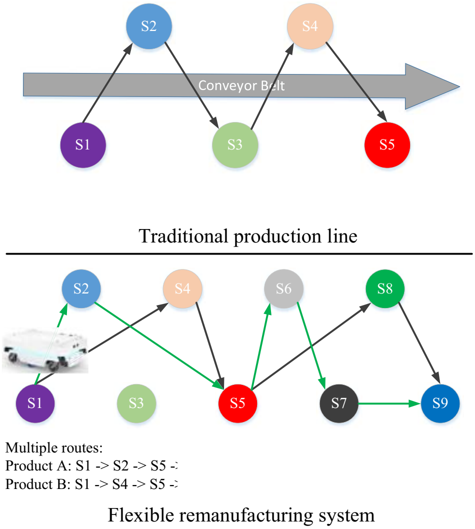


Figure 2: Comparison of a traditional production line with a Flexible Remanufacturing System and the use of AGVs as material handling systems.

III. Traditional Methods for the Integration and Scheduling of AGVs

Automated guided vehicles (AGVs) have been in use in industrial plants for several years. The use of these technologies within intralogistics results in increased flexibility of the material flow compared to conventional solutions such as conveyor belts. AGVs plan their exact path between two points independently using path planning algorithms based on a 2D map of their environment and the information where they are currently located within that map. The environment is also perceived by sensors attached to the AGV so that changes in the environment can be detected. This gives the AGV a high degree of autonomy and can deviate from its planed path if it encounters an obstacle there. For this purpose, a new calculation of the path is carried out, taking into account the current environment, so that the AGV can autonomously bypass the obstacle. AGVs are not only to be found in a purely industrial environment, but are also used, for example, in hospitals. Here, for example, the transport of food to the patient is carried out.

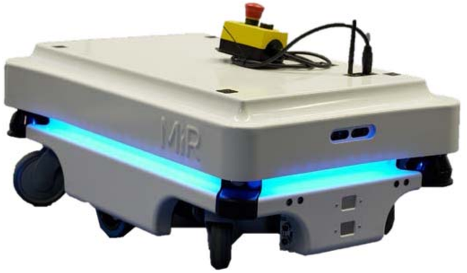


Figure 3: AGV MiR 100 from Mobile Industrial Robots.

In most practical applications, AGVs are integrated into the manufacturing system via the use of fleet management systems. These systems are usually connected to the respective ERP system. The fleet management system receives the individual transport orders to be executed from the ERP system. The fleet management system then attempts to schedule the individual orders optimally between the available AGVs, taking into account battery charge conditions and possible collisions during the journey between the AGVs. Also the integration of AGVs without the use of the coupling with the ERP system can be found. Here, for example, the individual machines independently request an AGV when completing a product. The disadvantage of this method is that global optimization is not possible due to the elimination of a central scheduling unit.

A. State-of-the-art: Fleet Manager for AGVs

AGVs are used as flexible material handling systems in the manufacturing and remanufacturing industry and able to move material without pre-defined routes. Commercial available AGVs provide different approaches of self-guided navigation in order to find a collision-free path between workstations. If several AGVs exist on the shop-fleet, fleet management systems are used for the scheduling and supervision of the AGV fleet. Currently available fleet managers just focus on the localisation and navigation of the AGVs. To minimize the transport time of materials just the optimization of routes and the allocation of the best AGV for the task at hand is taken into consideration.

Different fleet managers are developed and available from various manufacturers of AGVs. *Mobile Industrial Robots* provides the *MiRFleet* [26] which allows the collision free routing of various robots. The system also provides the ability to assign tasks with priority rules. Furthermore the system monitors the battery charge levels of the AGVs and manages automatically the charging processes. The fleet management systems from *KUKA AG,* the *KUKA.NavigationSolution* [27], and the *AGV Manager* from BA *Systèmes* [28] attempt to reduce the overall travel time by taking the production environment, the traffic and the required target location into consideration. These systems provide job scheduling and real-time routing as outputs. *DEMATIC*'s *E'tricc* AGV fleet manager [29] selects the AGVs to the tasks through analysing the work flow as well as re-evaluating assignments. An analysis of historical travel routing and operation data is implemented in the *AGV MANAGER* from *Sidel* [30] and the *SGV Manager* from *JBT* [31] in order to optimise the performance of the AGV fleet in industrial environment. The *Vehicle Manager* from *savant automation* [32] is able to process inputs from network computer systems, discrete I/O, PLC network etc. in order to assign the available AGVs to tasks. Furthermore the *Vehicle Manager* achieves and takes historical data into consideration.

Besides the listed commercially available software solutions, the subject of fleet management for AGVs is also the subject of some current research projects. Srivastava et al. [33] for example presents an agent-based approach for operation control of an AGV fleet. The goal was to find a collision-free and time optimised path in the AGV path networks. The simulation functions for evaluating different scenarios within this fleet manager provide an efficient and a validated solution for AGVs running in complex flow networks. Regarding the fleet management of multiple AGVs in industrial warehouses Cardarelli et al. [34] proposed cooperative cloud robotics architecture. Through cooperative data fusion from various sensor systems a continually updated global live view of the environment was achieved. The goal was to provide a collision-free path if unexpected obstacles occur in the environment. The methodology was successfully validated in a real industrial environment. Yao et al. [35] provides an Smart AGV Management System to optimize the scheduling of AGVs in a manufacturing process. The proposed approach uses the combination of real-time data analysis and a digital twin model to optimize the schedule. For a proof of concept, the approach was successfully tested on demonstrator with a manual assembly station.

However none of the listed, available software solutions as well as none of the stated research projects provides the possibility regarding an integration for the simultaneous scheduling of machines and AGVs to optimize the overall scheduling with the use of real-time production data [35]. This leads to a not optimal schedule and leaves room for optimization potential. One possibility to realize this optimization potential is the simultaneous scheduling of machines and AGVs which will be examined in the next chapter.

IV. State-of-the-art simultaneous scheduling of machines and AGVs

Traditionally, planning problems have considered machines as the only important resource, but this is no longer true as material handling in an FMS becomes more valuable and transport times contribute to machine downtime as machines have to wait for the next part to be machined. Extensive research has been devoted to machine scheduling and vehicle scheduling independently, but the two problems are closely linked. Few have directed their research to the simultaneous planning of machines and AGVs in an FMS. The scheduling in an FMS is conceptually similar to Job-Shop Scheduling Problems (JSSP), with the difference that JSSP do not consider material handling. The goal of the JSSP is to assign the operation out of a set of jobs to a set of machines while minimizing the makespan. Sequence conditions between the operations of a job and the fact that one machine can only one operation at one time has to be taken into consideration. In the JSSP every operation is processed on one specific machine, whereas in the extension of the JSSP the Flexible Job-Shop Scheduling Problem (FJSSP) every operation can be processed on one or more machines. The JSSP and the FJSSP are NP-hard problems [36] and studied by many researchers. In the simultaneous scheduling of machines and AGVs (SSMA), all available machines are considered as well as the available AGVs, which have to transport the individual products between the machines. Only few research is done regarding the simultaneous scheduling of machines and AGVs. In the literature there are mainly studies which consider in this sense a JSSP, but no many Flexible Job-Shop Scheduling Problem (FJSSP), in with alternative machines for the execution of operations are taken into consideration. In Figure 4 the joint schedule for machines (M1… M4) and AGVs (V1, V2) is shown. For the AGV schedule L marks the loaded trips und E the empty trips, where the AGV moves but no product is transported. Regarding the machines each colored block is an operation executed on this specific machine. The first number within these blocks represents the job to which the operation belongs and the second number represents the operation within the according job. White blocks represents idle time of the machine.

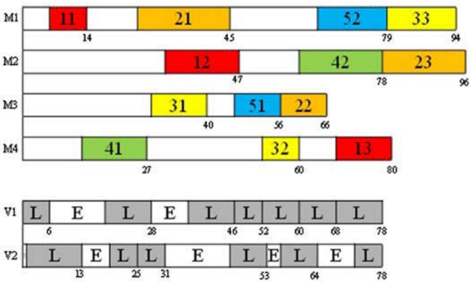


Figure 4: Gantt chart representation of the schedule for machines and AGVs by [1]

The FJSSP can be described as follows [37]:

Parameters: (i=1,…,m; j=1,…,n; h=1,…,h$_j$)

n number of jobs

m number of machines

$a_{i,j,h}$ describes whether a machine from the available machines $M_{j,h}$ can perform the respective operation $O_{j,h}$

$$a_{i,j,h} = \begin{cases} 1 \text{ if } O_{j,h} \text{ can be executed on machine i} \\ 0 \text{ else} \end{cases}$$

$a_{i,j,h}$ process time of operation $O_{j,h}$ on machine $i(p_{i,j,h} > 0)$

C_{max} makespan, maximum execution time of the orders

$$y_{i,j,h} = \begin{cases} 1 \text{ if machine } i \text{ is choosen for operation } O_{j,h} \\ 0 \text{ else} \end{cases}$$

$$x_{i,j,h,k} = \begin{cases} 1 \text{ if operation } O_{j,h} \text{ is executed on machine } i \text{ with priority } k \\ 0 \text{ else} \end{cases}$$

$t_{j,h}$ start time for operation $O_{j,h}$

$Tm_{i,k}$ start time for machine i with the priority k

k_i number of operations assigned to machine i

$Ps_{j,h}$ processing time of operation $O_{j,h}$ on the assigned machine

Goal is to minimize C_{max}:

$$C_{max} \geq t_{j,h_j} + Ps_{j,h_j} \quad \text{für} \quad j = 1,\ldots,n$$

With the extension to the simultaneous scheduling of machines and AGVs the objective of the problem is to minimize the makespan defined through the following formulation [1]:

$P_{i,j}$ is processing time for operation i of job j

$T_{i,j}$ is the travel time for operation i of job j

$L_{i,j}$ is the operation completion time for operation i of job j

$$L_{i,j} = T_{i,j} + P_{i,j}$$

C_j is the completion time of job j: $C_j = \sum_{j=1}^{n} L_{i,j}$

Objective function: *Min makespan = Max(C$_1$, C$_2$, C$_3$, … , C$_n$)*

In the following a state-of-the-art regarding the simultaneous will be given, which will be split in research works where a JSSP is studied and in works where a FJSSP is studied.

A. SSMA in the JSSP environment

The first study which considers the simultaneous scheduling of machines and AGVs is published by Bilge and Ulusoy [38]. Benchmark instances for the problem were presented. These include four different layouts with regard to the arrangement of the machines under consideration, resulting in different travel distances and thus travel times between the machines. In addition, ten order sets were presented, which are to be executed accordingly on each of the layouts. These are still used today as benchmark instances. A non-linear mixed integer programming model was presented to solve the problem.

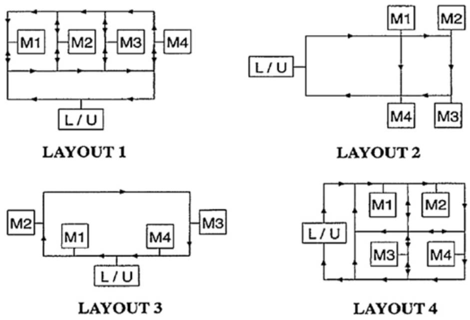


Figure 5: Layout for the benchmark instances from Bilge und Ulusoy [38].

Nageswararao et al. [39] propose a Binary Paricle Swarm Vehicle Heuristic Algorithm (BPSVHA) for the simultaneous planning of AGV's and machines in the area of flexible manufacturing systems (FMS). The algorithm was compared with other metaheuristic algorithms. In the study, however, no dynamic was considered. Erol et al. [40] propose a multi-agent system implemented in JADE for the dynamic and simultaneous planning of machines and AGVs in manufacturing systems. The proposed system was tested against five optimization algorithms for deterministic cases. Compared to the optimization algorithms, almost exclusively poor results were achieved with respect to the benchmarks investigated. The consideration of the flexibility of the approach required for the insertion of new orders, routing flexibility, machine failure, etc. was not part of this study. In addition, the approach was tested against different traditional dispatching, with the result that the proposed MAS outperformed most of them. Mousavi et al. [41] present a hybrid algorithm consisting of genetic algorithm and particle swarm optimization for simultaneous planning of machines and AGVs. The aim of the optimization is the minimization of the throughput time as well as the minimization of the required AGVs. The necessity of battery charging processes was also taken into account in the optimization. In this study, however, neither the consideration of dynamics nor the consideration of alternative machines took place. Chaudhry et al. [42] proposed a genetic algorithm for the solution of the simultaneous planning of machines and AGVs and tests this algorithm at the benchmark instances of Bilge and Ulusoy. Here, too, neither alternative machines nor dynamics are considered. Fontes and Homayouni [43] use a mixed integer linear programming model, which was implemented in the commercial software Gurobi. Disadvantage of this method is the partly very high computing time of the method for finding good solutions. In this study, however, neither the consideration of dynamics nor the consideration of alternative machines took place. Lacomme et al. [44] proposed a modified disjunctive graph to model the simultaneous scheduling problem the a memetic algorithm to solve the scheduling problem. The approach was tested on the instances of Bilge and Ulusoy. Dynamic was not taken into consideration. Fauadi and Murata [45] proposed an Binary Particle Swarm Optimization (BPSO) to solve the scheduling problem with the goal to minimize the makespan. The approach was tested on the instances from Bilge und Ulusoy and compared with their results. The average values of all instances per layout were compared. The BPSO presented achieved a better result in all four layout variants compared to the results of Bilge and Ulusoy [38]. However dynamic was not taken into consideration.

B. SSMA in the FJSSP environment

The simultaneous scheduling of machines and AGVs in an FMS with alternative machines for operations, respectively modelled as a FJSSP has not been researched a lot. As far as the authors are aware, only seven works provided solution approaches for this problem. These nine works consists of six works using heuristic approaches, one mixed integer linear programming (MILP) approach, one simulation-based approach and one approach based on the use of a multi-agent system. The first study of the simultaneous planning of machines and AGVs in the context of a FJSSP, i.e. with alternative machines, was published by Deroussi and Norre [46]. It was suggested to use an Iterative Local Search (ILS). An extension of the instances of Bilge and Ulusoy [38] by alternative machines as well as an extension of known FJSSP benchmark instances by layouts concerning the routes of FTS between the machines was also proposed. Zhang et al. [10] proposed a hybrid algorithm with a combination of a genetic algorithm (GA) and a Tabu-Search (TS) to address the scheduling problem. In the first step the GA is used to assign every operation to a machine and each transport task to an AGV. Afterwards the TS is used to optimize the solution of the GA. In [47] Zhang et al. improved their approach through the implementation of a shifting bottleneck (SBN) procedure. Like in [10] the GA is used in the first step to allocate machines and AGVs to every operation respectively transport task. Afterwards the SBN is used to find a scheduling for machines and AGVs, which is then improved through the TS. For validation of the proposed approach computational experiments using the problem instances from [46] were executed. The approaches from Zhang et al. found a better solution for two of the ten instances while for six instances the same makespan was found. Kumar et al. [48] uses a combination of Differential Evolution (DE) algorithm to plan the operation sequence and uses a vehicle assignment heuristic presented by him for the subsequent assignment of the FTS. Finally, a machine selection heuristic checks whether the use of an alternative machine for one of the operations leads to an improvement of the cycle time. This process is repeated iteratively. In contrast to the aforementioned approaches, alternative machines are considered in the sense of an FJSSP, but deterministic is assumed and no dynamics are considered. Sahin et al. [49] propose a multi-agent system for the simultaneous planning of machines and material flow systems in the environment of a flexible manufacturing system. The system was developed with Prometheus and programmed in the JACK environment. The proposed dynamic planning system was compared with other approaches using dynamic and static problem sets. In addition, as in [48], a consideration of alternative machines is carried out. However, in comparison to the results from [48], the results of the presented MAS are considerably worse in most benchmark instances and in only a few cases equally performant, or better. Lin et al. [1] proposed a simulation-based optimization approach for the simultaneous planning of machines and AGVs with uncertain machining times in FMS. To explore good design alternatives based on the simulation result, they used a combination of local search and a genetic algorithm. In addition to uncertain processing times, other noise factors such as congestion and downtime in AGV handling were also considered. To validate their approach, Lin et al. used different benchmark sets. Deroussi [50] proposes a hybridization of particle swarm optimization (PSO) with a stochastic local search. The PSO is used to solve the AGV scheduling problem and then each operation is allocated to the first available machine, finally sequencing of the operations for each machine is achieved by selecting the operations according to their arrival time in the machine buffer. A local search procedure based on the classical exchange and insertion movements is then used to try to improve the existing solution. The authors only reported results of average performance on the ten problem cases that were of lower quality than those of [10], [47]. Nouri et al. [51] recently proposed a hybrid metaheuristic consisting of a combination of a GA and a TS. The GA is used to create solutions and a TS is used to improve these solutions. The GA has a two-part chromosome, the first part represents the assignment problem with regard to the machines, while the second part represents the sequence problem. At the end of each GA generation, the solutions obtained are bundled according to proximity, i.e. the population is divided into a given number of groups. Then TS is applied to the best solution within each group to search for a better solution in its neighborhood. The authors report improvements in the results regarding [10], [47] in seven benchmark instances. However, in [52] it is shown that the results in [51] are not plausible and possible. This is justified by the fact that the lowest possible lead time of the shortest order is already higher than the lead time presented in [52] for the entire order set. Homayouni and Ponto [52] present a MILP model for solving the simultaneous planning of machines and AGVs in the context of an FJSSP and can achieve the same or better results than [10] in all benchmark instances investigated. However, the computing time is really higher, which can severely limit the procedure in real use.

Figure 6 gives an overview regarding the present works in the field of simultaneous scheduling of machines and AGVs (SSMA). Whereas there are some more works published regarding the SSMA in the JSSP there are, to the author's best knowledge, only the nine presented works published regarding the SSMA in the JSSP. Regarding works which include dynamic as well as FJSSP only two works can be found.

Article	Methode	Dynamic	FJSSP	JSSP
Bilge and Ulusoy [38]	Non-linear mixed integer programming model	O	O	✓
Nageswararao et al. [39]	Binary Paricle Swarm Vehicle Heuristic Algorithm (BPSVHA)	O	O	✓
Erol et al. [40]	Multi-Agent System	O	O	✓
Mousavi et al. [41]	Hybrid algorithm consisting of Genetic Algorithm and Particle Swarm Optimization	O	O	✓
Chaudhry et al. [42]	Genetic Algortihm	O	O	✓
Fontes and Homayouni [43]	Mixed Integer linear Programming model	O	O	✓
Lacomme et al. [44]	Memetic Algorithm	O	O	✓
Faudi and Murata [45]	Binary Paricle Swarm Optimization	O	O	✓
Deroussi and Norre [46].	Iterative Local Search (ILS)	O	✓	O
Zhang et al. [12]	Hybrid Algorithm with a combination of a Genetic Algorithm and a Tabu-Search	O	✓	O
Zhang et al. [47]	Hybrid Algorithm with a combination of a Genetic Algorithm, a Shifting Bottleneck Procedure and a Tabu-Search	O	✓	O
Kumar et al. [48]	Hybrid Algorithm with a Differential Evolution algorithm a Vehicle Assignment Heuristic and a Machine Selection Heuristic	O	✓	O
Sahin et al. [49]	Multi-Agent System	✓	✓	O
Lin et al. [1]	Simulation-based optimization	✓	✓	O
Deoussi et al. [50]	Hybrid algortihm of Particle Swarm Optimization with a stochastic Local Search	O	✓	O
Nouri et al. [51]	Hybrid Algorithm of a Genetic Algorithm and a Tabu-Search	O	✓	O
Homayouni and Ponto [52]	Mixed Integer Linear Programming model	O	✓	O

Figure 6: Overview of the presented works regarding the simultaneous scheduling of machines and AGVs.

V. CONCLUSION

The paper at hand presents a state-of-the-art overview regarding fleet management systems and approaches for the simultaneous scheduling of machines and AGVs. Compared to the traditional approach to integration and scheduling of AGVs through fleet management systems, simultaneous scheduling offers advantages in that the two closely related and interacting problems of machinery and transportation planning are considered together. This allows improvements to be made to the overall efficiency of the system. However, the simultaneous scheduling currently only takes place in an academic environment and no commercial systems can be found to realize this. In addition, there are hardly any studies that investigate a realistic view of dynamics and alternative machines. Here there is a need for further research, especially when considering machine failures, AGV failures, or changing the route of the product during processing. The application of the control of an FMS by means of simultaneous sequence planning of machines and AGVs has also not yet been investigated in the form of a case study.

Therefore, in future work at the Trier University of Applied Sciences, Environmental-Campus Birkenfeld, the simultaneous scheduling with the consideration of dynamics will be investigated. In addition, a case study for the control of a flexible remanufacturing system using the simultaneous scheduling of machines and AGVs will be carried out at a model factory.

ACKNOWLEDGMENTS

This work was funded by „ European Regional Development Fund“ (EFRE) in the scope of the INTERREG V A Program Greater Region within the Project “Robotix-Academy”.

REFERENCES

[1] J. T. Lin, C. C. Chiu, and Y. H. Chang, “Simulation-based optimization approach for simultaneous scheduling of vehicles and machines with processing time uncertainty in FMS,” *Flex. Serv. Manuf. J.*, pp. 1–38, 2017.

[2] K. Z. Gao, P. N. Suganthan, T. J. Chua, C. S. Chong, T. X. Cai, and Q. K. Pan, “A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion,” *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7652–7663, 2015.

[3] J. Gao, L. Sun, and M. Gen, “A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems,” *Comput. Oper. Res.*, vol. 35, no. 9, pp. 2892–2907, 2008.

[4] S. V. Kamble, S. U. Mane, and A. J. Umbarkar, “Hybrid Multi-Objective Particle Swarm Optimization for Flexible Job Shop Scheduling Problem,” *Int. J. Intell. Syst. Appl.*, vol. 7, no. 4, pp. 54–61, 2015.

[5] G. Zhang, X. Shao, P. Li, and L. Gao, “An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem,” *Comput. Ind. Eng.*, vol. 56, no. 4, pp. 1309–1318, 2009.

[6] J. Q. Li, Q. K. Pan, and K. Z. Gao, “Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems,” *Int.*

J. Adv. Manuf. Technol., vol. 55, no. 9–12, pp. 1159–1169, 2011.
[7] J. Q. Li, Q. K. Pan, and Y. C. Liang, "An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems," *Comput. Ind. Eng.*, vol. 59, no. 4, pp. 647–662, 2010.
[8] L.-N. Xing, Y.-W. Chen, P. Wang, Q.-S. Zhao, and J. Xiong, "A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems," *Appl. Soft Comput.*, vol. 10, no. 3, pp. 888–896, 2010.
[9] M. Yazdani, M. Amiri, and M. Zandieh, "Flexible job-shop scheduling with parallel variable neighborhood search algorithm," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 678–687, 2010.
[10] Q. Zhang, H. Manier, and M.-A. Manier, "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1713–1723, 2012.
[11] H. E. Nouri, O. Belkahla Driss, and K. Ghédira, "Controlling a Single Transport Robot in a Flexible Job Shop Environment by Hybrid Metaheuristics," *LNCS Trans. Comput. Collect. Intell.*, vol. 28, pp. 93–115, 2018.
[12] A. Kouider and B. Bouzouia, "Multi-agent job shop scheduling system based on co-operative approach of idle time minimisation," *Int. J. Prod. Res. - INT J PROD RES*, vol. 50, pp. 409–424, 2012.
[13] P. Lou, S. K. Ong, and A. Nee, "Agent-based distributed scheduling for virtual job shops," *Int. J. Prod. Res. - INT J PROD RES*, vol. 48, pp. 3889–3910, 2010.
[14] M. E. Leusin, M. Kück, E. M. Frazzon, M. U. Maldonado, and M. Freitag, "Potential of a Multi-Agent System Approach for Production Control in Smart Factories," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1459–1464, 2018.
[15] B. Ramasubramanian, "Re-entrant Manufacturing Systems," pp. 1–27.
[16] J. Zhang, "Multi-Agent-Based Production Planning and Control," 2017.
[17] M. L. Junior and M. G. Filho, "Production planning and control for remanufacturing : Literature review Production Planning & Control : The Management of Operations Production planning and control for remanufacturing : literature review and analysis," *Prod. Plan. Control*, no. June, 2012.
[18] C. Zikopoulos, "Remanufacturing lotsizing with stochastic lead-time resulting from stochastic quality of returns," *Int. J. Prod. Res.*, vol. 55, pp. 1–23, 2016.
[19] R. Zhang, S. K. Ong, and A. Y. C. Nee, "A simulation-based genetic algorithm approach for remanufacturing process planning and scheduling," *Appl. Soft Comput.*, vol. 37, pp. 521–532, Dec. 2015.
[20] P. Golinska-dawson, M. Kosacka, and A. Nowak, "Automotive Parts Remanufacturing – Experience of Polish Small Automotive Parts Remanufacturing – Experience of Polish Small Companies," no. January, 2015.
[21] M. Andrew-Munot, A. Yassin, S. T. Syed Shazali, M. Sawawi, S. J. Tanjong, and N. Razali, "Analysis of production planning activities in remanufacturing system," *J. Mech. Eng. Sci.*, vol. 12, no. 2, pp. 3548–3565, 2018.
[22] Y. Cui, Z. Guan, C. He, and L. Yue, "Research on remanufacturing scheduling problem based on critical chain management," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 215, no. 1, 2017.
[23] V. D. R. Guide, "Production planning and control for remanufacturing: industry practice and research needs," *J. Oper. Manag.*, vol. 18, no. 4, pp. 467–483, 2000.
[24] Shiyong Wang, Jiafu Wan, Di Li, and Chunhua Zhang, "Implementing Smart Factory of Industrie 4.0: An Outlook," *Int. J. Distrib. Sens. Networks*, vol. 2016, 2016.
[25] A. G. Schuh *et al.*, "Cyber Physical Production Control," in *Industrial Internet of Things: Cybermanufacturing Systems*, S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, Eds. Cham: Springer International Publishing, 2017, pp. 519–539.
[26] Mobile Industrial Robots, "MiRFleet | Mobile Industrial Robots." [Online]. Available: https://www.mobile-industrial-robots.com/de/products/mir-add-ons/mirfleet/. [Accessed: 25-Mar-2019].
[27] K. AG, "KUKA.NavigationSolution | KUKA AG." [Online]. Available: https://www.kuka.com/en-gb/products/mobility/navigation-solution. [Accessed: 25-Mar-2019].
[28] BA Systèmes, "AGV Manager: AGV fleet management software | BA Systèmes." [Online]. Available: https://www.basystemes.com/en/agvs/supervision/. [Accessed: 25-Mar-2019].
[29] DEMATIC, "E'tricc - Cutting Edge AGV Management Software." [Online]. Available: http://egeminusa.com/automated-guided-vehicles/software/etricc/. [Accessed: 25-Mar-2019].
[30] Gebocermex, "AGV supervision system." [Online]. Available: https://www.gebocermex.com/en/intralogistic-systems/automated-guided-vehicles/agv-supervision-system-pd-111. [Accessed: 25-Mar-2019].
[31] JBT, "SGV Manager - JBT." [Online]. Available: https://www.jbtc.com/automated-systems/products-and-applications/products/software-and-services/software/sgv-manager. [Accessed: 25-Mar-2019].
[32] savant automation, "Controls & Software | Savant Automation." [Online]. Available:

http://www.agvsystems.com/controls-software/. [Accessed: 25-Mar-2019].

[33] S. C. Srivastava, A. K. Choudhary, S. Kumar, and M. K. Tiwari, “Development of an intelligent agent-based AGV controller for a flexible manufacturing system,” *Int. J. Adv. Manuf. Technol.*, vol. 36, no. 7–8, pp. 780–797, 2008.

[34] E. Cardarelli, V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, “Cooperative cloud robotics architecture for the coordination of multi-AGV systems in industrial warehouses,” *Mechatronics*, vol. 45, pp. 1–13, 2017.

[35] F. Yao, A. Keller, M. Ahmad, B. Ahmad, R. Harrison, and A. W. Colombo, “Optimizing the Scheduling of Autonomous Guided Vehicle in a Manufacturing Process,” *Proc. - IEEE 16th Int. Conf. Ind. Informatics, INDIN 2018*, pp. 264–269, 2018.

[36] M. R. Garey, D. Johnson, and R. Sethi, “The Complexity of Flowshop and Jobshop Scheduling,” *Math. Oper. Res. - MOR*, vol. 1, pp. 117–129, 1976.

[37] P. Fattahi, M. Saidi Mehrabad, and F. Jolai, “Mathematical modeling and heuristic approaches to flexible job shop scheduling problems,” *J. Intell. Manuf.*, vol. 18, no. 3, pp. 331–342, 2007.

[38] Ü. Bilge and G. Ulusoy, “A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS,” *Oper. Res.*, vol. 43, no. 6, pp. 1058–1070, 1995.

[39] M. Nageswararao, K. Narayanarao, and G. Ranagajanardhana, “Simultaneous Scheduling of Machines and AGVs in Flexible Manufacturing System with Minimization of Tardiness Criterion,” *Procedia Mater. Sci.*, vol. 5, pp. 1492–1501, 2014.

[40] R. Erol, C. Sahin, A. Baykasoglu, and V. Kaplanoglu, “A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems,” *Appl. Soft Comput. J.*, vol. 12, no. 6, pp. 1720–1732, 2012.

[41] M. Mousavi, H. J. Yap, S. N. Musa, F. Tahriri, and S. Z. Md Dawal, “Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization,” *PLoS One*, vol. 12, no. 3, pp. 1–24, 2017.

[42] I. A. Chaudhry, S. Mahmood, and M. Shami, “Simultaneous scheduling of machines and automated guided vehicles in flexible manufacturing systems using genetic algorithms,” *J. Cent. South Univ. Technol.*, no. October 2011, 2011.

[43] D. B. M. M. Fontes and S. M. Homayouni, “Joint production and transportation scheduling in flexible manufacturing systems,” *J. Glob. Optim.*, no. June 2018, pp. 1–30, 2018.

[44] P. Lacomme, M. Larabi, and N. Tchernev, “Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles,” *Int. J. Prod. Econ.*, vol. 143, no. 1, pp. 24–34, 2013.

[45] M. H. F. Bin Md Fauadi and T. Murata, “Makespan minimization of machines and Automated Guided Vehicles schedule using Binary Particle Swarm Optimization BT - International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, March 17, 2010 - March 19, 2010,” vol. III, pp. 1897–1902, 2010.

[46] L. Deroussi and S. Norre, “Simultaneous scheduling of machines and vehicles for the flexible job shop problem Solution approach : basic ideas,” *Int. Conf. Metaheuristics Nat. Inspired Comput.*, no. February, pp. 2–3, 2010.

[47] Q. Zhang, H. Manier, and M.-A. Manier, “Metaheuristics for Job Shop Scheduling with Transportation,” in *Metaheuristics for Production Scheduling*, 2013, pp. 465–493.

[48] M. V. S. Kumar, R. Janardhana, and C. S. P. Rao, “Simultaneous scheduling of machines and vehicles in an FMS environment with alternative routing,” *Int. J. Adv. Manuf. Technol.*, vol. 53, no. 1–4, pp. 339–351, 2011.

[49] C. Sahin, M. Demirtas, R. Erol, A. Baykasoğlu, and V. Kaplanoğlu, “A multi-agent based approach to dynamic scheduling with flexible processing capabilities,” *J. Intell. Manuf.*, vol. 28, no. 8, pp. 1827–1845, 2017.

[50] L. Deroussi, “A Hybrid PSO Applied to the Flexible Job Shop with Transport,” in *Swarm Intelligence Based Optimization*, 2014, pp. 115–122.

[51] H. E. Nouri, O. Belkahla Driss, and K. Ghédira, “Simultaneous scheduling of machines and transport robots in flexible job shop environment using hybrid metaheuristics based on clustered holonic multiagent model,” *Comput. Ind. Eng.*, vol. 102, pp. 488–501, Dec. 2016.

[52] S. M. Homayouni and D. B. M. M. Fontes, “Joint scheduling of production and transport with alternative job routing in flexible manufacturing systems,” *AIP Conf. Proc.*, vol. 2070, no. January, 2019.