Interreg

EUROPEAN UNION

Grande Région | Großregion

Robotix-Academy

Fonds européen de développement régional | Europäischer Fonds für regionale Entwicklung

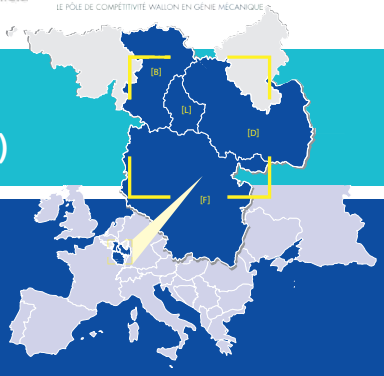
Axe prioritaire | Prioritätsachse 4
Compétitivité et attractivité
Wettbewerbsfähigkeit und Attraktivität

Robotix-Academy Conference for Industrial Robotics (RACIR) 2017

Partenaires du projet | Projektpartner:

ZeMA · uni.lu Université du Luxembourg · Liège université · Université de Lorraine · Hochschule Trier Umwelt-Campus Birkenfeld · Pôle MecaTech

Rainer Müller, Peter Plapper, Olivier Brüls, Wolfgang Gerke,
Gabriel Abba, Bassem Hichri, Matthias Vette-Steinkamp (Hrsg.)



www.robotix.academy

Robotix-Academy Conference for Industrial Robotics (RACIR)

Preface:

Robotix-Academy Conference for Industrial Robotics (RACIR) is held in Luxembourg University during June 06-07, 2017. The venue for RACIR 2016 is the Campus Kirchberg in Luxembourg. The University of Luxembourg aspires to be one of Europe's most highly regarded universities with a distinctly international, multilingual and interdisciplinary character. It fosters the cross-fertilistion of research and teaching, is relevant to its country, known worldwide for its research and teaching in targeted areas, and becomes an innovative model for contemporary European Higher Education. Today, after over ten years of intense developments, the University has an internationally relevant research University with students originating from 115 countries, academic staff from 20 countries as well as 78 partner universities around the globe.

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 acknowledge for their contributions and participation to the conference.

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

Content

Towards Intelligent Robot Assistants for the non-destructive Disassembly of End of Life Products

Jan Jungbluth
Technology Research Group
SEW EURODRIVE
Bruchsal, Germany
jan.jungbluth.w@sew-eurodrive.de

Wolfgang Gerke
Faculty of Environmental Planning and Technology
University of Applied Sciences Trier
Birkenfeld, Germany
w.gerke@umwelt-campus.de

Peter Plapper
Faculty of Science, Technology and Communication
University of Luxembourg
Luxembourg, Luxembourg
peter.plapper@uni.lu

***Abstract* - The effective collaboration between humans and robots in complex and task rich environments like End of Life product disassembly depends on the ability of the robot to anticipate the workflow as well as the assistance the human co-worker wants. Our approach towards such an intelligent system is the development of an informed software agent that controls the robot assistance behavior. We inform the agent with procedural and declarative knowledge about the disassembly domain through models of the product structure and actor/object models. The product structure is then transformed to a directed graph and used to build, share and define a goal-orientated coarse workflow. Depending on the tasks and wanted assistance, the system can generate adaptable and detailed workflows through searching in the situation space on the basis of predefined and task dependent actions. The created detailed workflow consists of a sequence of actions that are used to call, parameterize and execute robot programs for the fulfillment of the assistance. The aim of this research is to equip robot systems with higher cognitive skills to allow them to be autonomous in the performance of their assistance to improve the ergonomics of disassembly workstations.**

Index Terms – Human-Robot Interaction, Human-Robot Collaboration, Robot Assistant, Cognitive Robotics, Disassembly Assistance.

I. INTRODUCTION

Disassembling is involved in many processes, for example remanufacturing, corrective maintenance, proper disposal and manufacturing. However, fully automated disassembly lines are, compared to product assembly lines, rare. One reason is that disassembling at the end of a product lifetime is much harder to automatize than assembling. In disassembly we have to cope with fouling, wear, damaged or only absent parts. Furthermore, we have to deal with product manipulations, such as individual extensions or improvised fixes, which are not obviously visible. It is the unpredictable condition of a product that prohibits further automation. Even if better sensor technology could identify inappropriate product conditions, it would be impossible, or highly expensive, to treat all possibilities in a fully automated manner. Other challenges are the small lot sizes or individualized products (lot size 1). Today's fully automated processes are not flexible enough for the treatment of different products or variants. In corrective maintenance, it is a normal case to have an unsteady, unpredictable flow of different products. Dismantling processes in corrective maintenance may also have different target stages, depending on what part has to be replaced. Furthermore, there is, especially in central waste recycling plants, a lack of information about the product structure. For example, types of materials in the product. These are reasons why disassembly workplaces stayed unautomated, thus resulting in a bad situation for the workforce, who remain exposed to health problems due to heavy workload. In addition, the economic fitness of disassembly processes is highly reduced by the substantial amount of manual labor in the destructive disassembly process. Consequently, important concepts for environmental protection, such as remanufacturing, cannot spread wider into industry.

A solution for this problem would be intelligent robot-based assistants, which would be a compromise between automation and manual labor with great advantages in the disassembly domain. One advantage is that only humans have the cognitive abilities to identify and handle the aforementioned unexpected situations. Thus, humans are able to ensure the overall success of the process by contributing their awareness to adapting the process to a situation [1]. The robot instead can provide assistance with power and endurance over the complete disassembly process and thereby improve the ergonomics of the workplace. Therefore, multi-skilled robots may take over automatable tasks, such as unscrewing, or support the execution of tasks, e.g., handling of heavy parts. Especially over the 10 last years, with the development of lightweight and force-sensitive robots, new robot skills have extended the robots' fields of application. Those new skills provide realizable "assistance opportunities" in disassembly procedures, and some of them could be integrated into one robotic system. However, once integrated, the problem arises of controlling the multi-skilled robot and its behavior to assist the user in a situation- and goal-oriented manner. Even providing the system with needed precise positions information is big issue [2]. Manually programming the whole process would not be meaningful [3], especially if we consider many different products. Only intelligent systems with higher autonomy could support and interact fluently with humans in such a dynamic environment [4, 5]. Therefore, the system needs to identify the disassembly process and the work contend by itself. Then, the robot must

know how to assist the user in a certain task; it also needs necessary information and skills to perform this task. To meet these challenges, we take the approach of an informed software agent, the architecture of which we present and explain in Section 2. In Section 3, we explain the kind of information that is provided to the agent and in section 4 we clarify how we decompose the complex disassembly problem into individual subproblems. Coarse disassembly planning is used to identify the task sequence that strips down the assembly. In Section 5, we present our approach to assigning the work content of each task to the participants through a planning search in the possible situation space. Explained in Section 6 is how we execute the determined detailed plan by invoking and parametrizing robot programs to finally control the robot assistance behavior.

II. The Agent-Based Robot Control Architecture

Today's industrial robot controllers are not suitable for human -robot interaction in complex environments such as disassembly workstations [1]. In such environments, the objective and the boundary condition change from task to task and require some planning, higher skills, and knowledge to succeed efficiently. Our approach to empower industrial robots to this application field is to overlay the robot controller with an intelligent agent-based control system, which does high-level planning, defines and controls the robot behavior, e.g., the type of assistance, and guarantees the necessary information to the underlying robot controller. The developed software agent, which we present in this section, consists of different software modules (see Figure 1). In the agent's "Knowledge Base" module, we store a symbolic representation of the product structure, which we call the product model. The model contains information about the parts and how the parts are connected with each other in the assembly. Therefore, we model different classes of parts and connections. We also model the actions that the actors (the user and the robot) provide to the overall system, to act in the environment or on the product structure. We explain all the stored information in further detail at the time of its use, following the systems information flow and processing. In the "Coarse Planning" module we create, depending on the disassembly objective and the product model, a coarse plan that consists of a sequence of disassembly steps. Each disassembly step consists of one or more tasks. A task defines an independent subproblem, which is the removal of a specified connection and the referenced parts. Moreover, we use the coarse planning to assign the means of production, such as tools, robot effectors, and carriage cases, to a task. This is done through the "Stock Manager" module. Depending on the connection type, different assistance behaviors of the robot are available and selectable in a task by the user. Furthermore, the user can rearrange the task order and manipulate parts' condition states, for example, to mark a damaged part, over the human-machine-interface (HMI) module to adapt the process to the circumstances and his/her will. The coarse plan is then processed task by task by the "Detail Planning" module, in which we first create a discrete state space representation of the current and the target situations. A search through the possible state space, which considers the actions provided by the user and the robot, leads to a set of possible action sequences from which the fittest sequence is selected for execution. The selected action sequence is then performed, monitored and synchronized by the "Control" module through advising the user over the "HMI" module or invoking and parameterizing programs on the underlying robot controller.

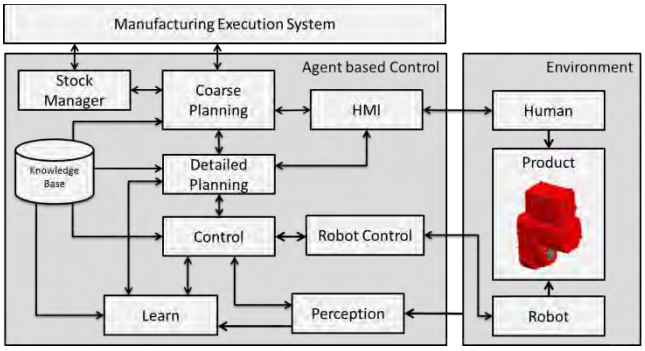


Fig. 1 The agent's architecture and the information flow between the different software modules.

Currently under construction is the "Perception" module. In the first place, it is foreseen to track the human hands to synchronize the process without action-executed confirmation-button pressing. We use the "Perception" module also to recognize and interpret gestures and voice commands to alter the robot behavior in the execution phase. The "Learn" module is considered in the agent architecture to adapt the assistance behavior to an individual user. A possible purpose of machine learning is to predict the user's desire for an assistance behavior by comparing the current situation with similar situations from recorded older interactions. Also, the agent could improve the coarse planning, particularly the ordering of parallel executable tasks, by learning from manually adapted plans. Furthermore, recording and analyzing such disassembly processes could produce valuable data and lead to deeper insights into the disassembly process. Next, we explain the system in more detail, starting in the knowledge base and considering the product model.

III. The Product Model

A technical product is an assembly of parts that are linked together through connections. The step-by-step removal of these connections is the process of disassembling. The sequence of removing the connections is not arbitrary but partially defined through the product structure. Finding this disassembly sequence is known as disassembly planning, which is the topic of Section 4. So to generate the disassembly sequence later, we first have to present the structure in an appropriate and machine readable manner. Disassembly planning is an ongoing research field which also gains interest, through the assembly-by-disassembly approach, from the well-studied area of assembly planning. The majority of reviewed publications in this area use undirected and directed graphs as well as hypergraphs to represent the product structure [6–14]. Other approaches are based on petri nets [15], description logic and object-oriented models [16], and more recent approaches are based on ontologies [17–21]. Most works focus on sequence generation and thus lack in providing metadata for the human-robot

interaction. For our usage, we have decided to develop an easy-to-use, light but comprehensive model to represent the product structure. After careful consideration, we have chosen the object-orientated approach based on its advantages of information encapsulation, easy implementation, and extension. In addition, we can create other structures from the object-orientated model. We now explain the two fundamental classes of which our product model is composed.

A. *Part Classes*

In contrast to the works in [6] and [9], we do not tie functional parts to connections; instead, we clearly separate parts and connections into our fundamental classes. To distinguish between different classes of parts that have different needs of information in the sense of disassembly, we use a flat taxonomy-like hierarchy of classes (see Figure 2). Each class inherited from the super class carries common and more specialized information in its attributes. A common attribute, for example, is the position and orientation of a part in reference to the main coordinate system of the assembly. More specialized information could be the thread length or the driver style of a screw class instance. An overview of the attributes we use in the super class is listed in the table of Figure 2. Whenever a component indicates some special needs of information in the disassembly process, we can create a new class or extend a similar one. Chiefly, this is necessary for fasteners or connection techniques, but it could also be used to model fluids or gases. While the removal processes of a joint can have parameters that depend on the attributes of a part, we uncouple this information from the process. For example, the thread length and the thread pitch of a screw define the number of twists until the screw is loose. In this sense, it is a great advantage to use variable process parameters and to uncouple part-dependent facts from processes. Furthermore, we implemented state descriptions and qualifiers for each part class to represent each class's possible and current condition. This could be used to mark a part that it is damaged or missing. Beyond representing several parts, it is essential to model subassemblies. Mostly, it is not advisable to strip down a subassembly in another assembly. Typically, it is better to fractionalize subassemblies after their removal from the main assembly. So treating a subassembly as a special part class is beneficial in the manner that subassemblies gets completely removed from the main assembly and then further dismantled. Also considered is information for the further treatment of the parts. In a remanufacturing process, we could be interested in adding information to the parts about a test and a rework process a part has to pass to get back into production. For appropriate recycling, the material is an important attribute.

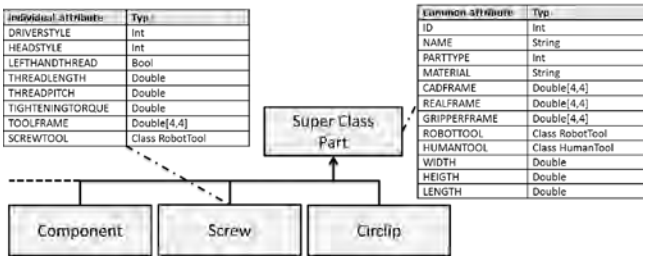


Fig. 2 The inheritance hierarchy of part classes with details of common and individual attributes.

This approach to equip the robot with the necessary information looks plausible, but it also means a substantial amount of knowledge engineering. Some of this information is already stored in today's CAD systems in a form of standard part libraries and could be used as a source of information for the product data model. Some information needs to be consolidated and made available in a clearly structured manner that is readable for humans and machines.

B. *Connection Classes*

Similarly to the part classes, we build hierarchies with connection classes (see Figure 3). Some common attributes are also illustrated in the table of Figure 3. A connection defines what kind of liaison is between two or among more parts and could be of any type. From a face-to-face contact, a weld joint or a magnetic attraction, any kind of liaison can be designed and implemented. The real power of the different connection classes is that they describe a formal process. If we can describe a certain state and objects in such a process, then we have a situation. Knowing the situation gives the assistant the ability to link commands dynamically with context information. In Section 5, we describe a formal process definition that makes use of a state representation and a set of actions, which could be performed by the acting agents (e.g., the user and the robot). We use this formal process description in detailed planning (Section 5) to find the best sequence of actions to remove the bound. As we see more clearly in Section 4.1, we can represent the product structure in a symbolical (machine-readable) manner by describing the parts and connections of an assembly.

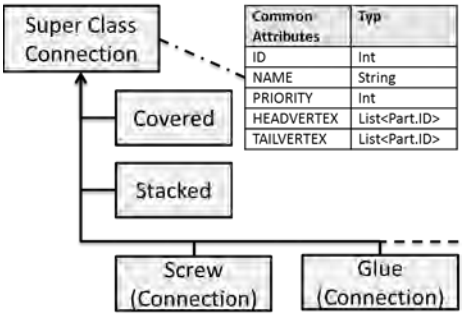


Fig. 3 Inheritance hierarchy of the connection class. The ordering implied the disassembly priority of each class from low to high. Also illustrated are the common attributes of the super class.

For the efficient manual design of these connections, we integrated functions into the NX 10 CAD system, which also generates the instances of the part classes during the connection creation. In future work, the product model will be automatically created by the product configurator. The format used to store the part and connection instances for a product model will be based on the XML format and be transferred via RFID directly from the assembly or via web services to the software agent.

IV. COARSE DISASSEMBLY PLANNING

Coarse disassembly planning reduces the amount of process definition done by the user to a significant level. While the disassembly task can vary among corrective maintenance, remanufacturing and recycling, we have to be able to create substructures of the complete disassembly for the exchange of wear and tear parts and target-oriented plans for the removal of valuable components. We also have to generate plans for the full dismantling of the product for recycling purposes. To achieve this, the disassembly task is communicated to the agent through the product model and the part to be removed. In this form, we are able to determine a plan for the removal of valuable or broken parts and, through selecting the root component of the assembly, the complete disassembly of a product. In the reviewed works [6–21] on assembly or disassembly planning, the main focus was on finding a sequence in which the parts could be added or removed to create a complete assembly or disassembly. It was not considered, to build substructures of assemblies, or to do further task and process planning. The used techniques vary depending on how the product structure was modeled. Most techniques used on graphs are mathematical and based on the adjacency matrix [11-13]. The inference is used by ontologies and description logic approaches [17–21], and rule-based systems use forward or backward chaining. Also found in the reviewed papers are applications of fuzzy logic and genetic algorithms [1]. We have decided to use the simple but powerful approach of topological sorting and do further task and process planning. We build the coarse plan in five steps. The first step is to create the product graph from the product model. In the second step, we create the minimal graph with the selected part as "root". Then we create a sequence of disassembly steps and tasks through topological sorting in the third step. In the fourth step, we assign the means of production to each task. In the last step, the user has to define the type of assistance we want in a certain task. Each step is shortly discussed as follows:

A. Creating the Product Graph

We represented the product structure internally through a directed graph **G**(**V**, **E**), which consists of a set of vertices **V** and edges **E**. A vertex **v** represents a part instance, and an edge $\mathbf{e}(\mathbf{v_i}, \mathbf{v_j})$ represents a link between two parts with the direction from the head $\mathbf{v_i}$ to the tail-vertex $\mathbf{v_j}$. A connection instance may have several edges depending on the connection type to better depict the interactions between parts. The graph is automatically created by the parts and connections described in the product model. For a better understanding of this topic, we will explain the graph creation and the part and connection models from the previous topic. For further explanation, kindly compare the sectional view of an electrical drive (in Figure 4) and its product graph (in Figure 5). We now explain three different connection types, two part types, and how we represent them in the graph. First, we consider the bolted joint of the four hex-head screws (Number 1.1–1.4 in Figure 5) that link together the cover plate (2), the sealing (3) and the drive body (15). The four screws are all instances of the "screw" part class. The sealing, the cover plate, and the drive body are instances of the "component" part class. The bolted joint is symbolically described by only one instance from the "screw" connection class because the screws are identical, and they link the same parts. In the screw connection instance, we define the screws as head- vertices and the other components as tail-vertices. This connection is represented in the graph with sixteen edges, from each screw to any of the other three connected parts. The next connection we have to specify is the cover plate, which is lying on the sealing. We represent it by an instance of the "stacked" connection class. This connection is represented by one edge from the cover plate to the sealing.

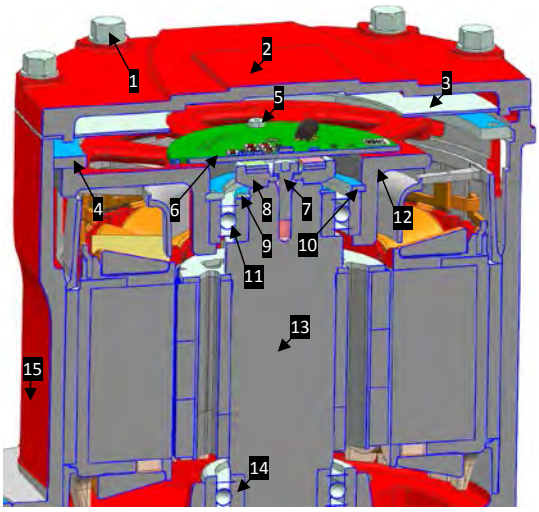


Fig. 4 A cross-sectional view of an electrical drive with numbered parts.

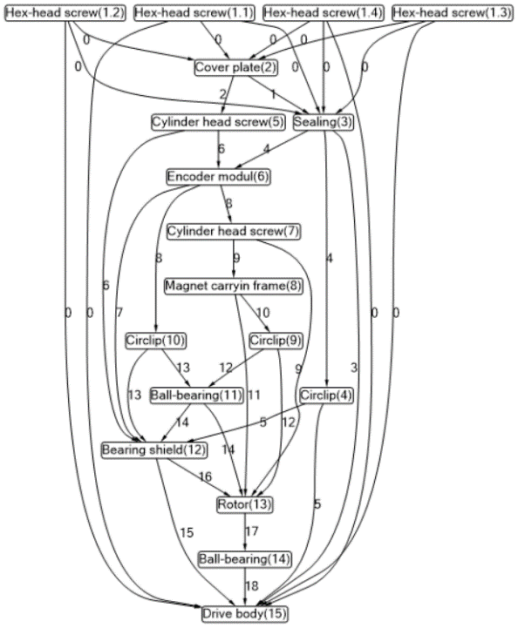


Fig. 5 A product graph of the electrical drive of Figure 4. The numbers on the edges are the IDs of connection instances.

An instance of the "covered" connection is used to describe the situation that one part avoids access to another part without any physical interaction. This case applies to the cover plate and the cylinder head screw (5). The covered instance is also illustrated on the graph by one edge from the cover plate to the screw.

B. Disassembly Planning

The part to be disassembled defines the start node of a breadth-first algorithm, which adds all parent nodes and corresponding edges to a new subgraph. Through topological sorting of the subgraph, we then create the disassembly sequence. In each step of the sequence, we identify part nodes that have no ingoing edges and could be removed. Through the outgoing edges of each part node, we get related connection instances. Then, a new task is created with the connection and part instances (see Figure 6), thereby taking care that parts that belong to the same connection instance are grouped together in the same task (in the current disassembly step). Moreover, if a part is attached to more than one connection, then we sort the connections by their disassembly priority in the task. The disassembly priority describes which connection of a set of liaisons has to threaten first. For example, if part A lies on part B and between A and B is a glue, then it is better to warm up the parts until the glue loses its strength and then remove part A. Because multiple parts and connections could be removed independently in a disassembly step, we can have multiple parallel executable tasks. If all removable parts are handled in one disassembly step, we delete the nodes and edges and create a new step. This process repeats until the subgraph has no more nodes.

C. Assigning the Means of Production

Especially in remanufacturing processes, we have to separate unequal, and usable from unusable, parts for their further treatment. To do so, we have to store the parts in different transport boxes and keep track of which part is in which box. Assigning the right box to a part, and vice versa, is one duty of the local "stock manager" module. The stock manager uses the width, height and length attributes of a part to find a suitable box in the local stock. The box is then assigned to the task and part. If the box is not available in the local stock, then the stock manager produces an order. While we model all process-involved objects in our environment, we have a class that represents boxes. A box object has attributes, such as its storage dimensions or their position and orientation in reference to the robot work frame, and qualifiers to represent their current state, e.g., if there is a box available, or it is ordered. Furthermore, the local stock manager determines the needed human and robot tools by accessing the attributes of a part with respect to the connection type. For example, a screw instance has three different tools: the tool for human use, such as a screwdriver, a robot tool for manipulating the part, usually a gripper, and a robot screw tool to loosen the joint. Like the box object, each tool is represented by an instance of a tool or robot tool class and has attributes, such as their positions, and state qualifiers. Furthermore, we assign the actors, e.g., the user and the robot, to the tasks to have all process relevant entities together.

D. Plan Manipulation and Assistance Definition

The sequence generated through the agent is a partially ordered plan. So there are different possible processes and the created one does not have to match a user's expectation. Furthermore, there could be other reasons that the user wants to rearrange or manipulate the found order. To rearrange the process, the user can move tasks back and forth in the disassembly sequence and manipulate the process in other methods by modifying the state qualifiers of the parts and objects to adapt the process to the current product state. More input is needed regarding the fact that the user has to define how he expects to be supported. Therefore, the user has to choose for each connection one type of assistance from a set of recommended assistance behaviors. At the moment, this is done by right clicking on the link with the computer mouse, which is a very unnatural method to communicate and flow-break. Using multimodal communication, through gestures and voice commands, to define or change the form of assistance would be a great advantage and is currently under investigation. To reduce the amount of assignment in this kind, future work will have to investigate if machine learning or case-based reasoning is able to predict a user's wish for assistance with tolerable accuracy.

To summarize the steps to this point, we have used a decomposition strategy [22] to split the complex disassembly problem into smaller, independent subproblems. The subproblem we consider is the removal of a certain type of connection, which is a process that we can generally describe formally and solve efficiently because of its smaller size. Furthermore, we can think of the coarse planning in a manner that it determines work that has to be done, without assigning it to the user or robot. The detailed planning in the next step solves the subproblem and assigns the work to the actors.

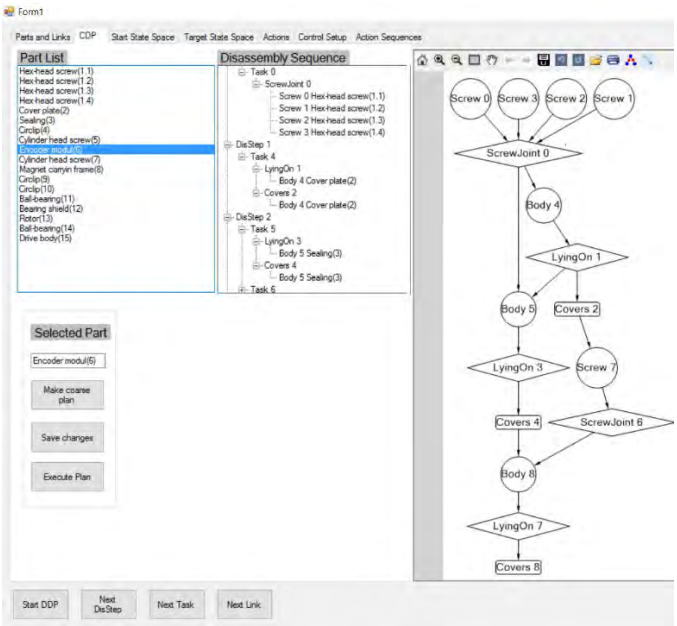


Fig. 6 The disassembly sequence computed by the agent for the removal of the incremental encoder of the electrical drive of Figure 4. The graph on the left illustrates how the different part and connection instances are connected.

V. THE DETAILED DISASSEMBLY PLANNING

In the ideal case, the manual removal of a connection takes place through a fixed sequence of actions. In disassembly workstations, we regularly find cases in which the sequence has to be adapted, probably with other actions, to produce the wanted output. Collaborative work also means that actions might be executed by one or another agent, in a dynamic manner that produces many variants of the process. The actors can also provide assistance to each other, for example, the user can change a robot effector so that there are even more possibilities. With this combinatory problem in mind, we have decided that the process should not be explicitly defined through the use of fixed finite state machines or petri nets. Instead, we use a search approach to find a suitable sequence of actions to solve our subproblem. To represent our issue as a search problem, we have first to decide on a vocabulary of conditions, objects, and actions. Then, we have to encode actions from our domain and define a problem instance by defining the initial and the target conditions. We now explain how we generate the initial state and target state, represent the actions and what algorithms we used to search for a solution by using an easy example task. We consider the loosening of a "stacked" connection, in which part A is simply lying on part B.

A. The Task State Description

The state we want to represent depends on the connection type and the involved objects, such as part(s), tool(s), boxes and the acting agents. Since we have already collected all these objects in a task description, we can create the initial state by merging all objects' state descriptions. Some objects' state values are predefined, for example, the *PartState*, *PartPos*, and *ConnectionState* (see Table 1), but could be manipulated by the user. The value of the *BoxState* and *BoxPos* are defined by the stock manager, depending on whether, if the box is ordered, it is full or ready to use. The robot states are defined by sending queries to the robot controller. The users *HandPos* state will be tracked, and the *HandState* value is estimated by the perception module. These state values represent literals, which formally represent the condition of the task at the beginning, the end, and in between through a state vector. In a state, we also save the parent's ID, the costs and the action type that created the state. For the initial state, these attributes are zero. The end condition of the task state is partially defined, through the default values of the *ConnectionState* and *PartPos* (see Table I). It is the users' choice to add other state dimensions and values to the end condition to define it more precisely.

B. Describing actions

Methods to describe or analyze manual and robot-automated tasks are well known, for example, Method-Time-Measurement (MTM) and Robot-Time-And-Motion (RTM). For the new type of collaborative work, there is quite a lack of methods. An adapted Method-Time-Measurement as process logic for cognitive automated assembly was mentioned in [23], which did not mention a human-robot interaction. Other studies such as [24] only link the basic Methods-Time-Measurement (MTM-1) system to equivalent robot actions without any collaborative actions. We have decided to describe actions only with the motion and end-effector elements of RTM, thus treating the human as a robot, and added a new element: collaborative actions. The removal of a stacked connection is described by the actions listed in Table II.

TABLE I
OBSERVED OBJECTS IN THE TASK. EACH OBJECT CAN HAVE SEVERAL DIMENSIONS AND STATE VALUES. THE GREEN VALUES INDICATE PREDEFINED STATES OF THE INITIAL TASK STATE. THE LITERALS IN RED ARE DEFAULT VALUES FOR THE TARGET STATE OF THE TASK.

Object type	State dimension	State values
Stacked Connection	Connection-State	isStacked, isDetached, isRemoved
Human	HandState	isEmpty, isNotEmpty
	HandPos	atUnknownPos, atPartPos, atBoxMagazinPos, atRobotGuidePos
Robot	RobotState	isUnknown, isIdle, isRunning, isGuided
	RobotPos	atUnknownPos, atHomePos, atPartPos, atBoxMagazinPos, atRobotToolMagazinPos,
RobotTool	RobotToolState	isUnknown, isOpen, isClosed
	RobotToolPos	atUnknownPos, atRobotFlange, atRobotToolMagazinPos
Part	PartState	isOK, isNOK,
	PartPos	atUnkownPos, atPartPos, atGripper, atHand, atBox
Box	BoxState	isAvailable, isOrdered, isFull
	BoxPos	atUnknownPos, atBoxMagazinPos

Each action has a precondition and transmission vector, which describes when the action is executable and how it affects the state. Each action defines which program on the robot controller is called when the action gets executed. The attribute list of an action is used to parametrize the robot program with the needed parameters. For example, the actions *GotoPartPos* and *GotoBoxMagazinPos* in Table 2 have the position of the part and box in the actions parameter list. Both actions call the same robot program, which only moves from its current position to the "send" part or box position. Furthermore, actions have a cost value that is dynamically assigned with respect to the user's choice of assistance. If an action is part of the assistance the user wants, it gets a lower cost assigned. In this example, we have mentioned three different robot behaviors. The *Manual* behavior defines only human actions and, therefore, manual work. *Automatic* means that the robot works autonomously. The *Collaborative* form describes that the user guides the robot to the unknown part position, and the robot can then grasp the part and put it in the transport box. We use this approach to provide to the users a set of known and predictable robot behaviors that stay adaptive to allow necessary modifications.

B. Searching for solutions

Although the problem of removing part A, which is lying on part B, does not look very complicated, we have to be aware of the combinative size of the problem. In a naive breadth-first search approach, with a branching factor of 8 and the minimal

process depth of 4, we have 8^4 (4,096) different paths to compute.

TABLE II
ALL ACTIONS THAT COULD TAKE PLACE IN THE PROCESS OF REMOVING THE STACKED CONNECTION. THE DIFFERENT ASSISTANCE OPPORTUNITIES AND THE CORRESPONDING WEIGHTING OF THE ACTION ARE INDICATED BY L FOR LOW AND H FOR HIGH ACTION COSTS.

Actor	Type	Action	Manual	Automatic	Collaborative
Human	Motion Element	GotoPartPos	L	H	H
		GotoBoxMagazinPos	L	H	H
		GotoRobotGuidePos	H	H	L
	End-effector Element	GrabPart	L	H	H
		ReleasePart	L	H	H
	Collaborative Element	GuideRobotToPartPos	H	H	L
Robot	Motion Element	GotoHomePos	H	H	H
		GotoPartPos	H	L	H
		GotoBoxMagazinPos	H	L	L
	End-effector Element	OpenGripper	H	H	H
		CloseGripper	H	H	H
		GrabPart	H	L	L
		ReleasePart	H	L	L
	Collaborative Element	GetGuidedToPartPos	H	H	L

A method of improving the algorithm is the use of an extended state list that stops us from extending paths on nodes we have already extended. Furthermore, we have implemented a policy that forbids the algorithm to use two motion elements in a row. This already gives us (for the problem mentioned) a good computational state space. To find the best path of actions, we have used a branch and bound search with extended list and the policy (see Figure 7). If the initial state could be a starting point of all assistance behaviors, we would get without dynamic weighting only one form of assistance. With dynamic weighting, we get the user selected form of assistance. This approach of assigning the work content to the agents might seem overloaded on the problem we mentioned, but it uses a generic approach that we can use for all other disassembly subproblems, such as screwing and so on. Also, our technique gives the assistance system a highly adaptable behavior. If the user leaves the workplace, his/her actions are not usable, and the only behavior is the *Automatic* one. If the part position is unknown in the *Automatic* mode, the next found assistance form is the *Collaborative* behavior (see Figure 8). In the case that the search algorithm might not find a suitable sequence of actions (solution), it can explain its reasoning process through the symbolic structure of the situation state space. Also, a great advantage of the search approach is that when we build these detail plans, task after task, we do not have to worry about plan merging, because at each task we create a new plan on the actual situation.

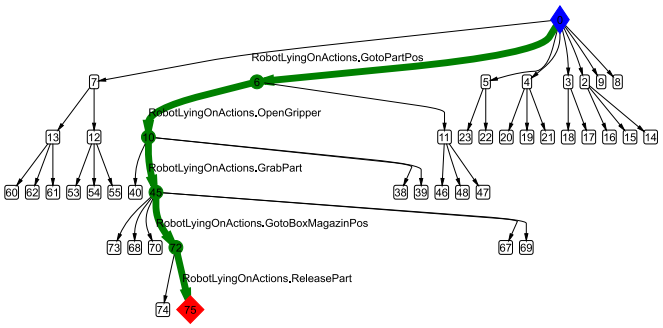


Fig. 7 A state space created by the search algorithm. From the initial state (blue diamond) and with the *Automatic* assistance behavior, a solution state (red diamond) was found, and the robot actions on the green path could be executed.

VI. EXECUTION OF THE DETAILED PLANS

The "Control" module of the agent is responsible for the execution of the robot programs, for the guidance of the user and for the synchronization of both. The robot programs are invoked and started through a TCP/IP socket connection from the Robot Controller. Then, the robot program starts and connects as a client to our agent to receive the actions parameters list items. At the moment, the disassembly is processed task by task and action after action. The synchronization of the user and robot actions rely on the user's pressing of the confirmation button. To improve the synchronization, we want to track the human hand movements to estimate if an action was carried out.

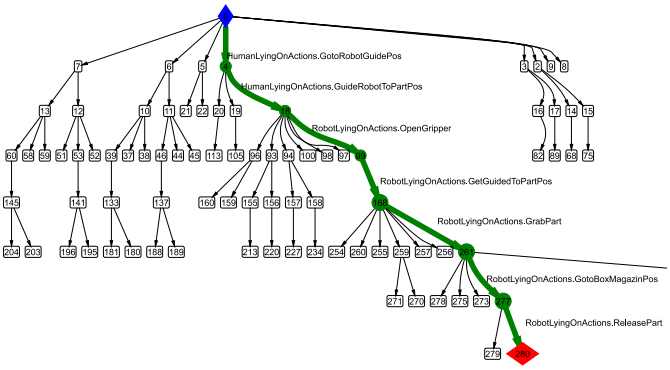


Fig. 8 This state space was created by the search algorithm also with the *Automatic* assistance selected, but through the fact that the part position is unknown, the solution consists of the actions of the *Collaborative* assistance behavior.

This lets us keep track of the user progress. For example, if we observe that the user moves his hand to the part position and then to the box position, we can assume that he has removed the part. This would be a very efficient way to synchronize the users' and robots' actions. Another important goal is to enable the parallel execution of tasks. The reduced robot motion in collaborative workplaces and the step-by-step workflow wastes much time, and the user gets bored waiting for his/her turn. So this is a very crucial skill for the assistant. Another ambition is to enable the user to give commands through a gesture or voice in full operation. This could be used to give simple commands, such as "stop," "open gripper" or "close gripper," or to change

the assistance behavior completely and force detailed replanning.

VII. Conclusion

In this paper, we have represented an approach of a robot-based disassembly assistant controlled by an informed software agent. We have discussed the need for a common workflow for a fluent, safe and purposeful assistance in collaborative disassembly. We have further described our approach to inform the agent with product models and our developed two-stage process of workflow planning. Therefore, we have explained the first planning step, which is based on the product graph, topological sorting, and task planning algorithms. We have also illustrated the second task-based planning step, which has focused on the refining of the workflow depending on the situation and the user-chosen assistance through a branch and bound search algorithm. In Section 6, we have discussed executing the robot assistant behavior. Finally, we can summarize that the information and methods we have provided to the robot assistance system enable higher autonomy to perform valuable assistance.

VIII. Acknowledgments

This research is conducted in a cooperation among the Company SEW EURODRIVE, the University of Luxembourg and the University of Applied Sciences Trier - Environmental Campus Birkenfeld.

References

[1] Wolfgang Gerke, *Technische Assistenzsysteme: Vom Industrieroboter zum Roboterassistenten*. Berlin: De Gruyter Oldenbourg, 2015.

[2] S. L. Kendal and M. Creen, *An Introduction to Knowledge Engineering*. London: Springer-Verlag London Limited, 2007.

[3] A. Perzylo *et al.*, "Intuitive instruction of industrial robots: Semantic process descriptions for small lot production," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2293–2300.

[4] B. Ludwig, *Planbasierte Mensch-Maschine-Interaktion in multimodalen Assistenzsystemen*. Berlin: Springer Vieweg, 2015.

[5] R. R. Murphy, *Introduction to AI robotics*. New Delhi: Prentice-Hall of India, 2004, c2000.

[6] J. M. Henrioud and A. Bourjault, "Computer aided assembly process planning," Laboratoire d'Automatique de Besancon, Besancon, France, 1992.

[7] T. de Fazio and D. Whitney, "Simplified generation of all mechanical assembly sequences," (en), *IEEE J. Robot. Automat.*, vol. 3, no. 6, pp. 640–658, 1987.

[8] G. Dini, F. Failli, and M. Santochi, "A disassembly planning software system for the optimization of recycling processes," (en), *Production Planning & Control*, vol. 12, no. 1, pp. 2–12, 2001.

[9] L. S. Homem de Mello and A. C. Sanderson, "A correct and complete algorithm for the generation of mechanical assembly sequences," in *Proceedings, 1989 International Conference on Robotics and Automation: A correct and complete algorithm for the generation of mechanical assembly sequences*, May. 1989, pp. 56–61.

[10] L. S. Homem de Mello and A. C. Sanderson, "AND/OR graph representation of assembly plans," (en), *IEEE Trans. Robot. Automat.*, vol. 6, no. 2, pp. 188–199, 1990.

[11] J. Yu, L. D. Xu, Z. Bi, and C. Wang, "Extended Interference Matrices for Exploded View of Assembly Planning," (en), *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 1, pp. 279–286, 2014.

[12] Y. Jyh-Cheng and L. Yi-Ming, "The Structure Representation for the Concurrent Analysis of Product Assembly and Disassembly," Department of Mechanical and Automation Engineering, 2005.

[13] N. S. Ong and Y. C. Wong, "Automatic Subassembly Detection from a Product Model for Disassembly Sequence Generation," (en), *The International Journal of Advanced Manufacturing Technology*, vol. 15, no. 6, pp. 425–431, 1999.

[14] H. J. Kim, S. Kernbaum, and G. Seliger, "Emulation-based control of a disassembly system for LCD monitors," *The International Journal of Advanced Manufacturing Technology*, vol. 40, pp. 383–392, 2009

[15] G. D. GRADI, "PETRI-NET Modelling of an Assembly Process System," Institute of Information Theory and Automation - Department of Adaptive Systems, Prague, Czech Republic

[16] Xu, C. Wang, Z. Bi, and J. Yu, "Object-Oriented Templates for Automated Assembly Planning of Complex Products," (en), *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 2, pp. 492–503, 2014.

[17] M. Merdan, W. Lepuschitz, T. Meurer, and M. Vincze, "Towards ontology-based automated disassembly systems," in *IECON 2010 - 36th Annual Conference of IEEE Industrial Electronics*, pp. 1392–1397.

[18] N. Lohse, H. Hirani, S. Ratchev, and M. Turitto, "An ontology for the definition and validation of assembly processes for evolvable assembly systems," in *(ISATP 2005). The 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing, 2005*, Jul. 2005, pp. 242–247.

[19] K.-Y. Kim, D. G. Manley, H. Yang, and B. O. Nnaji, "Ontology-based virtual assembly model for collaborative virtual prototyping and simulation," in *Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems, 2005*, pp. 251–258.

[20] S. Chen, J. Yi, H. Jiang, and X. Zhu, "Ontology and CBR based automated decision-making method for the disassembly of mechanical products," (en), *Advanced Engineering Informatics*, vol. 30, no. 3, pp. 564–584, 2016.

[21] S. Balakirsky, Z. Kootbally, C. Schlenoff, T. Kramer, and S. Gupta, "An industrial robotic knowledge representation for kit building applications," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, pp. 1365–1370.

[22] Q. Yang, *Intelligent planning: A decomposition and abstraction based approach*. Berlin: Springer, 1997.

[23] B. Britzke, *MTM in einer globalisierten Wirtschaft: Arbeitsprozesse systematisch gestalten und optimieren*. s.l.: mi-Wirtschaftsbuch, 2013.

[24] D. Schröter, P. Kuhlang, T. Finsterbusch, B. Kuhrke, and A. Verl, "Introducing Process Building Blocks for Designing Human Robot Interaction Work Systems and Calculating Accurate Cycle Times," (en), *Procedia CIRP*, vol. 44, pp. 216–221, 2016.

Human-Robot-Collaboration for dismantling processes

Sebastian Groß M.Eng., Prof. Dr.-Ing. Wolfgang Gerke
Department of Robotics and Automation
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* – Remanufacturing of used technical products allows saving of raw material and energy. The complex and mostly manual disassembly of the products prevents the application of remanufacturing in industry. The fully automated dismantling can not be achieved and the manual work causes high labour costs. Companies can profit by remanufacturing. Using human-robot collaboration for disassembly allows a certain degree of automation and provides the flexibility and knowledge of the human. This is a promising approach to economically realise a disassembly and remanufacturing of products which are difficult to disassemble or have a high variance.**

Index Terms – human-robot-interaction, human-robot-collaboration, robot-assisted disassembly, remanufacturing

I. Introduction

The nondestructive disassembly is one of the key tasks during a remanufacturing process. To reuse the parts of a product as spare parts or to use them in a new product, it is essential that they don't get damaged during the disassembly process. Otherwise only the raw materials of the part can be recycled. The advantage of a remanufacturing over a material recycling is that higher cost and energy savings can be achieved. That's because the cost and energy which are required to bring the raw material into the form of the end product are saved. This provides also an environmental benefit by saving the required amount of energy needed to execute the reprocessing of the raw material. The amount of energy needed for the refinement processes in materials is also saved. In addition to these potential opportunities for companies they also get forced by the government to proceed recycling and remanufacturing. One such law is the "Closed Substance Cycle Waste Management Act" by the European Union, which entered into force at the beginning of 2016.

Today remanufacturing is still just occasionally proceeded. One of the reasons behind that is the high staff deployment needed to execute the manually proceeded disassembly. Different approaches to reduce the regarded amount of staff through automation failed because of the increasing variance of products and the unknown product condition [1, 2, 3]. One promising approach to solve this problem is the use of a human-robot collaboration. Process steps which require the flexibility and judgement of the human can be proceeded by the human, with the assistance of the robot. Process steps that are automatable on the other hand can be completely proceeded by the robot. With this approach, the strengths of the human and of the robot can be combined. Using collaborative robots, it is also possible to improve the ergonomics for the human. Heavy parts can be handled by the robot and unhealthy postures can be avoided.

In this paper we will touch on the following subjects:

- Which role can the human occupy during the disassembly process within the human-robot interaction?
- Which information requires the human requires to deciding about the process flow in the context of disassembly?
- What are the ways on presenting this information to the human?
- How can the disassembly process plan be used to run modular structured programs on the robot controller to realize the required flexibility to disassemble different variances of the product? Through different combinations of this modular programs various disassembly tasks of different product variances can be executed instead of running fixed robot programs which allows only the disassembly of one product variance per program.

II. Human-robot-interaction in disassembly

There are different roles for the human in a human-robot interaction (HRI):

- Supervisor
- Operator
- Collaborator
- Co-operator

The supervisor monitors the robot and gives him instructions. The operator controls the robot. As collaborator or co-operator he works in both cases together with the robot on an overall goal. The difference between this two interaction-roles is that as a collaborator human and robot are also working together on sub goals which isn't the case during a co-operation [5]. To understand this difference, we will look at a real use case, the disassembly of a cooling water pump, which is shown in the picture below. To disassemble this pump different tasks must be done. For example, the unscrewing of bolts or the removal of different parts, which are both sub goals. The overall goal is

the complete disassembly of the cooling water pump. If human and robot working together at the unscrewing, the process is a human-robot collaboration. If the human does the unscrewing and the robot removes all other parts it is a human-robot cooperation, because human and robot don't work together on a sub goal.

Fig. 1 Cooling water pump which has different parts and links between them.

Which of these interaction-roles the human takes over depends on the case at hand. During the human-robot interaction of a certain process the human can take over one or more of this interaction roles.

During a human-robot interaction in a disassembly process the human can act in every task as a supervisor and as a collaborator or co-operator. Because the condition of the parts in a used product aren't known the human must decide how to execute the current disassembly task. The human acts here as supervisor. If the parts are in good condition and the robot can execute the current task automated the human can instruct the robot to do the task alone. That's an example of co-operation. Isn't the robot able the execute the task alone the human can instruct the robot to assist him. An example for this case is the unscrewing process. The robot is able to loosen the screw but can't thread the tool in the screw head. The collaboration here would be that the human guides the robot to screw and threads the tool in the screw head, afterwards the robot loosens the screw. In this example, the human acts as supervisor and collaborator. If a part is damaged and the robot can't help executing the task the human must execute the current task alone. But he can instruct the robot to provide him information or tools which are required to solve the present task.

To instruct the robot the human needs a communication channel between him and the robot. This channel shouldn't be unidirectional to enable the robot to provide the human information about the required steps to solve a certain task or to give information about the current process status. In the next chapter, we will look at different communication channels between human and robot in the context of disassembly.

III. COMMUNICATION BETWEEN HUMAN AND ROBOT

To realise a communication between human and robot different communication channels are available:

- Electrical channel (GUI, AR, VR)
- Mechanical communication
- Optical communication (Gesture recognition)
- Acoustical communication (Speech recognition)

It is possible to use one or more of this channels in one application. Which of this channels are most qualified in the present case depends on the information that needs to be communicated. To instruct the robot to execute a certain task all the listed communication-channels can be used. During a disassembly process the use of all these channels is conceivable.

As mentioned earlier the human acts as a supervisor during the disassembly process. To decide about the further process flow the human needs to know which actions must be executed in the next task. To provide the human this information one approach is to show him the part which gets removed in the next task and how it gets removed. This information can be presented through picture and video one a graphical-user-interface. The picture shows the part which gets removed in the next task and the videos shows how this part gets disconnected from the other parts of the product. Following to the decision making over the further process flow the human must communicate the chosen decision to the robot. Therefor different communication channels are possible. One possibility is to communicate this information through the graphical-user interface by pressing a button. Another possibility is the use of speech or gesture communication. By doing a certain gesture or spelling a keyword the information over the chosen decision gets communicated to the robot. During a disassembly process this both communication channels have different advantages and disadvantages. Using speech recognition, the human doesn't have to interrupt the process flow because he can contain the work he is doing with his hands. A problem with speech recognition are the loud noises in an industrial environment. It is possible that keywords don't get acoustically understand because of noise interference. That's not an issue when using gesture recognition. The disadvantage of gesture recognition is the addressed issue with interrupting the process flow. Another problem of gesture and speech recognition is that also unintended keywords and gesture can be recognised through the system. This can lead to unwanted actions. To solve this problem activating the gesture and speech recognition system only when it is need is an approach.

Also, the use of a mechanical communication channel during a disassembly process can be executed. One assistance function is handing the required tool to remove the current connection to the human. Therefor the robot grabs the tool with a force controlled gripper and hands it to the human, like shown in illustration below. To open the gripper afterwards a mechanical communication can be used. If the human grabs the tool and pushes or pulls it the robot detects the force peak and opens the gripper. To measure the force we use the internal force measurement, referred to the TCP, of the used Universal Robots UR3. The UR3 robot is a part of a demonstrator which we use to disassemble different products by using human-robot-collaboration. This demonstrator provides different ways of human-robot-interaction through different communication channels.

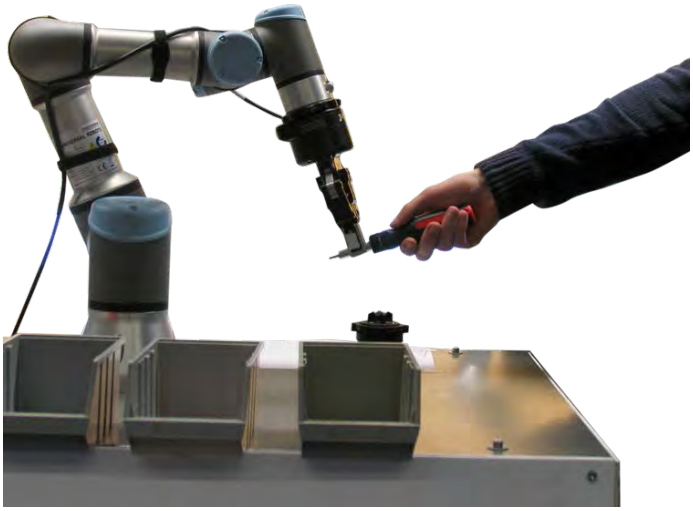


Fig. 2 Robot hands the human a tool which the human can take through a mechanical communication.

Opening the gripper by using other communication channels is also possible, but the use of this mechanical communication allows an instinctively interaction between human and robot. It replicates the way a human would give a tool to a co-worker.

IV. Process Execution

After the human has chosen how the next task will be executed appropriate robot programs must be executed. The structure of the robot program depends on the given task and on how this task should be executed. If the task is executed as a collaboration the robot program will be different as if the robot performs the task alone. Developing a fixed robot program for every possible disassembly workflow will end up in huge number of programs. For products that require a lot of tasks for their disassembly the creation of all this robot programs isn't executable. An automated process planning is needed. Another point, which results in the same issue is the growing product variance. To solve the problem of planning disassembly processes Jungbluth developed an agent-based control of the human-robot collaboration [5, 6]. His approach considers the different product variances and the different ways on how to execute a task within a human-robot interaction. Based on the product structure and the chosen way on how to execute the present task his software choses and executes a sequence of robot actions to assist. The product structure is described by graphs, like shown in the picture below. The boxes represent the single parts inside of the product and contain the related part names. The arrow between the boxes represent the links. To get a better understand of the product graph we refer to [5].

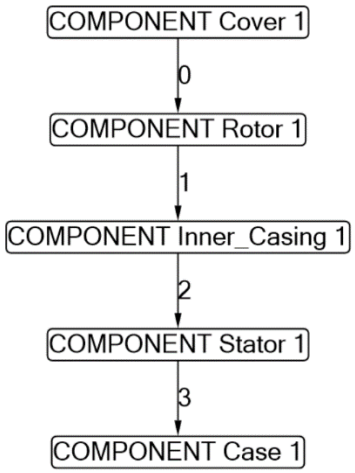


Fig. 3 Product structure with all the parts and links within a product.

After the human has decided how the disassembly of a link should be executed the planning of the required process starts. In the received process plan the different actions which every actor has to execute are listed. Actors can be humans, robots or other machines.

Based on this approach we developed the control of the robot, by using modular structured robot programs. Therefor we use one robot program for every possible action Jungbluths approach contains. Examples are:

- Gripping a part at its outside contour
- Gripping a part at its inside contour
- Moving to a given position
- Guiding the robot to a position
- Loosen a screw
- …

Through different combinations of these programs the required sequence of actions of a task can be executed.

The robot programs is loaded, started and parameterized through a software running on a PC. The communication between this software and the robot control system is based on an Ethernet TCP/IP protocol. This communication is also used to synchronise the actors (human, robot).

IV. Proof of Concept

To test and evaluate the described concept we developed a disassembly demonstrator for human-robot interaction. We use a Universal Robots UR3 and a gripper from ROBOTIQ to handle the different parts during the disassembly process. By using the infinite turning sixth axis of the robot and a screwing tool it is also possible that the robot loosens screws. The graphical user interface to present the mentioned disassembly information is shown in the picture below. The three buttons on the top are used to select how the present task should be executed. The following options are available:

- Human: Task executed completely manual
- Robot: Task executed completely automated (if possible)

- Collaboration: Task executed from both agents. (The collaboration process depends on the current link)

At the lower half, the picture and video regarding the current task are represented. In the lower right-hand corner a picture of a tool is shown. If this button is selected the robot hands the tool needed for the current tasks to the human.

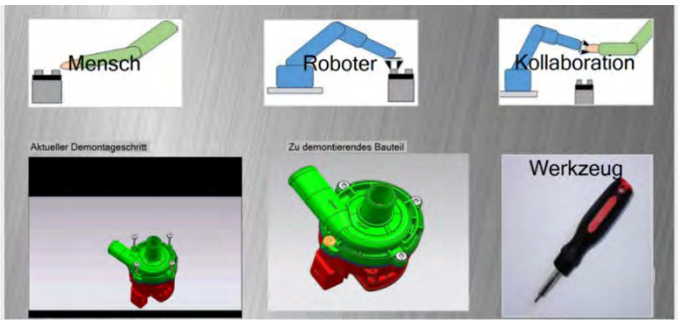


Fig 4. Graphical user interface to present the human disassembly information and to select the further process flow.

To test the application the disassembly of a cooling water pumper, shown in figure 1 was evaluated. The pump contains ten parts. To test the functionality of the application different ways to disassemble the pump were executed. The difference between the tested disassembly processes is the kind of collaboration during the disassembly. Therewith the planning of the disassembly process is tested by modular robot programs based on the software generated process plan.
With the developed demonstrator, it was possible to disassemble the cooling water pump through a human-robot-collaboration. Also the different process flows were successfully tested.
One issue is that the path of the robot isn't optimized in the respect of time, which results in a higher cycle time for the automated tasks. The reason for this is that the modular structured programs are developed to allow the disassembly of different products which don't necessary leads to a time optimized path for the present product. Also at the shift between the present robot program and the new robot program a certain waiting time is monitored. The new robot program must be loaded and started which interrupts the process flow and is one reason for this waiting time. In addition to that the Ethernet communication between robot and PC must be started and ended in every robot program which also causes the mentioned waiting time. This is necessary for the process synchronization and to be able to parametrize the robot programs.

V. Conclusion

The idea to control a human-robot collaboration during a disassembly process by executing modular structured robot programs based on a disassembly process plan was proofed through the test of a disassembly of a cooling water pump. During the test, every executed disassembly process flow achieved the goal of disassembling the pump. Because the human acts as a supervisor over the process flow it was possible to detect damaged product parts and adjust the further process flow so that a successful disassembly of the pump was possible.

In further work the economic efficiency of the collaboration during the disassembly of different product should be evaluated. To decrease the cycle time the planning of a time optimized path based on the CAD-model of the current product/robot cell would be an approach. To improve the mentioned waiting time during the communication between robot controller and the PC other ways of communication should be test. An additional decrease of the cycle time can be achieved through this optimization. The speech and gesture recognition is not implemented in the disassembly application yet and a work package for the future.

References

[1] Hohm, K.; Müller Hofstede, H.; Tolle, H.: Robot Assisted Disassembly of Electronic Devices. In: Proc. 2000 IEEE/RSJ, S. 1273 – 1278. IEEE, 2000

[2] Nave, M.: Beitrag zur automatisierten Demontage durch Optimierung des Trennprozesses von Schraubenverbindungen. Dissertation Technische Universität Dortmund, 2003

[3] BDU-Projekt: Untersuchung zur automatisierten Demontage von elektronischen Altgeräten und Bestimmung der Recyclingmöglichkeiten. Technische Universität Braunschweig –Institut für Werkzeugmaschinen und Fertigungstechnik, 2012

[4] Onnasch L.; Maier1 X.: Jürgensohn T.; Mensch-Roboter-Interaktion - Eine Taxonomie für alle Anwendungsfälle. In: baua: Fokus, Bundesanstalt für Arbeitsschutz und Arbeitsmedizin, 2016

[5] Jungbluth J.; Gerke W.; Plapper P.: Agentenbasierte Steuerung der Mensch-Roboter-Interaktion. In: atpedition Nr. 12 2016, S. 55-68, 2016

[6] Jungbluth J.; Gerke W.; Plapper P.: Demontage von Elektroantrieben mit Assistenzrobotern zum wirtschaftlichen Recycling. AALE 2016, 2016

Robot control based on human motion analysis with IMU measurements

Robin Pellois, Laura Joris and Olivier Brüls
Multibody and Mechatronic Systems Laboratory
University of Liège
Liège, Belgium
Email: {robin.pellois, ljoris, o.bruls}@ulg.ac.be

***Abstract*—The measurement of the operator motions can lead to innovation in the field of human-robot collaboration. Motion capture data can be used for robot programming by demonstration as well as controlling and adapting robot trajectories. To measure motion, IMUs represent an interesting alternative compared to other sensing technologies such as vision. In order to illustrate this strategy, a demonstrator has been created to command a robot by human motion. Two IMU sensor modules are set on the human arm and forearm. The orientation of the module with respect to the inertial frame is computed by the fusion of accelerometer, magnetometer and gyroscope data. This information coupled with a simple human arm kinematic model is used to compute the wrist trajectory. The accuracy of this measurement has to be quantified. For that purpose, the estimated trajectory based on the IMU measurement is compared to the trajectory measured using a reference 3D optoelectronic motion analysis system (Codamotion, Charnwood Dynamics Ltd) available in the Laboratory of Human Motion Analysis of the University of Liège.**

I. Introduction

Human-robot cooperation is an emerging paradigm which aims at combining the complementary skills of a human operator and an industrial robot in order to achieve complex tasks. The human operator has superior ability to analyse a situation, make a decision, plan the action and coordinate the motion in complex and unstructured environments. On the other hand, an industrial robot can be extremely performant in terms of motion accuracy, reproductibility, reactivity and load carrying capacity. Improvements in this field yield to many new situations. Human and robot can now share the same work space, the robot can see the human, the human can manipulate the robot and other kind of interaction are possible. This new field of robotic also bring new programming methods. In their survey [1], Biggs and MacDonald put forward manual programming against automatic programming. The first one, already implemented on commercalized robots, involves direct modifications of the robot program by the human. The second one, still at the developpment step, consists of using information from the operator and/or the environnement in order to make the robot modify his program by itself. In that way programming robot becomes easier, faster, more intuitive and no technical skills are necessary.

Inside the category of the automated robot programming, one of the most common field is Programming by Demonstration (PbD). The source of inspiration of this field takes place in the human-human interaction. When someone explains something to someone else, it could use different ways of communication, by verbal explanations, by showing, by touching, by using appropriate voice intonation... The same ways of communication can be found in the PbD field. The ”touching” interaction already exists, the robot is moved by the operator, motions or positions are recorded and reproduced. An example is the robot Sawyer (Rethink Robotics) wich can be handled by the human, positions are recorded, making programming very intuitive. The robot Pepper (Aldebaran) also master the game ”bal-in-the-cup” after a machine learning process initiated by a demonstration where the human guided the robot hand. The human can speak to the robot either to give instructions or to correct a movement where the robot tries to interprete cues in the human speech. Some research has been done in this field as the work of Lauria et al. [2]. They manage to control a mobile robot by natural speech. It is also possible to ”show” to the robot what to do. The robot acquires data from the human who is executing the task. After a processing step, the robot is able to reproduce the task.

In the problem of showing to the robot, the first question to ask is how to acquire information about human motion. Several technical solutions has been developped. It is possible to distinguish 3 types of solutions based on the acquisition system. One solution category includes all systems based on exoskeletons. Those devices measure directly the joint angle on human limbs. The work of Ijspeert et al. [3] can be mentionned as an exemple. A lot of work is also based on data gloves which measure the motion of the fingers. A second and common technical solution uses optoelectronic devices composed of one or several cameras, lasers, active or passive markers and/or infrared devices. Finally, a third option can be considered : Inertial Measure Unit (IMU). This technology takes already place in the robotic field as in the work of Prayudi et al. [4] or Liu et al. [5] to analyse the human motion in order to improve robot behaviour. Those sensors have some advantages compared to the others systems. They are wireless, easy to use and do not require any special environnement like the opto-electronic systems. They also are lighter, less invasive, smaller than exoskeletons. This work tries to demonstrate that IMUs can be used to measure the human motion in order to command and control an industrial robot arm.

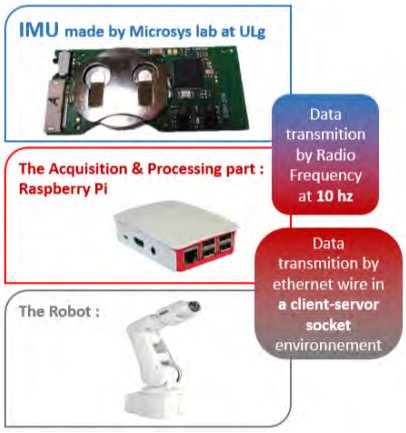


Fig. 1. Diagram of the demonstrator

In a first time the setup and each component are presented. Secondly this article presents the method used to measure the human motion explaining how data are exploited. Finally results are given and first considerations about the accuracy are discussed.

II. THE SET UP

The objective for the robot is to follow in real time the motion of the wrist of the operator. In order to simplify the problem for the first step, the orientation of the wrist is not taken in account and the shoulder of the operator is assumed to be fixed. Thus, the end-effector should follow the relative motion of the operator's wrist with respect to his shoulder. The setup is composed of 3 different parts : the sensors, the acquisition and processing system and the robot, organized as shown in the figure 1. The trajectory of the human wrist is computed with 2 sensor modules.

A. The sensors

The sensor module used in this work has been developped by the Microsys (Laurent et al. [6]) lab in the University of Liège. These 47x25x5 mm PCB are composed of an accelerometer and a gyroscope in the same component and a magnetometer in another component (fig. 2). The wireless module, powered by a 3 volt battery, sends informations by radio frenquency every 0.1 second. The sensors provide the following data defined in a reference frame attached to the sensor module (fig. 2):
The accelerometer measures the linear acceleration $\vec{a}$ and the gravity field $\vec{g}$.

$$\overrightarrow{Accelero} = \begin{pmatrix} a_x + g_x \\ a_y + g_y \\ a_z + g_z \end{pmatrix}$$

The magnetometer measures the magnetic field composed of the Earth magnetic field $\vec{B}$ pointing toward the north pole and some disturbances due to the environnement $\vec{n}$.

$$\overrightarrow{Magneto} = \begin{pmatrix} B_x + d_x \\ B_y + d_y \\ B_z + d_z \end{pmatrix}$$

The gyroscope measures the rotational speed of the module.

$$\overrightarrow{Gyro} = \vec{\omega} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

B. The Acquisition and Processing System

This system receives data from sensors, processes them and finally sends the displacement command to the robot. These functions are managed by a Raspberry Pi 3 which has the advantages of being easy to programme in C language and well documented. A radio antenna has been added to receive information from sensors. The Raspberry Pi 3 communicates with the robot by ethernet wire via socket connection. To command the robot, the Raspberry Pi sends just the displacement between two time steps.

C. The Robot

The robot applies this exact displacement relatively to an initial position. To get a smooth motion the speed of the robot has to be adapted accordingly to the length of the displacement. The robot used for the demonstrator is an ABB IRB120 with a reach of 580 mm and handle a payload of 3 kg. The Robot is programmed in RAPID language.

III. THE MOTION ESTIMATION

The first idea was to set one module on the wrist of the operator and estimate the position from a well known start position. But this idea involves a double integration of the accelerometer signal which would introduce an increasing error of the position over the time. In order to avoid this issue, several modules are used.

A. The References Frames

Two references frames are necessary and need to be defined. Firstly, the inertial frame, noted E is based on Earth features as shown in the figure 2. The z-axis is oriented along the gravity vector and pointing upward. The x-axis is perpendicular to $\vec{g}$ and oriented towards the north pole. Then the third axis can be deduced from the two others axis.
Secondly a local frame attached to the sensor module is defined according to the figure 3. During the human motion, this frame is constantly moving with respect to the inertial frame. At the step time n, the local frame will be noted S_{ni}, where i represents the module number.

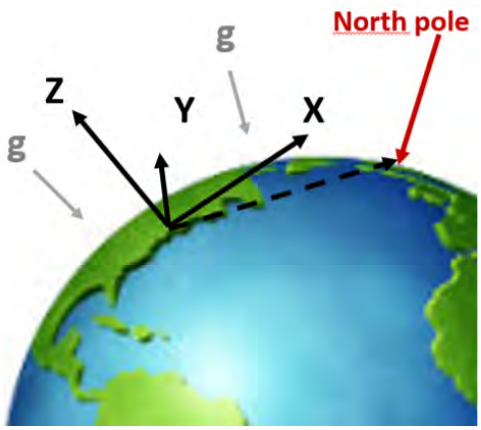


Fig. 2. The inertial frame

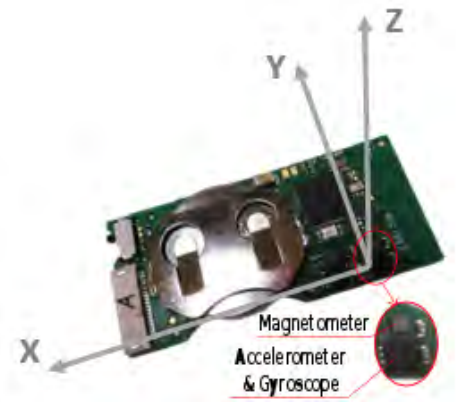


Fig. 3. The sensor module and the local frame

B. The Human Arm model

To be able to compute the human wrist trajectory, a model of the human arm is necessary. The chosen model is simply composed of 2 segments for the arm and forearm and 2 spherical joints for the elbow and the shoulder as shown on figure 4. The length of each segment has been measured directly on the operator. The objective is to compute the wrist trajectory with respect to the inertial frame centered in the shoulder since it is considered fixed. According to the human arm model the wrist trajectory is equivalent to $\overrightarrow{AC}(t)_E$ which can be expressed as follows :

$$\overrightarrow{AC}(t)_E = \overrightarrow{AB}(t)_E + \overrightarrow{BC}(t)_E$$

$$\overrightarrow{AC}(t)_E = R^E_{S_{n_1}} \times \overrightarrow{AB}(t)_{S_{n\,1}} + R^E_{S_{n_2}} \times \overrightarrow{BC}(t)_{S_{n\,2}}$$

with $R^E_{S_{n_1}}$ the rotationnal matrix allowing the changement between the local frame $S_{n\,1}$ attached to the module 1 and the inertial frame E. $R^E_{S_{n_2}}$ is the same rotational matrix related to the second module.
Furthermore the expression of $\overrightarrow{AB}(t)_{S_{n\,1}}$ and $\overrightarrow{BC}(t)_{S_{n\,2}}$ only depends on how the modules are set on each segment of the human arm. By consequence, the problem boils down to estimate the rotationnal matrix R^E_{Sn} for each sensor.

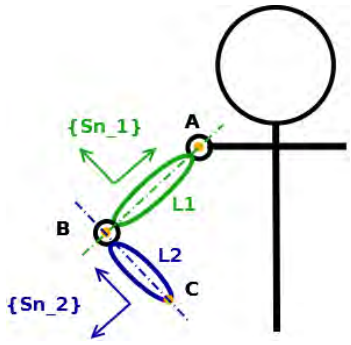


Fig. 4. The human arm model

C. The Rotationnal Matrix

Two methods are now presented to compute the rotational matrix $R^E_{S_n}$, one using the accelerometer and the magnetometer data and another one using gyroscope data.

1) Accelerometer and magnetometer data based method: This method is only applicable in phases during which the linear acceleration of the arm is negligible compared to the gravity. Under this condition the gravity vector is known in both inertial and local frames which leads to a first set of 3 equations used to calculate a part of the rotational matrix.

The same idea can be applied to the magnetometer. The vector $\overrightarrow{Mag}$ is the measure from the magnetometer express with respect to the local frame S_n and is equivalent to the x-axis in the inertial frame. This lead to 3 more independent equations.

Finally, using the orthogonality property of a rotational matrix, 3 other independent equations can be extracted. The complete matrix $R^E_{S_n}$ can then be evaluated by solving a system of 9 equations.

2) Gyroscope data based method: The data from the gyroscope $\vec{\omega}_{n\,S_n}$ represent the rotational speed at the moment n. By integration on the duration of the time step, the orientation of the frame S_n at the moment n with respect to the frame S_{n-1} is known. Applying this equation at every time step enables to compute the rotational matrix $R^{S_{n2}}_{S_{n\,1}}$ for any moment n_1 and n_2. Finally, using the first method based on accelerometer and magnetometer data to initialised the process enables to compute any $R^E_{S_n}$.

3) Combination of both methods: Both methods have advantages and drawbacks. The first one is sensitive to noise from the magnetometer and has to be rejected if the sensor undergos a too high linear acceleration. But this method enables to compute independently the rotational matrix from any S_n frame at any time step n to the inertial frame E.
On the other side, the method based on gyroscope data has the advantage of not being sensitive to the environnemental disturbances and is not limited by the motion of the arm itself. But the signal is integrated during the process which involve a

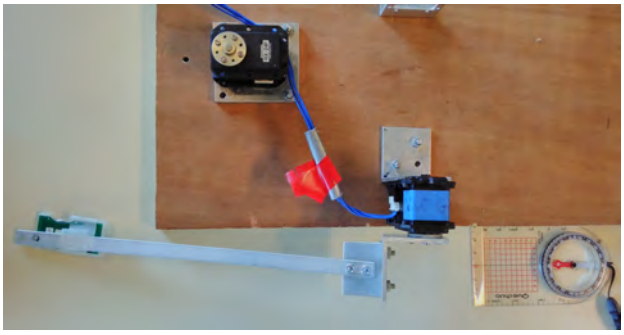


Fig. 5. Experimental setup

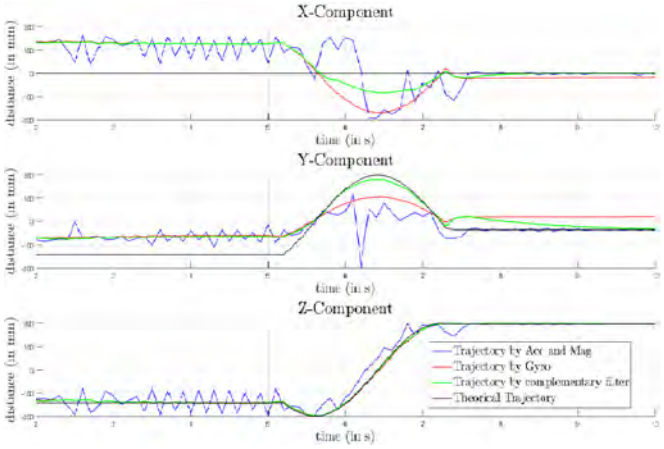


Fig. 6. Components of a circular trajectory measure by 3 differents methods.

drift over time. This phenomenon makes this method useless for the long time intervals.

A solution is to mix both methods to compute a rotational matrix. A complementarity filter [7] has been implemented into the algorithm. This filter mixes the results of both methods at a certain percentage. However, the filter is not applied directly on the rotational matrix but on the vector representing the angle-axis coordinates of the rotational matrix.

IV. Results

This filter bring a significant improvement to the trajectory. As an illustration of this improvement, the trajectory computed by the filter has been compared to the trajectory from the gyroscope data and from the accelerometer and the magnetometer data. One sensor module has been set on a 200 mm aluminium bar attached to a servomotor wich impose a circular trajectory from 135 deg to -100 deg in the yz-plane with respect to the inertial frame. A compass has been used for the orientation of the setup. The figure 5 describes the experimental setup. A static period before and after the circular motion has been observed. The figure 4 shows the results of each component of the trajectory. The black curve is the reference trajectory. First the x-component should be zero since the movement is in the yz-plane. This is due to disturbances in the magnetic field. Some considerations about this issue follow later in this article. Obviously the trajectory estimated by the first method (blue curve) is useless during the motion part. This chaotic trajectory is amplified by the vibration of the aluminium bar used for the setup. However, during the second motionless parts, the results are pretty accurate. On the other side, the red line, based on the gyscope data, give a curve following the theorical trajectory. But a shift appear quickly and is never caught up. The trajectory from the filter better fits the reference trajectory during the motion part. It also catches up the gap with the theorical trajectory in the motionless part. A significant example is the motionless part after the movement of the y-component where the filtered trajectory progressively follows the trajectory computed by the accelerometer and magnetometer data which fit with the reference trajectory.

At this point the robot is able to follow the wrist trajectory in real time. But a large imprecision is visible and attributed to the magnetometer noise. This high sensitivity of the magnetometer to magnetic disturbances was confirmed by additional dedicated experiments.

V. Conclusion

The IMU is an interesting technology to measure human motion in order to programme, command, control and collaborate with a robot. The method presented here uses two sensor modules, one for each segment of the human arm, in order to estimate the wrist trajectory. Each module is used to calculate the rotational matrix representing the transformation from the local frame to the inertial frame. Two methods are used to calculate this matrix, one based on accelerometer and magnetometer data and the other one based on gyroscope data. Both methods have advantages and drawbacks. The first one is accurate in really slow movements and is very sensitive to magnetic disturbances, the other one give a smooth trajectory but drift over time. A complementary filter is used to take avantage of both methods. The estimated trajectory of the wrist is then used to move a robot arm. An important trajectory default has been observed between the human wrist trajectory and the movement of the robot. It has been experimented that the magnetometer, easily disturbed by the environnement, is the main source of error in comparison to the accelerometer. The future work will focus on how to improve the measurement of the wrist trajectory by a better magnetometer signal or by another method not based on the magnetometer. In a second time, the next steps will be to integrate the position and the orientation of the wrist and then the hand.

Acknowledgment

This work was supported by the Interreg V programme of the Greater Region within the Robotix Academy project.

References

[1] G. Biggs and B. MacDonald, “A survey of robot programming systems.”
[2] E. K. Stanislao Lauria, Guido Bugman Theocharis Kyriacou, “Mobile robot programming using natural language.”
[3] S. S. Auke Jan Ijspeert, Jun Nakanishi, “Movement imitation with nonlinear dynamical systems in humanoid robots.”

[4] D. K. Iman Prayudi, Eun-Ho Seo, "Implementation of an inertial measurement unit based motion capture system," *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*.
[5] K. S. T. Liu, H. Utsunomiya Y. Inoue, "Synchronous imitation control for biped robot based on wearable human motion analysis system."
[6] F. A. P. Laurent, F. Dupont S. Stoukatch, "Ultra-low power integrated microsystems."
[7] P. Jan, "Reading a imu without kalman: The complementary filter."

PID and Biomimetic Variable Structure Path Tracking Control in Automated Surface Finishing Processes ⋆

S. Klecker * **P. Plapper** * **B. Hichri** *

* *Faculty of Science, Technology and Communication, University of Luxembourg, Luxembourg, (Tel: +3524666445654; e-mail: sophie.klecker@uni.lu).*

Abstract: This paper addresses freeform surface following control as one of the main challenges in automating surface finishing processes. Successive changes in constraints between the tool attached to the robotic manipulator and its surroundings are due to complex workpiece-geometries and result in a switched nonlinear system. The control problem of the latter is addressed by industrial state-of-the-art conventional PID control as well as by biomimetic variable structure control which are both applied to a benchmark path tracking problem as characteristically encountered in surface finishing.

Keywords: Tracking, Control of constrained systems, Control of switched systems, Sliding mode control, Adaptive control, Robots manipulators, Intelligent robotics.

1. INTRODUCTION

Freeform surfaces improve product structures by allowing enhanced performance at minimized size and mass, e.g. Lin et al. (2014). However, their geometric particularities complicate in general manufacturing operations operations, e.g. grinding, polishing in particular. The latter, represents the final step of the production process and is responsible for the final product quality and performance as its purpose is to bring the workpiece to the specified tolerance through the removal of small amounts of material. These operations are still predominantly performed manually. Consequently, these processes rely on human labour with their skills and experience. Not only does this fact lead to fluctuations in quality and lead time but also do the monotonous tasks in non-ergonomic working conditions lead to an ever increasing lack in motivated workforce. Due to their time- and cost-intensive nature: different studies suggest shares of up to 30-50% of the entire manufacturing time and up to 40% of the total cost, surface finishing processes are the bottleneck of the considered industry, e.g. Dieste et al. (2013); Feng and Su (2000); Lee et al. (2001); Lin et al. (2014); Pagilla and Yu (2001); Robertsson et al. (2006); Roswell et al. (2006); Wilbert et al. (2012); Yixu et al. (2012). In the context of producing parts with a consistent quality while minimizing time- as well as cost-related efforts and solving the labour problem, automation appears to be a promising alternative. Conventional tools, e.g. CNC machines, though successfully applied for operations involving simple, known workpieces, low geometrical accuracies and large material removal rates, reach their limits when confronted with complex freeform geometries, difficultly accessible areas, small product batch sizes with a high amount of variants and tasks requiring a certain level of intelligence. According to the International Federation of Robotics IFR only 2% of industrial robotic manipulators are performing tasks like grinding or cutting, e.g. Schneider et al. (2014). Indeed, processes asking for human skills remain an important challenge for the involved research community. Applied to surface finishing processes this means that while humans intuitively adapt to unknown surfaces and regulate contact forces, force control and trajectory tracking are challenging for robotic manipulators. The latter presents one of the first steps and one of the main challenges in surface finishing processes, e.g. Park et al. (2008). The accurate following of complex paths while keeping constant contact between robotic end-effector and workpiece is indeed a key-element for surface finishing processes. This paper addresses freeform surface following control as one of the main challenges in automating finishing processes. Successive changes in constraints between the tool attached to the robotic manipulator and its surroundings are due to complex workpiece-geometries and result in a switched nonlinear system. This in combination with highly non-linear, coupled as well as time-varying behavior and internal as well as external uncertainties deteriorates the stable performance of the system. The underlying control problem has triggered the interest of industrial and academic stakeholders. According to industrial state-of-the-art, PID (proportional-integral-derivative) controllers are the most widely implemented controllers in actual robotic manipulators. Simplicity in structure and design, robustness and low cost are counted among the assets of the linear fixed-gain PID controllers which can be found in the majority of commercially available robots. Despite these advantages and the fact that they have proven efficient in numerous industrial applications, conventional PID-controllers lack the necessary theoretical foundations and the flexibility or adaptability

⋆ Sponsor and financial support acknowledgment goes here. Paper titles should be written in uppercase and lowercase letters, not all uppercase.

to cope with nonlinear, strongly coupled, time-varying systems in the presence of disturbances and uncertainties, e.g. Adar and Kozan (2016); Cervantes and Alvarez-Ramirez (2001); Jenkins and Kurfess (1999); Kelly et al. (1997); Kuc and Han (2000); Lee et al. (2001); Prabhu and Garg (1996); Rocco (1996); Siciliano and Khatib (2008); Wai (2003); Wang (2009). One of the most prominent concepts claiming to remediate the PID-controllers shortcomings is variable structure control with its main representative being sliding mode control. The basic idea of this discontinuous nonlinear controller is to apply a high-frequency switching law to alter the system-dynamics by bringing the systems state trajectory onto a user-defined sliding surface and to maintain it there for subsequent times. In contrast to the industrially prevalent PID-controller, the sliding mode controller is well suited for the control of nonlinear, uncertain systems because of it being invariant to both internal and external disturbances. Next to these assets, sliding mode controllers present some important drawbacks like requiring a priori knowledge on system and constraints as well as the occurrence of undesirable vibrations known as chattering. In order to overcome the previously listed deficits, conventional control schemes have been extended, more and more with bio-inspired concepts. Recently biomimetic, i.e. bio-inspired engineering has celebrated resounding successes due to its innovative, relatively simple and intuitive solutions to highly complex problems,e.g. Huang (2013); Forestiero et al. (2013); Pedro and Kobayashi (2015); R et al. (2011). Biomimetic research aims for human-made solutions mimicking biological solutions, taking advantage of the strength of natural evolution. In this work, a controller which was developed by Lucas et al. (2004) based on the emotional learning behaviour of the mammalian brain: BELBIC (brain emotional learning based intelligent controller) is used. The source of inspiration for this controller was the emotional learning behaviour of the Amygdala-Orbitofrontal system of the human brain and more precisely, Moren and Balkenius computational model of the abstracted human limbic system comprised of the amygdala and the orbitofrontal cortex, e.g. Moren and Balkenius (2000); MorÉn (2001). The choice for this particular concept is based on its key advantages: improved learning capabilities, fast training, robustness, reduced computational burden and intuitiveness. In this sense, most of the highly appreciated assets of the industry-dominating PID-controller are preserved. The rest of this paper is organized as follows: following the present introduction, section 2 is dedicated to the concise description of the problem at stake. Section 3 describes the conventional PID-controller as found in a majority of industrial robotics settings, the fourth section introduces the more advanced BELBIC-Sliding Mode Controller. The simulation-results of both algorithms are shown in the next paragraph. A comparison of both algorithms alongside a discussion on future work is found in section 5 and followed by a conclusion for the proposed work.

2. PROBLEM DESCRIPTION

The dynamics of an n-link robotic manipulator with switching constraints can be expressed in Lagrange form:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau = u + Q_i \qquad (1)$$

with q, $\dot{q}$ *and* $\ddot{q}$ $\subset$ R^n being respectively the measured and desired link position, velocity and acceleration, $M(q)$ $\subset$ R^{nxn} the inertia matrix, $C(q,\dot{q})$ $\subset$ R^n, the centripetal/Coriolis vector, $G(q)$ $\subset$ R^n the gravitational torque-vector, $\tau \in$ R^n the bounded external disturbance, $u \in$ R^n the vector of applied input torque. Q_i is the global constraint force vector with $Q_i = J^T(q)D_i^T(\alpha)\lambda$ with $J(q)$ $\subset$ R^{nx6} the manipulator's Jacobian, $\lambda \in$ R^m the vector of Lagrange multipliers and $\frac{\delta\Phi_i(\alpha)}{\delta\alpha}$ the gradient of the task space constraints with $\Phi_i(\alpha)$ being the i^{th} kinematic constraint due to the surroundings of the system. $i = 1, 2, ..., m$ presents the constraint-index for the case of multiple switching constraints where is the total number of constraints. E.g. when $i = 2$ the second constraint is active while the other constraints are not.

The benchmark trajectory tracking case considered in this paper is taken from the context of surface finishing processes of freeform surfaces, i.e. the following of a succession of straight, concave or convex surfaces connected through jump-switches and including constant depths of cut. Similar motions can be found in for example industrial grinding processes.

3. PID CONTROLLER

Due to their ease of design, simple structure and robustness, conventional linear fixed-gain, model-free PID-controlled robotic manipulators have been predominant in industrial settings since their introduction in 1940 , e.g. Adar and Kozan (2016). Its obvious physical interpretation presents an additional asset of error-driven PID control: P, proportional compensates for the present error-value, i.e. the difference between current and desired state, I, integral accounts for the accumulated past value of error-signals, D, derivative presents the predictive effort for compensation of possible future error-values, e.g. Adar and Kozan (2016); Cervantes and Alvarez-Ramirez (2001); Jenkins and Kurfess (1999); Kelly et al. (1997); Kuc and Han (2000); Lee et al. (2001); Prabhu and Garg (1996); Rocco (1996); Siciliano and Khatib (2008); Wai (2003); Wang (2009).

The interest of the international research community in the application of PID-schemes in the control problem of robots following pre-defined paths is proven by a significant number of scientific publications in the field. While Xu and Qiao (2013) use the Lyapunov stability theory and simulation in the MATLAB-environment to show the value of conventional PID-controllers, Cervantes and Alvarez-Ramirez (2001) investigated PID-control in the trajectory tracking problem of robots equipped with revolute joints and Parra-Vega et al. (2003) elaborated on the structure, design and stability of closed-loop PID-controlled systems. Alvarez-Ramirez et al. (2000) combined conventional robotic PID-control with modern tools to introduce an improved PID-controller including a systematic tuning procedure. Pervozvanski and Freidovich (1999) analysed robots with PD, PID as well as PID-like controllers and concluded that these simple and robust controllers combined with well-conceived mechatronic designs can provide satisfactory results for several industrial use cases.

The parallel PID-controller structure is depicted in Fig. 1. Its output $u(t)$ can be expressed in the form of an equation:

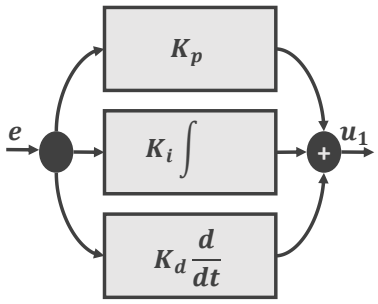


Fig.

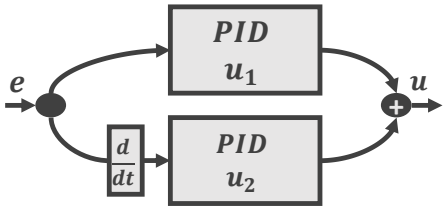


Fig. 2. Cascaded PID-controllers

$$u(t) = K_p e(t) + K_i \int e(t')dt' + K_d \frac{de(t)}{d(t)} \tag{2}$$

With K_p, K_i *and* K_d being the tuning parameters, i.e. the proportional, integral and derivative gains respectively. Further, with $e(t)$ the error-value, i.e. the difference between the desired and the actual value, t the time and t' the integration variable. Although conventional PID-controllers offer advantages ranging from simplicity to proven industrial applicability, they also do present some non-negligible disadvantages. Their constant fixed parameters as well as their linearity make it hard to cope with nonlinear, time-varying systems and the presence of uncertainties or disturbances. Their lack of flexible adaptability and the impossibility to increase gains arbitrarily due to actuator limitations, the occurrence of instabilities and noise sensitivity; e.g. Kuc and Han (2000); Siciliano and Khatib (2008) limit the application areas of PID-controllers. Finally, researchers and practitioners in the field agree that conventional model-free, linear, fixed-gain PID-controllers are not suited for freeform trajectory tracking in general and cutting or surface finishing processes in particular with highly nonlinear, coupled robotic systems, e.g. Adar and Kozan (2016); Cervantes and Alvarez-Ramirez (2001); Jenkins and Kurfess (1999); Kelly et al. (1997); Kuc and Han (2000); Lee et al. (2001); Prabhu and Garg (1996); Rocco (1996); Siciliano and Khatib (2008); Wai (2003); Wang (2009). A possible remedy for the conventional control-schemes drawbacks is the cascading of multiple PID-controllers which is applied in this work. Indeed, cascading several PID-controllers presents one of the only enhancing adaptations which does not affect or change the PID-control concept itself.

In this work, two parallel PID-controllers are combined in a parallel fashion (cf. Fig. 4) which results in a total control input of

$$u = u_1 + u_2 \tag{3}$$

With u_1 and u_2 as (2) where in $u_1 e = q - q_d$, the error-value of the position-signal while in $u_2 e = dotq - dotq_d$, the error-value of the derivative of the position-signal, the velocity-signal. q_d, $\dot{q}_d$ *and* $\ddot{q}_d \subset R^n$ are the desired link position, velocity and acceleration.

4. BELBIC-SMC-CONTROLLER

Sliding mode control, a form of variable structure control is one of the most widespread concepts chosen to overcome the PID-controllers drawbacks and to successfully control nonlinear systems in the presence of uncertainties and disturbances, e.g. to address the trajectory tracking problem of robotic manipulators. The concept involves a high-speed switching control law which is used to primarily guide the system's state trajectory onto a specified switching, i.e. sliding surface in the state space and to then maintain it on this surface for subsequent times. Thanks to its assets like robustness and invariance to internal as well as external disturbances, sliding-mode control has been extensively discussed in the scientific literature and has also been applied to trajectory tracking problems. (Sun, 1999) used sliding mode control for trajectory tracking combined with fuzzy systems theory as an adaptive estimator for unknown nonlinear robot dynamics. Ouyang et al. (2014) combined a conventional linear PD-controller with a nonlinear sliding mode-controller to respectively stabilize the system and reduce errors through compensation of disturbances or uncertainties. The suggested concept is applied to trajectory tracking control of a linear translational robotic system. Lee et al. (2001) developed a sliding mode-controller to reduce tracking errors in a polishing application. A comparison with the industrially widely implemented PD-controller was performed where the suggested sliding-mode controller outperformed the conventional PD-controller. Jasim and Plapper (2013) suggested to control a switched nonlinear system, i.e. a robotic manipulator with continuously switching constraints with an adaptive sliding mode controller. Despite the fact that sliding mode control has shown a satisfactory and robust tracking performance in a wide range of applications, two main drawbacks restrict its applicability. First, undesired oscillations known as chattering result from imperfect control switchings. To keep the system's state trajectory on the switching surface would require infinitely fast switching. However, in practice, switching is restricted to a finite frequency and an oscillation around the smooth sliding surface is observed. Second, a priori knowledge on system and constraints which is hardly ever present in reality, is required. Bio-inspired extensions with the aim to counterbalance the sliding mode-controller's downsides have been suggested. Bio-inspired, i.e. biomimetic engineering profits from the strength of natural evolution by mimicking solutions observed in nature when engineering man-made designs. A variety of those extensions to conventional sliding mode controllers inspired by nature have been suggested. A popular source of inspiration for control engineering problems is the human brain. Neural networks for example are artificial biomimetic networks copying the natural mammalian central nervous system. Composed of a significant number of simple but highly interconnected units, i.e. neurons working together to solve a particular problem loosely mimics the structure and functioning of

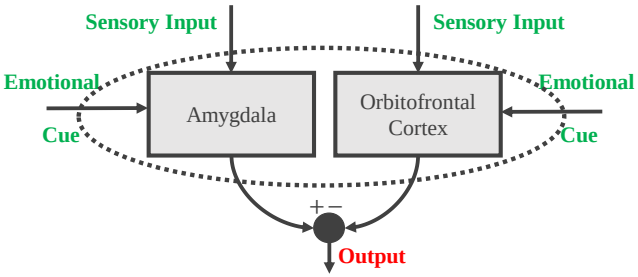


Fig. 3. Structure of a BELBIC-concept

the human brain. Since their introduction in the 194 these information processing paradigms are widely us Van Cuong and Nan (2016) addressed the trajectory tra ing problem in a two-link robot through an adaptive rad basis function neural network controller with a robu sliding mode compensator. Sun et al. (2011) present a combination of robust sliding mode control, neu network approximation and adaptive technique to so the trajectory tracking problem for robotic manipulat whose stability was proven by Lyapunov stability analys Tao et al. (2016) suggested a RBFNN-SMC concept, i a sliding mode control based on a radial basis functi neural network for the control of joint movements in an industrial robotic deburring application. The promising results included robustness as well as efficient position tracking. BELBIC, brain emotional learning based intelligent controller is based on the emotional learning behaviour of the Amygdala-Orbitofrontal system of the human brain. The computational model, e.g. MorÉn (2001); Moren and Balkenius (2000), of the abstracted human limbic system comprised of the amygdala and the orbitofrontal cortex forms the starting point for the controller, e.g. Lucas et al. (2004). In a structure, similar to that of a network, the amygdala, i.e. the actuator, links sensory stimuli to their respective emotional responses while the orbitofrontal cortex, i.e. the preventer replies to environmental changes by inhibiting the no longer useful connections. The BELBIC-scheme can be considered as a multiple input: sensory stimuli and emotional cues single output: action-generator system. Its structure can be illustrated in a graphical manner (cf. Fig. 3). Kim and Langari (2009) applied a BELBIC-controller to a mobile robot and satisfactory target tracking was obtained. Sharbafi et al. (2010) used a BELBIC-based controller in an omni-directional three-wheel holonomic robot. In this MIMO-system the authors showed that BELBIC decreases the control effort while following the desired movement. Jafarzadeh et al. (2008) showed the BELBICs superiority when compared to a PID-controller for path tracking of autonomous vehicles. Yi (2015) then showed that a combination of sliding mode and brain limbic system control outperforms both, a conventional and a fuzzy sliding mode-controller for trajectory tracking of robotic manipulators.

In this work a combination of sliding mode and BELBIC-control concepts is discussed, similar to the one presented in Klecker and Plapper (2016). The choice for these specific concepts is based on the key advantages discussed above. Sliding mode control is chosen because of its robustness and its suitability for the control of nonlinear, coupled systems in the presence of disturbances and uncertainties.

The addition of a BELBIC-element is justifiable by the aim to overcome the sliding mode control's deficiencies, i.e. to add some intelligence to the primary robust controller. BELBIC is chosen due to its enhanced learning capabilities, fast training and intuitive nature. With this suggested combination most of the highly appreciated assets of the industry-prevalent PID-controller are preserved.

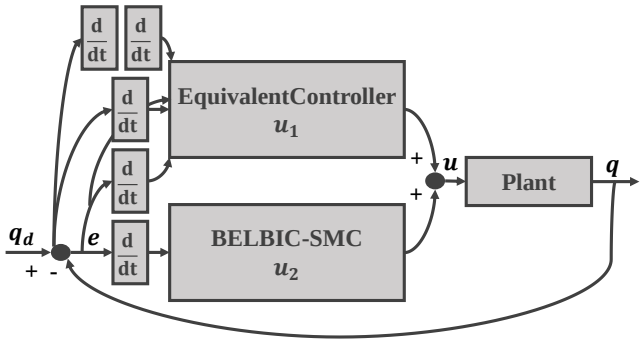


Fig. 4. Suggested control scheme

The first equivalent control term, based on weighted error-values can be expressed by the following equation

$$u_1 = M(q)\ddot{q}_e + C(q,\dot{q})\dot{q} + G(q) - M(q)(N_2 e + N_3 \dot{e}) \quad (4)$$

where $\ddot{q}_e = \ddot{q}_d - N_1\dot{e}$ with N_1, N_2 *and* N_3 independent, constant, positive-definite square diagonal gain matrices. The second uncertainty-compensating control term combines BELBIC and sliding mode control concepts. The interplay of emotions and environmental inputs represented by positional error-values as well as switching surface here defined as $s = \dot{e} + S_1 e$ with S_1 an independent, constant, positive-definite square diagonal gain matrix allows to control the systems future behaviour based on an adaptive learning rate. The output of the amygdala combines first, sensory input, i.e. environmental perception $s_i = \sigma s$ where σ a constant with second, connection weight g_a obeying the adaptation rule $\Delta g_a = \alpha(si\ max(0, ec - a))$ with α the constant learning rate and the rewarding emotional cue function $ec = (\varepsilon_1 - \varepsilon_2)(si - \epsilon u_2)$ where the first term represents the distance between desired and measured link position, velocity and with ϵ a constant. It can be expressed as:

The output of the orbitofrontal cortex combines first, sensory input with second, connection weight obeying the adaptation rule with the constant learning rate and can be expressed as:

$$\alpha - si\ g_a \quad (5)$$

The output of the orbitofrontal cortex combines first, sensory input with second, connection weight g_{oc} obeying the adaptation rule $\Delta g_{oc} = \beta(si(o - ec))$ with β the constant learning rate and can be expressed as

$$oc = si\ g_{oc} \quad (6)$$

The output finally is obtained by subtracting the inhibitory outputs of the orbitofrontal cortex from the amygdaloidal outputs

$$o = \alpha - oc \quad (7)$$

The second control term is represented by the following equation

$$u_2 = -C_1 M(q) s \; sats - C_2 M(q) \int_0^T (si^o o) dt \qquad (8)$$

With C_1, C_2 gain matrices, T the movement-time and $sats$ the saturation function obeying the rule: $if \mid s \mid > \Delta$: $sats = sign(s)$ $\quad else : sats = \frac{1}{\Delta} s$ with Δ the boundary layer.

5. RESULTS

For an objective comparison between conventional PID-control and the suggested BELBIC-SMC-control, both algorithms are simulated in the Matlab/Simulink-environment. A simple planar two-link rotary robotic manipulator with the following characteristics is considered:

$$\begin{aligned}
M_{11} &= (m_1 + M_2) l_1^2 \\
M_{12} &= m_2 l_1 l_2 (\sin q_1 \sin q_2 + \cos q_1 \cos q_2) \\
M_{21} &= m_2 l_1 l_2 (\sin q_1 \sin q_2 + \cos q_1 \cos q_2) \\
M_{22} &= m_2 l_2^2 \\
C_{11} &= 0 \\
C_{12} &= -m_2 l_1 l_2 (\cos q_1 \sin q_2 - \sin q_1 \cos q_2) \dot{q}_2 \\
C_{21} &= -m_2 l_1 l_2 (\sin q_1 \sin q_2 - \sin q_1 \cos q_2) \dot{q}_2 \\
C_{22} &= 0 \\
G_1 &= -(m_1 + m_2) l_1 g \sin q_1 \\
G_2 &= -m_2 l_2 g \sin q_2
\end{aligned} \qquad (9)$$

With the masses of link 1 and 2 $m_1 = m_2 = 1kg$, their link lengths $l_1 = l_2 = 1m$and gravitational acceleration $g = 9.8ms^-2$. The parameters for the algorithms: PID-controllers are Matlab-internal controller with $K_p = 100$, $K_i = 1$, $K_d = 50$; for the BELBIC-SMC-controller $N_1 = 10I_2$; $N_2 = 17I_2$; $N_3 = 8I_2$; $s_1 = I_2$; $\Delta = 0.05$; $\sigma = 0.45$; $\epsilon = 0.5$; $\alpha = 0.5$; $\beta = 0.6$; $C_1 = I_2$; $C_2 = 500I_2$. The tracking performance is tested for a system with switching constraints. The systems desired motion is depicted (Fig. 5).

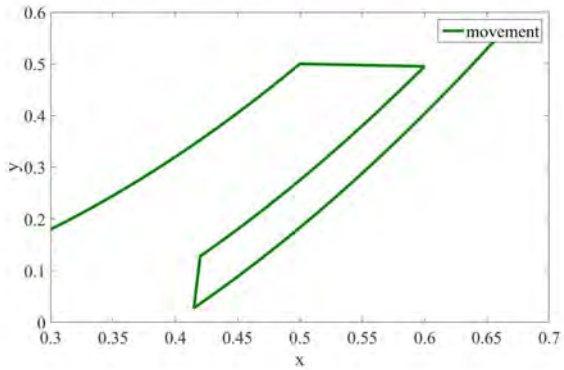


Fig. 5. Desired motion

The benchmark test case includes a succession of curved trajectories linked by jump-switches and including constant depths of cut. As an illustration of the obtained results position tracking of link 1 for both algorithms is shown (Fig. 6 and Fig. 7).

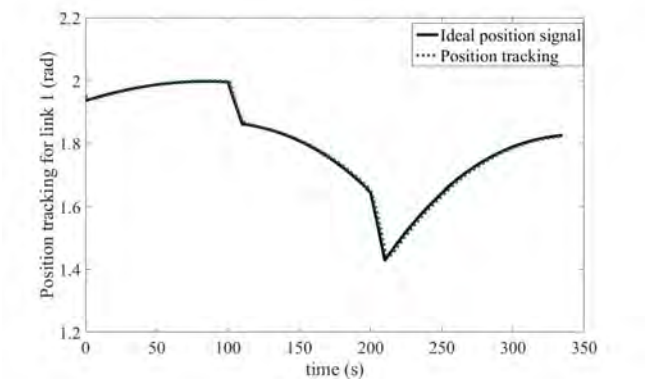


Fig. 6. Position tracking for link 1 with the BELBIC-SMC algorithm

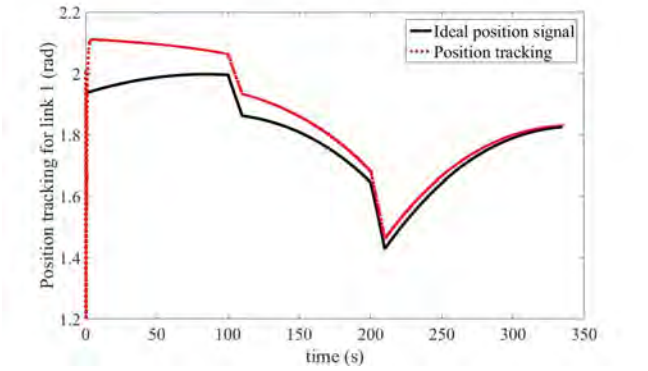


Fig. 7. Position tracking for link 1 with PID-control

6. DISCUSSION

The purpose of simulating the behaviour of both, the conventional cascaded PID-controller and the newly suggested biomimetic BELBIC-SMC-controller is to compare their performances and to investigate whether a switch from PID to more advanced control algorithms would be beneficial for industry. As can be seen from the results depicted in the previous section, the tracking performance of the BELBIC-SMC-controller outrivals the achievements of the PID-controller. This is especially true in the initial phase of the path following exercise. This fact, i.e. BELBIC-outperforming a PID-controller has also been presented by Ale Aghaee et al. (2012). The authors showed that compared to a traditional PID-controller, BELBIC was able to reduce oscillation-amplitudes and settling times in the regulation of the load frequency error in a two-area power system. Although the BELBIC-concept outperforms the conventional PID-controller, the latter still offers a decent tracking performance which is satisfactory for a range of industrial applications. For some processes requiring higher precision, e.g. surface finishing processes, however, it might be worth taking the effort to switch to a more advanced scheme. In contrast to the conventional PID-controller, the proposed BELBIC-SMC-controller is not a model-free controller. A problem that might occur when trying to implement it on an industrial robot is the required knowledge about the robot parameters. As in practice the robot kinematics and dynamics are seldom known, these parameters have to be estimated. The approximation of the robot dynamic parameters with

a neural network and the subsequent validation of the algorithm-performance on a commercially available robot are current work.

7. CONCLUSION

This paper asked the question whether a more advanced control scheme, in this case the suggested BELBIC-SMC-controller could outperform a conventional PID-controller which is still predominantly used in industrial settings. More precisely the trajectory tracking results were analyzed with the goal to achieve sufficient performance for the automation of surface finishing processes. The simulation showed promising results and current work is the experimental validation on an industrial robotic manipulator including the estimation of unknown robot dynamic parameters.

ACKNOWLEDGEMENTS

The publication is funded by the European Regional Development Fund (ERDF) within European Union's INTERREG V A-program Greater Region, project Robotix Academy.

REFERENCES

Adar, N. and Kozan, R. (2016). Comparison between real time pid and 2-dof pid controller for 6-dof robot arm. *Acta Physica Polonica A*, 130(1), 269–271.

Ale Aghaee, S., Lucas, C., and Amiri Zadeh, K. (2012). Applying brain emotional learning based intelligent controller (belbic) to multiple-area power systems. *Asian Journal of Control*, 14(6), 1580–1588.

Alvarez-Ramirez, J., Cervantes, I., and Kelly, R. (2000). Pid regulation of robot manipulators: stability and performance. *Systems & control letters*, 41(2), 73–83.

Cervantes, I. and Alvarez-Ramirez, J. (2001). On the pid tracking control of robot manipulators. *Systems & control letters*, 42(1), 37–46.

Dieste, J., Fernández, A., Roba, D., Gonzalvo, B., and Lucas, P. (2013). Automatic grinding and polishing using spherical robot. *Procedia Engineering*, 63, 938–946.

Feng, H.Y. and Su, N. (2000). Integrated tool path and feed rate optimization for the finishing machining of 3d plane surfaces. *International Journal of Machine Tools and Manufacture*, 40(11), 1557–1572.

Forestiero, A., Pizzuti, C., and Spezzano, G. (2013). A single pass algorithm for clustering evolving data streams based on swarm intelligence. *Data Mining and Knowledge Discovery*, 26(1), 1–26.

Huang, Z. (2013). A classification rules extraction algorithm base on fish swarm optimization. 31–33.

Jafarzadeh, S., Mirheidari, R., Motlagh, M.R.J., and Barkhordari, M. (2008). Designing pid and belbic controllers in path tracking problem. *International Journal of Computers Communications & Control*, 3, 343–348.

Jasim, I.F. and Plapper, P.W. (2013). Adaptive sliding mode control of switched constrained robotic manipulators. In *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, 305–310. IEEE.

Jenkins, H.E. and Kurfess, T.R. (1999). Adaptive pole-zero cancellation in grinding force control. *IEEE Transactions on control systems technology*, 7(3), 363–370.

Kelly, R., Santibanez, V., and Berghuis, H. (1997). Point-to-point robot control under actuator constraints. *Control Engineering Practice*, 5(11), 1555–1562.

Kim, C. and Langari, R. (2009). Target tracking control of a mobile robot using a brain limbic system based control strategy. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5059–5064. IEEE.

Klecker, S. and Plapper, P. (2016). Belbic-sliding mode control of robotic manipulators with uncertainties and switching constraints. In *International Mechanical Engineering Congress and Exposition*.

Kuc, T.Y. and Han, W.G. (2000). An adaptive pid learning control of robot manipulators. *Automatica*, 36(5), 717–725.

Lee, M., Go, S., Lee, M., Jun, C., Kim, D., Cha, K., and Ahn, J. (2001). A robust trajectory tracking control of a polishing robot system based on cam data. *Robotics and Computer-Integrated Manufacturing*, 17(1), 177–183.

Lin, W., Xu, P., Li, B., and Yang, X. (2014). Path planning of mechanical polishing process for freeform surface with a small polishing tool. *Robotics and Biomimetics*, 1(1), 1.

Lucas, C., Shahmirzadi, D., and Sheikholeslami, N. (2004). Introducing belbic: brain emotional learning based intelligent controller. *Intelligent Automation & Soft Computing*, 10(1), 11–21.

Moren, J. and Balkenius, C. (2000). A computational model of emotional learning in the amygdala. *From animals to animats*, 6, 115–124.

MorÉn, Christian Balkenius, J. (2001). Emotional learning: A computational model of the amygdala. *Cybernetics & Systems*, 32(6), 611–636.

Ouyang, P., Acob, J., and Pano, V. (2014). Pd with sliding mode control for trajectory tracking of robotic system. *Robotics and Computer-Integrated Manufacturing*, 30(2), 189–200.

Pagilla, P.R. and Yu, B. (2001). Adaptive control of robotic surface finishing processes. In *American Control Conference, 2001. Proceedings of the 2001*, volume 1, 630–635. IEEE.

Park, J., Kim, S.H., and Kim, S. (2008). Active compliant motion control for grinding robot. *IFAC Proceedings Volumes*, 41(2), 4285–4289.

Parra-Vega, V., Arimoto, S., Liu, Y.H., Hirzinger, G., and Akella, P. (2003). Dynamic sliding pid control for tracking of robot manipulators: theory and experiments. *IEEE Transactions on Robotics and Automation*, 19(6), 967–976.

Pedro, H.T. and Kobayashi, M.H. (2015). On biomimetic engineering design. *IEEE Potentials*, 34(2), 7–9.

Pervozvanski, A.A. and Freidovich, L.B. (1999). Robust stabilization of robotic manipulators by pid controllers. *Dynamics and Control*, 9(3), 203–222.

Prabhu, S.M. and Garg, D.P. (1996). Artificial neural network based robot control: An overview. *Journal of Intelligent and Robotic Systems*, 15(4), 333–365.

R, P.V., M, N.A., and J, R.M. (2011). Document clustering using hybrid aco-ts. *Int. J. of Recent Trends in Engineering and Technology*, 6(1), 167–172.

Robertsson, A., Olsson, T., Johansson, R., Blomdell, A., Nilsson, K., Haage, M., Lauwers, B., De Baerdemaeker, H., Brogardh, T., and Brantmark, H. (2006). Implementation of industrial robot force control case study: high

power stub grinding and deburring. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2743–2748. IEEE.

Rocco, P. (1996). Stability of pid control for industrial robot arms. *IEEE Transactions on Robotics and Automation*, 12(4), 606–614.

Roswell, A., Xi, F.J., and Liu, G. (2006). Modelling and analysis of contact stress for automated polishing. *International Journal of Machine Tools and Manufacture*, 46(3), 424–435.

Schneider, U., Drust, M., Ansaloni, M., Lehmann, C., Pellicciari, M., Leali, F., Gunnink, J.W., and Verl, A. (2014). Improving robotic machining accuracy through experimental error investigation and modular compensation. *The International Journal of Advanced Manufacturing Technology*, 1–13.

Sharbafi, M.A., Lucas, C., and Daneshvar, R. (2010). Motion control of omni-directional three-wheel robots by brain-emotional-learning-based intelligent controller. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6), 630–638.

Siciliano, B. and Khatib, O. (2008). *Springer handbook of robotics*. Springer Science & Business Media.

Sun, T., Pei, H., Pan, Y., Zhou, H., and Zhang, C. (2011). Neural network-based sliding mode adaptive control for robot manipulators. *Neurocomputing*, 74(14), 2377–2384.

Tao, Y., Zheng, J., and Lin, Y. (2016). A sliding mode control-based on a rbf neural network for deburring industry robotic systems.

Van Cuong, P. and Nan, W.Y. (2016). Adaptive trajectory tracking neural network control with robust compensator for robot manipulators. *Neural Computing and Applications*, 27(2), 525–536.

Wai, R.J. (2003). Tracking control based on neural network strategy for robot manipulator. *Neurocomputing*, 51, 425–445.

Wang, L. (2009). *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media.

Wilbert, A.D., Behrens, B., Dambon, O., and Klocke, F. (2012). Robot assisted manufacturing system for high gloss finishing of steel molds. In *International Conference on Intelligent Robotics and Applications*, 673–685. Springer.

Xu, J. and Qiao, L. (2013). Robust adaptive pid control of robot manipulator with bounded disturbances. *Mathematical Problems in Engineering*, 2013.

Yi, H. (2015). A sliding mode control using brain limbic system control strategy for a robotic manipulator.

Yixu, S., Hongbo, L., and Zehong, Y. (2012). An adaptive modeling method for a robot belt grinding process. *IEEE/ASME Transactions on Mechatronics*, 17(2), 309–317.

Feedforward control of vibrations in flexible and lightweight robots

Arthur Lismonde[1], Valentin Sonneville[2] and Olivier Brüls[1]

[1]*Department of Aerospace and Mechanical Engineering, University of Liège, Belgium, {alismonde,o.bruls}@ulg.ac.be*
[2]*Aerospace Engineering, University of Maryland, USA, vspsonn@umd.edu*

Abstract

Lightweight and flexible robots have a high potential in today tendency to use collaborative automation. Thanks to their reduced weight and increased compliance, such systems benefit from an intrinsic safety that reduce the risk of injury in case of unexpected collision. However, the controller of such system has to be carefully designed to deal properly with the flexible behavior of the links and joints.
This work focuses on the control of the flexible behavior in general 3D robotic manipulators. In particular, an innovative feedforward control command is developed to reduce vibrations in the robot during its motion. First, a finite element model of the robot is built using rigid bodies, flexible beam elements and kinematic joints elements [1]. Based on it, the inverse dynamics is solved using so-called stable inversion techniques [2]. These methods can deal with the non-minimum phase nature, i.e. unstable nature, of such flexible and nonlinear systems. Here, a constrained optimization formulation, introduced in [3] for 2D systems and extended here to 3D systems, is used to solve the inverse dynamics problem. In future work, this method could be implemented on a robotic testbed with a flexible end-link to test its performances.

1 Introduction

Robot manipulators are evolving in order to meet current needs regarding accuracy and safety. To improve the latter, lighweight and collaborative robot can be a good alternative. However, controlling lightweight and flexible structures is not an easy task as they can be subjected to vibration and elastic deformation issues. Such flexible manipulators are said to be underactuated since they potentially have an infinite number of degrees of freedom (dof) and a finite number of actuators. To reduce such flexibility issues, the controller of such manipulator has to be designed carefully. Feedback action can be implemented to compensate for vibrations see, e.g., [4]. A second possibility is to model such flexible multibody system (MBS) in order to compute an input feedforward control signal that results in a vibration-free motion of the robot. Both the feedforward and the feedback control methods can be combined to achieve robust performances as presented in [5, 6].
To perform an end-effector trajectory tracking task, an example of feedforward commands for the manipulator would be the torques of each of its joints. To find those inputs, the inverse dynamics of the MBS needs to be solved. In the case of a flexible system, some internal dynamics remains when the output trajectory is prescribed. The system is said to be non-minimum phase when this internal dynamics is unstable. If the inverse dynamics of a non-minimum phase system is simply solved using time integration algorithms or computed torques methods, the resulting input control can be unbounded. In order to obtain a bounded solution, a non-causal solution can be considered. A time domain inverse dynamics method is presented and tested for a linear system in [7]. For flexible nonlinear systems, a stable inversion method is presented in [2] and is applied in [8, 9]. An optimal control approach is proposed in [3] for 2D multibody systems. The present work extends this last method to solve the inverse dynamics of flexible 3D systems. The flexible MBS is modeled using nonlinear beam finite elements [10], rigid bodies and kinematic joints [1] formulated on the special Euclidean group $SE(3)$. The inverse dynamics is then stated as an optimization problem where the amplitude of the internal dynamics has to be minimized. The prescribed end-effector trajectory is defined as an additional servo constraint of the optimization problem.
Please note that the present paper is a summary of references [11, 12], please see the latter for more details.

2 Dynamic model of the flexible multibody system

The finite element formalism can be used to model the dynamics of flexible MBS including rigid and flexible bodies interconnected by kinematic joints [1, 10]. With

the special Euclidean group $SE(3)$ formalism, the position and orientation of each element composing the finite element mesh is represented as a 4×4 homogeneous transformation matrix $\mathbf{H}_I$ with a rotation $\mathbf{R}_I \in SO(3)$ and a position $\mathbf{p}_I \in \mathbb{R}^3$ component.

$$\mathbf{H}_I = \begin{pmatrix} \mathbf{R}_I & \mathbf{p}_I \\ \mathbf{0} & 1 \end{pmatrix} \in SE(3)$$

This representation leads to a local frame representation that reduces the non-linearity of the equations of motion and allows a representation of the rotations without singularity issues.
The configuration $\mathbf{H}$ can then be represented as a bloc diagonal matrix that gathers each so-called nodal variable described above.

$$\mathbf{H} = \text{diag}(\mathbf{H}_1, ..., \mathbf{H}_N)$$

Thanks to r control inputs $\mathbf{u} = [u_1, ..., u_r]$, the end-effector position $\mathbf{y}_{eff}$ of the MBS follows a prescribed trajectory $\mathbf{y}_{presc}(t)$. The latter is evolving in time and is defined by r scalar components. If such MBS is defined using N nodal variables $\mathbf{H}_I$, the equations that govern the dynamics of such MBS are

$$\dot{\mathbf{H}}_I = \mathbf{H}_I \tilde{\mathbf{v}}_I \quad \text{with } I = 1, ..., N \tag{1}$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{g}(\mathbf{H}, \mathbf{v}) + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{A}\mathbf{u} \tag{2}$$

$$\boldsymbol{\Phi}(\mathbf{H}) = \mathbf{0} \tag{3}$$

$$\mathbf{y}_{eff}(\mathbf{H}) - \mathbf{y}_{presc}(t) = \mathbf{0} \tag{4}$$

where $\mathbf{M}$ is the system symmetric mass matrix, $\mathbf{v} = (\mathbf{v}_1^T, ..., \mathbf{v}_N^T)^T$ is the vector of nodal velocities, $\mathbf{g}$ is the vector of internal and complementary inertia forces, $\mathbf{B}$ is the gradient of the kinematic constraints $\boldsymbol{\Phi}$, which are used to represent the connections imposed by the kinematic joints. The matrix $\mathbf{A}$ is a boolean matrix that applies the controls $\mathbf{u}$ on the system. The m dimensional vector $\boldsymbol{\lambda}$ is Lagrange multipliers related to the m kinematic constraints $\boldsymbol{\Phi}$. The last equation is called the *servo* constraint [13] and fixes a part of the motion. It assures that the end-effector position $\mathbf{y}_{eff}$ follows the prescribed trajectory $\mathbf{y}_{presc}(t)$.
In the case of an underactuated 3D system, $6N - m - r > 0$ is the dimension of the internal dynamics i.e., the flexible dynamics, which is represented by Eqs. (1)-(4). The trajectory would be completely specified if some initial conditions were provided for the internal dynamics. However, if the internal dynamics is unstable, the forward propagation of the initial condition would lead to an unbounded solution requiring very large control efforts $\mathbf{u}$, as represented in Fig. 1. A bounded solution can be defined using an optimization formulation, in which the initial conditions on the dynamics are left free.

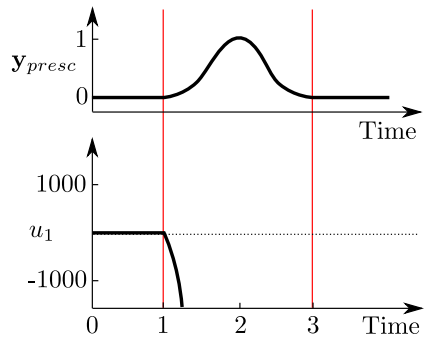


Fig. 1: Unbounded solution of the inverse dynamics when the internal dynamics is unstable.

3 Optimization problem

Based on the model of the flexible manipulator, the inverse dynamics problem is formulated as a constrained optimization problem where the internal dynamics is to be minimized. Considering that the internal dynamics can be represented using a function $\phi(\mathbf{H})$ depending on the nodal configurations, the optimization problem is the minimization of the objective function J on the time lap $T = t_f - t_i$.
Mathematically,

$$\min_{\mathbf{H}} J = \min_{\mathbf{H}} \frac{1}{2T} \int_{t_i}^{t_f} \|\phi(\mathbf{H})\|^2 dt \tag{5}$$

subjected to the equality constraints defined by the equation of motion of the flexible MBS Eqs. (1)-(4) for $t \in [t_i, t_f]$. One may observe that, in this fomulation, no initial and final values of $\mathbf{H}$ and $\mathbf{v}$ are defined. They are determined by the optimization algorithm itself.

3.1 Optimization process

To start the optimization process, an initial guess of the trajectory $\mathbf{H}(t)$ is required. To compute it, we can solve the inverse dynamics of an equivalent rigid manipulator which is a purely algebraic problem, since there is no internal dynamics in this case. Let this initial trajectory have a hat $\hat{\bullet}$ e.g., $\hat{\mathbf{H}}(t)$. The optimization is then carried out using a direct transcription method, i.e. the time interval is first discretized in s time steps t^k $(k = 1, ..., s)$ so that the optimization problem is reformulated as a discrete Nonlinear Programming (NLP) problem. Eventually, after a few iterations, the optimized trajectory is found as $(\mathbf{H}^1, ..., \mathbf{H}^s)$. Fig. 2 illustrates the process.
In the discrete settings, the minimization of the objective function, previously given by Eq. (5), can therefore be written in its discrete form

$$J = \frac{1}{2T} \sum_{k=1}^{s} \left[\|\phi(\mathbf{H}^k)\|^2 \right] h \tag{6}$$

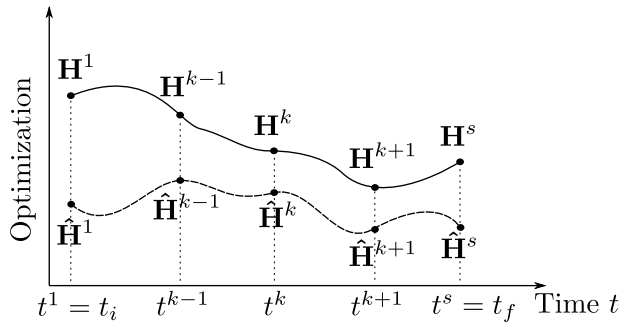


Fig. 2: Direct transcription method: optimization starting from the initial guess $(\hat{\mathbf{H}}^1, ..., \hat{\mathbf{H}}^s)$ that leads to the optimal trajectory $(\mathbf{H}^1, ..., \mathbf{H}^s)$.

subjected to the discrete constraints at each time step t^k

$$\dot{\mathbf{H}}_I^k - \mathbf{H}_I^k \tilde{\mathbf{v}}_I^k = \mathbf{0} \tag{7}$$

$$\mathbf{M}^k \dot{\mathbf{v}}^k + \mathbf{g}(\mathbf{H}^k, \mathbf{v}^k) + \mathbf{B}^{k,T} \boldsymbol{\lambda}^k - \mathbf{A}\mathbf{u}^k = \mathbf{0} \tag{8}$$

$$\boldsymbol{\Phi}(\mathbf{H}^k) = \mathbf{0} \tag{9}$$

$$\mathbf{y}_{eff}(\mathbf{H}^k) - \mathbf{y}_{presc}(t^k) = \mathbf{0} \tag{10}$$

where h is the time step size, $I = 1, ..., N$ and $k = 1, ..., s$. Additional time integration constraints are required to connect the discrete nodal configurations, velocities and accelerations.

3.2 Optimization variables

After discretization, the unknown variables of the optimization problem are

$$(\mathbf{H}^1, \mathbf{v}^1, \dot{\mathbf{v}}^1, \mathbf{a}^1, \boldsymbol{\lambda}^1, \mathbf{u}^1, ..., \mathbf{H}^s, \mathbf{v}^s, \dot{\mathbf{v}}^s, \mathbf{a}^s, \boldsymbol{\lambda}^s, \mathbf{u}^s)$$

with $\mathbf{H}^k = \text{diag}(\mathbf{H}_1^k, ..., \mathbf{H}_N^k)$ and each $\mathbf{H}_I^k \in SE(3)$. Obviously, some components of this set of variables belong to the $SE(3)$ group and are represented as matrices, for $k = 1, ..., s$. Classical optimization methods are not able to solve problems formulated with such matrix representation and dedicated methods are needed. Alternatively, in order to solve this problem using classical techniques, a reformulation based on a vectorial incremental variables is thus proposed.

We introduce the vector of incremental variables $\Delta\mathbf{q}^T = (\Delta\mathbf{q}_1^T, ..., \Delta\mathbf{q}_N^T)$, which determines the change between the initial guess $\hat{\mathbf{H}}$ and their current value $\mathbf{H}$.

At time step k, the relation between the configuration $\mathbf{H}^k = \text{diag}(\mathbf{H}_1^k, ..., \mathbf{H}_N^k)$ and the incremental variables $\Delta\mathbf{q}^{k,T} = (\Delta\mathbf{q}_1^{k,T}, ..., \Delta\mathbf{q}_N^{k,T})$ is

$$\mathbf{H}_I^k = \hat{\mathbf{H}}_I^k \exp_{SE(3)}(\widetilde{\Delta\mathbf{q}_I^k}) \tag{11}$$

where $\hat{\mathbf{H}}_I^k$ represents the position and orientation of node I at time step k for the initial guess.

The actual design variables $\mathbf{x}$ are thus

$$\mathbf{x} = (\Delta\mathbf{q}^1, \mathbf{v}^1, \dot{\mathbf{v}}^1, \mathbf{a}^1, \boldsymbol{\lambda}^1, \mathbf{u}^1, ..., \Delta\mathbf{q}^s, \mathbf{v}^s, \dot{\mathbf{v}}^s, \mathbf{a}^s, \boldsymbol{\lambda}^s, \mathbf{u}^s)$$

The optimization problem has now vectorial design variables and can be solved using a classical NLP algorithm. For consistency, the configurations $\mathbf{H}^k$ and $\mathbf{H}^{k+1}$ at two consecutive time steps are also related through another exponantial mapping and a time related incremental variable $\Delta\mathbf{Q}^k$. The relation between the relevant variables is illustrated in Fig. 3. Each arrow represents an exponential mapping $\exp_{SE(3)}(\tilde{\bullet})$ with either time incremental or configuration incremental arguments, i.e. $\Delta\mathbf{Q}^k$ and $\Delta\mathbf{q}^k$ respectively.

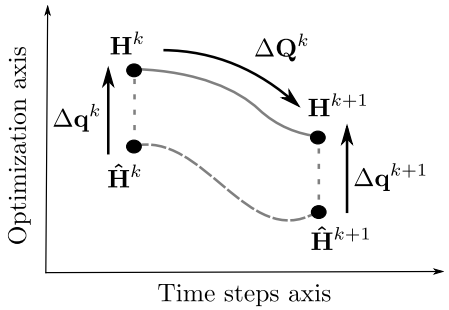


Fig. 3: Relation between variables $\Delta\mathbf{q}^k$, $\Delta\mathbf{Q}^k$ and $\mathbf{H}^k$.

4 3D serial example

The inverse dynamics of a flexible 3D system is now solved using the proposed approach. A serial 3 dof manipulator, as shown in Fig. 4, is considered. It is composed of two links: an upper arm and a forearm. The end-effector is modeled as a point mass m_{end} at the tip of the forearm. The upper arm and the forearm both have length l and a tubular square cross section. The upper arm has a side length a_1 and an edge thickness e_1. The forearm has a side length a_2 and an edge thickness e_2. While the former has a greater cross section and is considered as a rigid body element, the forearm is considered flexible and is modeled using 4 beam elements. The description of the beam formulation on $SE(3)$ can be found in [10]. The upper arm connects the first two hinge joints, controlled using inputs u_1 and u_2, to the third one, controlled using input u_3. The outputs of the system are the x, y and z components of the end-effector position $\mathbf{y}_{eff}$. The first hinge joint has its axis along axis z. The second and the third hinge joints initially have their axis along axis y. In the initial position, each link makes a 45° angle with respect to the x axis. By analysing the poles of the serial system, we find that the first unstable pole is located at the frequency of 13 Hz.

The trajectory the end-effector has to follow is a planar circular arc in the yz plane. The motion profile is built using a seventh order polynomial in order to insure continuity of the position, velocities, accelerations and

jerks over time. The end-effector starts from position $[2l\cos(45°)\ 0\ 0]$ and goes to position $[2l\cos(45°)\ l\ 0]$. The radius of the circular arc is thus $l/2$ m. The trajectory is covered in 1.1 s and the pre- and postactuation phases both last 0.2 s: the total simulation time is 1.5 s. The material parameters and dimensions of the 3D flexible arm can be found in Table 1.

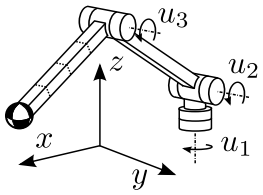


Fig. 4: Serial 3D arm system with one rigid body and 4 beam elements.

Tab. 1: Parameters of the serial arm system.

Upper arm	$l = 1$ m	$a_1 = 0.05$ m	$e_1 = 0.01$ m
			$\rho = 2700$ kg/m^3
Forearm	$l = 1$ m	$a_2 = 0.0075$ m	$e_2 = 0.0015$ m
	$E = 70$ GPa	$\nu = 0.3$	$\rho = 2700$ kg/m^3
End-eff.		$m_{end} = 0.1$ kg	

The convergence of the optimization process is quite sensitive to the initial guess of the problem. To compute it, a complete rigid system is considered. The beams of the forearm are replaced with a rigid body with the same geometrical and material properties. When gravity is acting on the robotic arm, it is important to correct the initial guess with the static deflection of the actual flexible arm. Regarding the numerical parameters of the generalized-α method, a spectral radius of $\rho_\infty = 0.01$ is considered ($\beta = 0.98$, $\gamma = 1.48$, $\alpha_m = -0.97$ and $\alpha_f = 0.01$). In order to best capture the system's dynamics, the system is discretized into $s = 150$ steps. This means that the time step size h is 0.01 s, which is about a tenth of the first unstable frequency.

Using the default tolerances of the FMINCON solver in Matlab®, the optimization is completed after 5 iterations and lasts 4 minutes (using a x64 bits i7-4600u CPU with 16 Gb RAM memory). The command inputs $\mathbf{u}$ and $\mathbf{u}_{rigid}$, with and without flexibility considerations respectively, are compared in Fig. 5. Some visible differences can be observed but the pre- and post-actuation in the input commands are hardly visible. These pre- and post-actuation phases can be observed by looking at the velocity profile of the three joints in Fig. 6. One can see that after 1.3 s, the velocity of the third joint is still varying. Although the torque u_3 is nearly zero in the post-actuation phase, it still results in some internal motion in the arm. It is important to point out that although some motion is present in the joints, the end-effector does not actually move during the post-actuation.

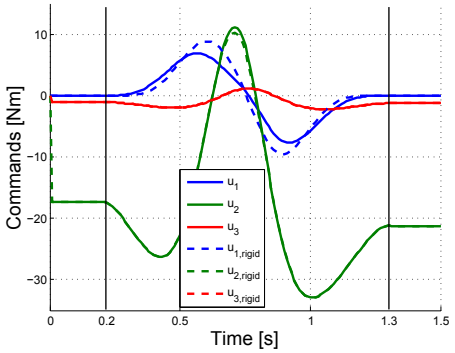


Fig. 5: Resulting joint velocity of a flexible 3D arm.

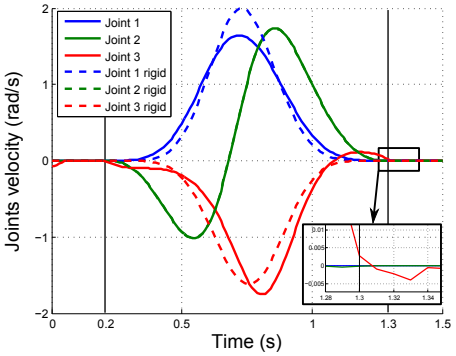


Fig. 6: Velocity of the three joint of a flexible 3D arm.

In order to verify the computed optimal inputs $\mathbf{u}$, both inputs $\mathbf{u}$ and $\mathbf{u}_{rigid}$ are applied to the flexible system and a direct dynamic analysis is performed. These inputs lead to end-effector trajectories shown in Fig. 7. The relative tracking error resulting from both direct simulations can be calculated at each time step t^k using (12).

$$e^k = \frac{\|\mathbf{y}_{presc} - \mathbf{y}_{eff}\|}{\|\mathbf{y}_{presc}\|} \tag{12}$$

where $\|\bullet\|$ is the classical Euclidean norm or L_2 norm. The relative rms error is then calculated as

$$e_{rms} = \sqrt{\frac{1}{s}\sum_{k=1}^{s}(e^k)^2} \tag{13}$$

The relative rms error e_{rms} is equal to 1.1% when $\mathbf{u}_{rigid}$ is used as input and drops down to $e_{rms} = 0.3\%$ when $\mathbf{u}$ is used.

5 Conclusion

In this work, the inverse dynamics of 3D flexible robotic arm is successfully solved using an constrained optimization formulation. The MBS is first modeled using finite elements formulated on the special euclidean

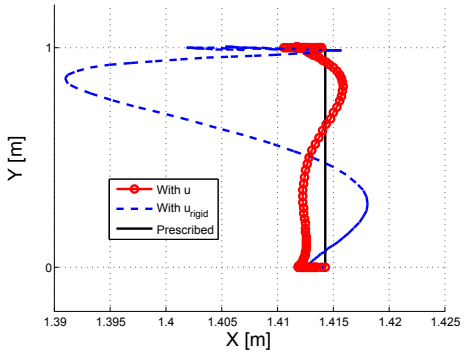


Fig. 7: End-effector trajectory of a flexible 3D arm using $\mathbf{u}$ and $\mathbf{u}_{rigid}$.

group $SE(3)$. Based on it, the inverse dynamics problem is defined as an optimization problem where the internal dynamics of the MBS is minimized. A direct transcription method is used to discretize the continuous optimization problem into a discrete optimization problem. In order to use classical optimization solvers, vectorial incremental variable are introduced to avoid optimization variables defined on $SE(3)$. The input commands computed using such methods manage to improve the tracking precision of the 3D flexible manipulator, as shown by the serial example. In further work, with the design of a suitable feedback loop, this method will be applied for off-line computation of the feedforward command of an experimental testbed.

Acknowledgements

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] V. Sonneville and O. Brüls, "A formulation on the special Euclidean group for dynamic analysis of multibody systems," *Journal of Computational and Nonlinear Dynamics*, vol. 9, 2014.

[2] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Transactions on Automatic Control*, vol. 41, no. 7, 1996.

[3] G. J. Bastos, R. Seifried, and O. Brüls, "Inverse dynamics of serial and parallel underactuated multibody systems using a DAE optimal control approach," *Multibody System Dynamics*, vol. 30, pp. 359–376, 2013.

[4] R. Franke, J. Malzahn, T. Nierobisch, F. Hoffmann, and T. Bertram, "Vibration control of a multi-link flexible robot arm with fiber-bragg-grating sensors," *In Proceedings of IEEE International Conference on Robotics and Automation*, 2009.

[5] J. Malzahn, M. Ruderman, A. S. Phung, F. Hoffmann., and T. Bertram, "Input shaping and strain gauge feedback vibration control of an elastic robotic arm," *In Proceedings of IEEE Conference on Control and Fault Tolerant Systems*, 2010.

[6] A. De Luca, "Feedforward/feedback laws for the control of flexible robots," *In Proceedings of the IEEE International Conference on Robotics & Automation*, 2000.

[7] D.-S. Kwon and W. J. Book, "A time-domain inverse dynamic tracking control of a single link flexible manipulator," *Journal of Dynamic Systems, Measurement and Control*, vol. 116, pp. 193–200, 1994.

[8] R. Seifried, *Dynamics of Underactuated Multibody Systems: Modeling, Control and Optimal Design*. Solid Mechanics and its Applications, Springer, 2014.

[9] R. Seifried and P. Eberhard, "Design of feedforward control for underactuated multibody systems with kinematic redundancy," *Motion and Vibration Control: Selected papers from MOVIC 2008*, 2009.

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

[11] A. Lismonde, V. Sonneville, and O. Brüls, "Solving the inverse dynamics of a flexible 3d robot for a trajectory tracking task," *In Proceedings of 21st International Conference on Methods and Models in Automation and Robotics*, 2016.

[12] A. Lismonde, V. Sonneville, and O. Brüls, "Trajectory planning of soft link robots with improved intrinsic safety," *In Proceedings of the 20th World Congress of the International Federation of Automatic Control*, 2017.

[13] W. Blajer and K. Kolodziejczyk, "A geometric approach to solving problems of control constraints: Theory and a dae framework," *Multibody System Dynamics*, vol. 11, pp. 343 – 364, 2004.

Robotic peg-in-hole process for assembly of light weight structures.

Abir GALLALA
University of Luxembourg
abir.gallala@uni.lu

Bassem HICHRI
University of Luxembourg
bassem.hichri@uni.lu

Meysam MINOUFEKR
University of Luxembourg
meysam.minoufekr@uni.lu

Peter PLAPPER
University of Luxembourg
peter.plapper@uni.lu

***Abstract*—Lightweight components used in aero-space sector are characterized by their versatility. Not only the part size, but also the geometry and design features change from one part to the other. Thus, using robots to manipulate such components is a delicate task that requires a highly accurate manipulation system. Indeed, one of the most useful solution is vision-based robotic manipulation. This presentation introduces a robust and efficient method for peg-in-hole process for assembly of lightweight structures. The proposed system is a combination of mechanical, control and process solution which consists in three steps: Pick, Place and Gluing. The input elements are irregular always-changing panels with different forms, dimensions and positions in addition to the inserts to be pegged on. However, the major problem is that it is quite difficult for the robot to distinguish between the different pins because of their small size and different shapes as well as to pick them, find their places and then put them in the adequate hole. As a second step, glue on the inside of very small holes without waste adhesive materials. To overcome this, we develop a 3D vision based algorithm for object localization and shape detection. Experimental results will be evaluated regarding the performance/time ratio.**

Keywords: Robotic assembly, 3D vision, Peg-in-hole.

I. Introduction

Using industrial robots to manage non-standard lightweight structures could be a bit tricky and sometimes challenging regarding the small size and the always-changing shape of the components. Furthermore, on the software scale, we need an algorithm to help the robot arm find the wright hole and place the object with less than 0.5 mm of error.

Thus, in this paper we introduce a vision-based peg-in-hole process solution for assembly of lightweight structures.

The general process flow of our solution is a combination of two main tasks. The first is to Pick small inserts then Place them into diverse panels and the second is to glue inside holes in the inserts. In order to achieve this approach, we distributed the solution into three main levels for each task: mechanical solution, control solution and process solution[1]. The mechanical solution includes the choice of hardware material, the control solution details the soft tasks and algorithmic approach and finally the process solution that combine and synchronize both mechanical and control solutions.

II. Requirements and constraints

The first constraint we have is the insert parameters: type of material, size and shape. All inserts have different types of materials and sometimes they could be constructed from a combination of two or more materials. In addition, the shapes and the sizes are variable. 36,16% of the inserts have a diameter of Ø11,40 mm and 15,22% have 9 mm as a height.

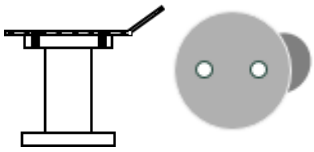


Fig. 1. An overview of the insert.

The second constraint is the dimensions of the panel in addition to the position, diameter and height of each hole in the panel. The maximum range of dimension that the a panel could take it is 2250 x 1500 x 30 mm.

III. Automatization roadmap

IV. Mechanical Solution

The mechanical solution introduces the hardware and material choice including the choice of the robot, the grasping and gluing tools, and the vision sensors.

A. Robot selection

The choice of the robot should respect the constraints related to the input objects (panel and inserts), the grasping method as well as the gluing method. The selected robot should be able to reach the whole area of the panel even in the case of a panel with maximum dimensions. The second parameter to take into account is the payload of the robot which is affected and basically related to the choice of the gluing tools which can be up to 35 kg as a maximum. First, before thinking in the choice of the robot we have to choose the kinematic: 6/7-axes, delta or Cartesian. The Delta robot has an astounding process speed but carrying heavy loads might cause speed reduction and has limit workspace thus we cannot use it because of the dimensions of our panels and the weight of the gluing tools and for the same reasons we cannot carry on with Cartesian. Thus, we kept the 6/7-axis kinematics. As a second step, we made a list of three 6/7-axis solutions: the FANUC M-710iC/45M can reach 2606 mm and handle around 45kg, the KUKA KR 60-3 can reach 2033 mm and handle 60 kg and the ABB - IRB 4600-40 that reach up to 2550 mm and handle 40 kg.

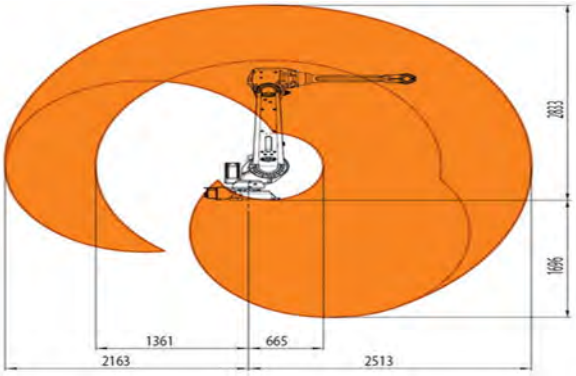


Fig. 2. The reach area of the ABB - IRB 4600/40

In the Figure above, we can see the work envelope of the ABB/IRB 4600/40 which could be our final choice.

B. Gripper selection

Since the first task of our project is the Pick&Place of the insert in the panel, the choice of the gripper and the grasping technology is quite important [2], [3], [4]. As shown in the Figure below, we have three main types of potential grasping technologies: Vacuum, mechanical and magnetic[5].

Fig. 3. Potential Grasping Technologies.

As shown in the Figure below, the vacuum gripper is ideal for handling flat components which is the case of our inserts. Even with the holes in the top of the inserts, the vacuum gripper still a good choice because of the small diameter of each hole [6].

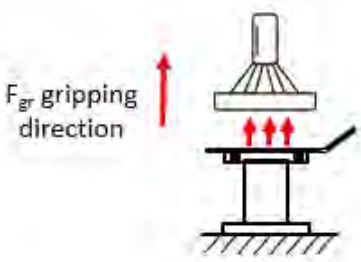


Fig. 4. Vacuum gripper.

On the contrary, it is difficult to use the magnetic gripper because of the unknown types of material of the inserts. Otherwise, for the mechanical grippers it is more complex.

Otherwise, the use of the mechanical gripper depends on the type of the gripper and its way of functioning. It is Suitable for both cylindrical and square shapes, but as shown in the Figure below, the use of the centric mechanical gripper could be much difficult and Risk tilting the insert due to its geometry.

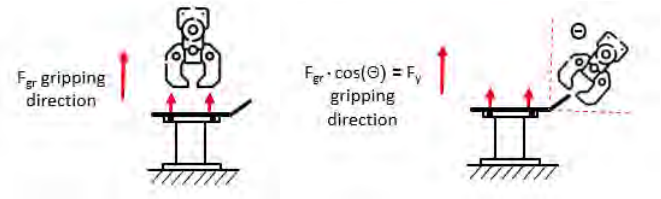


Fig. 5. Mechanical gripper.

Finally, the parallel and angular mechanical grippers are flexible for different types of inserts and especially suitable for small components such as our inserts [7].

As a conclusion, we kept two types of grasping technologies to test them with real robot and components, compare the results, time of execution and process and then decide for the best final choice.

C. Gluing solution

The gluing solution includes the method of gluing inside the holes in the top of the insert after place it in the panel and which is I the range of 2mm to 5mm. Basically, we have two types of tools for gluing: the cartridge based and the pomp based. Both methods are 6-Axis robot wrist mounting but the main difference is on the payload. The cartridge-based solution could up to 35kg of payload vs just six to seven kg for the pump-based solution.

D. Vision sensor selection

To pick and place the inserts inside different holes in the panel the robot needs an external visual sensor such as a 3D camera to localize, calibrate and recognize the different objects and distances[8], [9].

V. Scenario Selection

As we have two different tasks to do for each cycle (Pick&Place and gluing), we may need to design the possible scenarios we could apply. We have distributed the possible seniors into two lists according to the number of the workstations. For the two workstations scenarios we need two robots and we have four possible cases: serial workstations, parallel workstations, synchronized and independent workstations. For the one-workstation scenarios, we just have two cases: SMED workstation and “All in One” workstation.

a) Two serial workstations: The two serial workstations in the Figure below contains two separated process stations with two robots, one for the Pick&Place and the other one is for the gluing, and a conveyor belt to move the panels in a continuous process. The problem with this solution is if we have difference between the process times of the two processes.

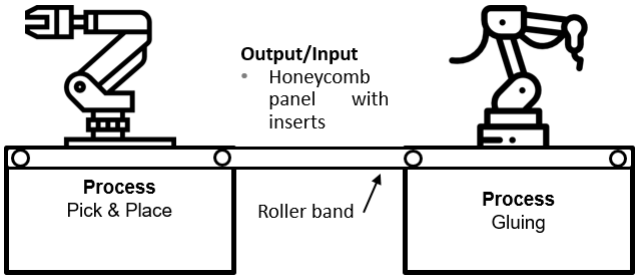


Fig. 6. Two serial workstations.

b) Two Parallel workstations: The solution of two parallel workstations, as shown in the Figure below, is faster because we could get two outputs per cycle but the fact that the robots have to change positions need more synchronization between them in order to avoid any risk of collision. Otherwise, in some cases one of the robots have to wait for the other process to be done before changing position, thus it could have some waste of time.

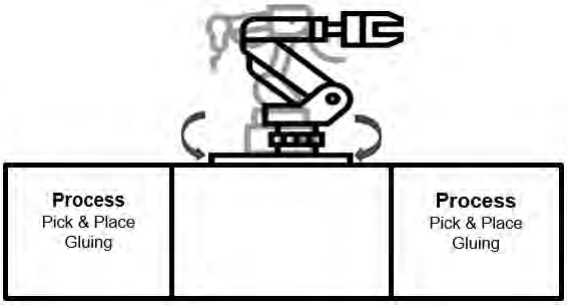


Fig. 7. Two Parallel workstations.

c) Two synchronized workstations: The synchronized workstation, as shown in the Figure below, consists of two robots working on the same workstation and the same object. This model has a very complex toolpath, complex trajectory and a high danger of collision.

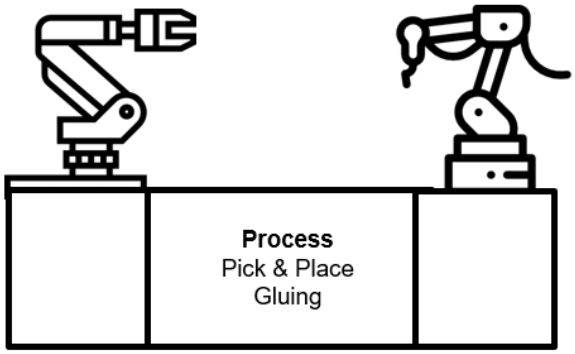


Fig. 8. Two synchrnized workstations.

d) Two independent workstations: The independent workstations as the name said and shown in the Figure below, consists of two completely independent processes. Each robot has separate inventories for its inputs and outputs, and work independently of the other one. The first for Pick&Place and the second for gluing.

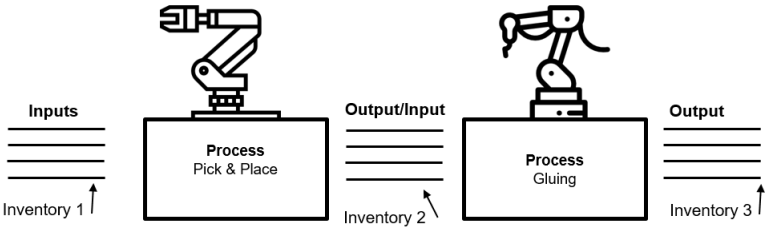


Fig. 9. Two independent workstations.

e) SMED workstation: The single minute exchange of die SMED-workstation means a rapid changeover of tools in order to do both tasks with the same robot in the same process. At first, the robot use the tools for the pick and place then it change it to glue in the same objects. However, the words "single minute" does not means it could takes only one minute.

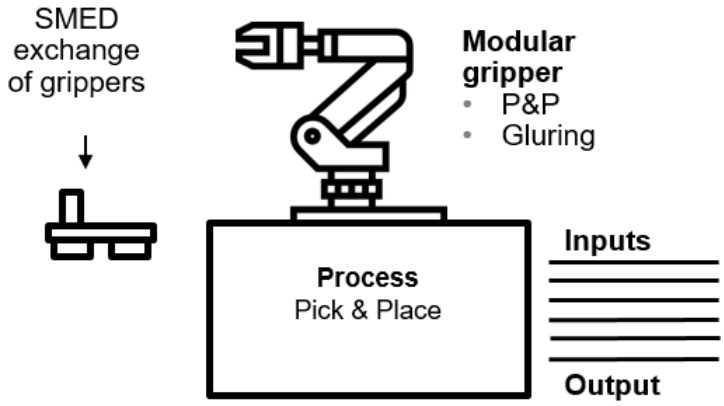


Fig. 10. SMED workstation.

f) "All in One"workstation: The concept of the "All in One" workstation is to have both tools for Pick&Place and gluing together. The aims is to complete both tasks for each insert in the panel and then move to the next insert. This model has reduced number of process, one single capex investment and time gaining. The unique limitation is the mechanical design of the terminal arm and tools.

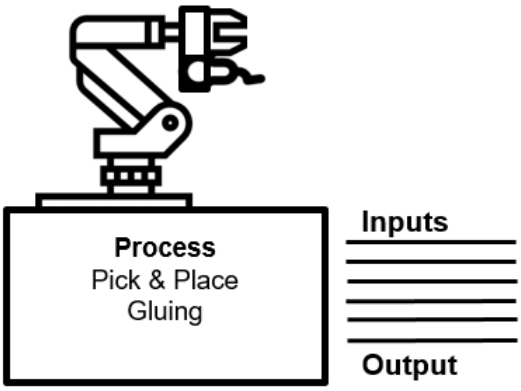


Fig. 11. "All in One" workstation.

g) Summary: Before choosing a final concept, we have to analyze each model, its limits and advantages, the process times and the number of outputs/cycle.

VI. Conclusion and Future work

A. Conclusion

This paper is an introduction to an industrial project. The aim is to pick small inserts and place them in different holes in a panel. Then glue inside small holes in the placed inserts. The dimension, shape and material is different from an insert to the other and from a panel to the other. As a first step we defined the general roadmap of the project and we started by defining the hardware selection which includes the choice of the robot, grasping tools, gluing tools and vision sensors. After analyzing each selection, we have to study to put together of all elements and the output of each final model.

B. Future work

In our lab we started by a simple scenario (single robot) for the Pick&Place using an ABB IRB 120. We will try to assemble this robot with a vacuum gripper that we design it in our lab and then we will built on this model in case we got good results.

References

[1] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[2] Kevin Tai, Abdul-Rahman El-Sayed, Mohammadali Shahriari, Mohammad Biglarbegian, and Shohel Mahmud. State of the art robotic grippers and applications. *Robotics*, 5(2):11, 2016.

[3] Keith Pearson. Gripper assembly for robot arm, February 28 1989. US Patent 4,808,898.

[4] Cheng-Hao Huang, Chi-Sheng Hsu, Po-Chien Tsai, Rong-Jyue Wang, and Wen-June Wang. Vision based 3-d position control for a robot arm. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1699–1703. IEEE, 2011.

[5] Robert Charles Downs and Mark Richard Weselak. Gripper mechanism, July 15 2003. US Patent 6,592,324.

[6] Jens Schick, Kurt Schmalz, Wolfgang Schmalz, and Thomas Eisele. Gripper system, in particular vacuum gripper system, January 7 2003. US Patent 6,502,877.

[7] Satish Sundar. Mechanical gripper for wafer handling robots, November 27 2001. US Patent 6,322,312.

[8] Rahul Kumar, Sunil Lal, Sanjesh Kumar, and Praneel Chand. Object detection and recognition for a pick and place robot. In *Computer Science and Engineering (APWC on CSE), 2014 Asia-Pacific World Congress on*, pages 1–7. IEEE, 2014.

[9] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.

A concept for construction of an adaptive reconfigurable robot manipulator to fit modern application scenarios

Prof. Dr.-Ing. Rainer Müller, Dipl.-Wirt.-Ing. (FH) Matthias Vette M.Eng., Ali Kanso M.Sc., ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gemeinnützige GmbH, Gewerbepark Eschberger Weg 46, Gebäude 9, 66121 Saarbrücken, Deutschland, a.kanso@zema.de

Abstract

The application fields of industrial robots have known a considerable development since their first robotic application by the production line GM in the USA, in 1961. Tasks and requirements are more and more complex and so are the diversity and variants of the product paired with ever shorter product life cycles. Those factors lead to a growing need for reconfigurable robotic systems. Conventional robot systems, with their fixed kinematic and dynamic dimensions, are not able to respond to theses challenges anymore. Therefore, they must be replaced with reconfigurable robot system, as well as additionals sensors and actuators. ZeMA continuously works on different scenarios in the course of the project FourByThree, in which the robots are set for the execution of the various applications, like bolting, welding, riveting and handling. Each of these applications poses specific requirements on the robot system. The welding process requires a high path accuracy as well as the possibility of sensory tracking of the rail data. The riveting process also requires a high pose accuracy and a programmable resilience. Handling processes require high repeatability, especially when the poses are taught in by Teach-In methods, while bolting processes require an online sensory tracking to be able to compensate the inaccuracy of the assembly unit. A concept for the construction of a reconfigurable robot system is being developed in the scope of the FourByThree project. Thanks to its modular elements, the built-up robot should be be able to quickly react to the varying application requirements. This is why the product is initially analyzed and later on examined in further detail. These information and boundary conditions collected throughout the analyze of the product and the process is the key constituent of the conception and development of the equipment. The developed manipulator of this robot system is modular and easy to reconfigure, and its adaptive and adjustable capacities fit the different tasks. A serial kinematic was chosen in order to guarantee the flexibility of the handling device and its ability to prevent obstacles, even when facing products with more complex volumes. This article presents a methodology to develop a reconfigurable robot manipulator. In order to check the validity of the proposed concept, a developed robot is tested in practical condition. Comparison of the theoretical results and experimental results shows the effectiveness of the proposed concept for development of reconfigurable robot manipulator.

Keywords: reconfigurable robotics, modularity, accuracy, calibration

1 Introduction

The industrial robot is designed to help workers with dangerous, repetitive and non-ergonomic tasks. Robots are generally used to increase the productivity and quality of the production equipment. Robotic systems play an important role in production and are used in various industries. According to the standard DIN 8373, a robot consists of the manipulator, which generates the movement of the robot, and the control, which provides both the path planning of the manipulator, as well as the communication with the environment through different interfaces.In addition to the robot, the robotic system includes the end effector and all the devices, gadgets or sensors necessary for the robot to perform the task and all the data communications interfaces through which the robot the devices or sensors operates, as far as these peripherals are monitored by the robot controller [1].

According to Siciliano and Khatib, the integration of a robotic system is executed in five steps, listed in the following order [2]: physical connection, establishment of the communication, setting of the configuration, the establishing of the application interface and finally, programming of the task. This integration effort of a robot system limits the ability of the companies to react to market changes and restricts the mastery of the variety of variants. The integration costs of an industrial robot system are ten times higher than the price of an industrial robot [3]. For this reasons, industrial robotic systems are widely used in mass production, such as in the automotive industry, but the use of these systems is low for small-scale production [4]. Armbruster noted that only 22% of the companies use robots in small-scale production and 14% in batch production [5]. Reconfigurable robot systems represent innovative solutions for minimizing the integration effort. So that, the links of the robot can be quickly reconfigured to respond to the change in the process workspace. They benefit from the flexibility and fast reconfigurability. Modular mechatronic construction kits are essential requirements to realize these systems. These innovative systems are being investigated at ZeMA, where a concept for construction of reconfigurable manipulators is developed.

Reconfigurable manipulators face significant challenges to ensure an acceptable deviation between the control and the physical model. This deviation has a tremendous impact on the performance characteristics of the robot that must be compatible with application requirements. In order to enable the robots to be used in accordance with their requirements, they must be individually assessed for their specific abilities in the scenario. Some performance features can be taken into account when designing the mechanical and kinematic structures. Other features of the robot can be improved by static calibration. Further capabilities may be achieved by supplementing the mechatronic system, e.g. additional sensors (e.g., camera, distance sensor) or actuators can be implemented.

2 State of the art

Industrial robots can be classified according to their kinematic architecture to serial, parallel or hybrid robots. In this paper, the focus is on serial robots and more precisely, on the modular serial robots. The latter relies on a mechatronic construction kit, i.e: modular arm, actuators, wrist and joints.

The advantage over conventional manipulators with predefined fixed kinematic lies in the possibility of flexibly adapting the robot structure to fit the current handling task. Thus, the requirments of the application specifies the robot structure which, in addition to the required workspace, should master the essential degrees of freedom. Defective modules can be quickly exchanged in case of a fault, and not usable modular elements are implantable in another modular system.

Figure 1 Left: Modular robot IGUS [6]; Right: Robot designed with the Schunk ERB modules [7].

The company IGUS offers several modular kits (see figure: 1), such as joints, articulated arm, drive units and tools, which can be reconfigured and expanded as required to reconfigure a serial robot. The material of the joints is plastic. The articulated arms can be provided in different lengths and designs.

The Robolink D is a directly driven joint with a plastic worm gearbox in which a stepper motor is used. This allows a serial manipulator with up to 4 DoFs to be reconfigured. The Robolink W is a joint with a rope drive and allows the combination of rotation and swiveling in a joint. This makes a serial manipulator reconfigurable with up to 6 DoFs [6]. Other manufacturers, such as Schunk [7], offer robot construction kits. The goal is to assemble a serial robot with different degrees of freedom (up to 6 DoF) and different kinematics from modular elements.

The robot system that has been built up so far is only considered on a physical level. Initially, the built-up system must be controlled, which means that the latter should be modeled and any change in the hardware should be considered in the control model. The so-called HD convention, developed by Hartenberg and Denavit in 1955 [8], is commonly used in handling technology to model serial kinematics. The controller concerns to generate the movement of the manipulator and monitors the pose of the robot. The direct kinematics calculates the end effector pose uniquely as a function of the joint variables. So that the pose in the Cartesian space will be determined depending on the manipulator configuration. Such a unique relation does not exist by solving the inverse kinematic problem. A fast and robust calculation of the inverse kinematic is essential in order to solve useable industrial applications with the modular robot.

The solutions of the inverse kinematic problem can be classified in closed-form and numerical solution [9]. Methods that offer closed-form solutions can provide all possible solutions for the robot configuration after a short calculation time. For this reason, they are preferred in the handling technologies. However, closed-from means a solution method based on polynomic of degree 4 or less. These solutions exist only for special cases and designs for serial robots with six degrees of freedom, such as three consecutive intersecting rotation axis (Inline wrist). That is why it makes sense to consider inline wrist as a modular element.

Serial robots with six degrees have in general 16 possible solutions of the inverse kinematic problem. The number of possible solutions depends on the design of the manipulator [10]. Angerer et al. discuss several methods to solve the inverse kinematic problem of a serial robot. Closed-form solution can not generally implemented to solve the inverse kinematic problem of a six degree of freedom serial robot. That is why they are not applicable to an arbitrary modular robot. Numerical solutions are time-consuming and not numerically stable since they depend on the initial value of the iteration. So they are hardly applicable for the online use with modular serial robots. Angerer et al. introduced the numerical elimination method and presented the Algorithm of Husty-Pfurner [11] and the method of Liu and Zhu [12] that bases on the algorithm of Raghavan-Roth as appropriate methods to solve the inverse kinematic problem of six degrees of freedom modular serial robot.

Manufacturing and assembly tolerances causes a devia-

tion between the physical and the control model of the robot. Therefore the assembled robot deviates from its model. The control model must be adjusted, so that the deviation between the physical model and itself will be reduced. Schrörer [13] mentioned that the kinematic parameters have the most significant factor that influences the pose accuracy of the robot. The kinematic parameters are interpreted in the HD-parameters that are represented in the control model. Calibration method base on identifying the kinematic parameters is essential after reconfiguring the manipulator to guarantee an acceptable accuracy which is needed in most of the application of the industrial robot. An intuitive concept to identify the kinematic parameters will be discussed in the coming chapters.

3 Development process

At ZeMA, various scenarios are investigated within the framework of the project FourbyThree Project, in which the robot is used for implementing various applications, such as welding, riveting, bolting and handling. A concept for the construction of a modular reconfigurable robot system is developed (see figure:2). Based on modular elements, the built- up robot can react to the varying application requirements. The product is analyzed in the first step, after which the process is considered more closely. The information and boundary conditions resulting from product and process analysis are the corner stone for the design and development of the production equipment. The requirements on the production equipment can be classified in kinematic and dynamic characteristics.

3.1 Kinematic dinemsions of the industrial robot

The application determines the necessary degrees of freedom of the robot. A handling device with an open kinematic chain or a serial robot will be examinated. Seral robot basically consists of regional and local structures.

3.1.1 Derivation of the regional structure

The regional structure consists of three joints (prismatic or revolute), connected by two links. This serves to achieve a desired position in the work space. The local structure consists of three revolute joints and is used to orient the wrist of the robot in the work space. The regional structure determines the kinematic structure and thus the work space of the robot. In general 20 useful regional structures exist, in which the type of joint (prismatic or revolute) and movement axis may vary from one structure to another [14]. The VDI guideline 2861 - Sheet 1 specifies the axis designation of a robot. According to it, A, B, and C respectively represent the rotation axis about z, y, and x axes. Whereby X, Y, and Z each represent the translation axis around the z, y, and x axes [15]. The regional structure can be determined according to the following way. The first regional axis always represent the z axis. If possible, the second axis is the z axis, otherwise its represented by the y axis. The third axis represent either the z axis if possible, or otherwise the y axis if possible, otherwise the x axis. A robot with a torus work space can achieve the process work space of the investigated scenarios. The selection is made on a CBB structure after having analyzed the work spaces of all useful structures. The regional structure guarantees the three degrees of freedom. Depending on the application though, the length of the links can vary. The length of the links is determined by modular elements with different dimensions. The local structure is determined by the degree of freedom of the robot.

Figure 2 Third FourByThree robot prototype.

For example, in applications requiring four degrees of freedom, the local structure consists of a joint, a link, and a flange. The arrangement of the joint is determined by the required orientation. The robot can be adapted by reconfiguring the joints and link elements, so that the robot can respond to the application modifications without a great effort. The controller should consider the mechanical change in the kinematics.

3.1.2 Work space analysis

The work space of a robot is defined by the set of positions of the end effector that can be reached. The developed robot must reach the set process work space with its simple work space. In this case, this means that the robot must reach the set work space with the defined degrees of freedom. Without the ability to solve the work space of the robot, it is impossible to state if the robot can fulfill the desired task. The work space of the robot is numerically determined by using the discretization method. In the discretization approach, the work space is covered by a regular arranged grid in either Cartesian or polar forms of nodes. To start the discretization, a volume based on the set work space is approximately taken. Since the regional structure of the robot is CBB, this implies a work space that takes the shape of a torus. The initial values of the robot dimensions are based on the dimensions of the product. Each node is examined to see whether it can be reached by the robot or not. This examination can be done by using the inverse kinematic model of the robot in order to determine the configuration of the robot at a desired position.

The work space analysis based on the discretization method may last up to several hours - if not days, depending on the work space volume and on the sampling frequency. This is why the closed-form method is the most preferred method among the users due to its short duration. In general, all six DoF robots have at least a numerical solution. In special cases, a closed-form solution can be developed depending on the kinematic, for exemple if the latter has several intersecting joint axes e.g. inline wrist.

Based on the result of the work space analysis, the HD-parameter will be modified to ensure that the robot achieves all the nodes in the set work space. In case the work space of the robot is much larger than the set work space, a shorter link length will be chosen, so that the difference between the two work spaces can be modified.

3.2 Dynamic dinemions of the industrial robot

After having determined the kinematic dimension of the robot, the dynamic characteristics can be studied. Drive forces and torques of the actuators can be recursively calculated based on payloads, as well as process forces, end effector velocity and acceleration. This calculation can be done by using a developed Matlab tool, and basing on the process requirements. The derivation of the actuator characteristics depends on the link and joint dimensions and masses. The moment of inertia of these elements affects the performance of the manipulator, especially for high dynamic applications.

3.3 Accuracy of the industrial robot

The recent researches in the field of robotics and its technological properties, as well as the help of computer science and electronics, have led to the development of highly precise robots. Thus, this leads to the creation of new fields of application of industrial robots. Conventionally, robot tasks have been programmed using teach-in method. In this method, the pose repeatability was the most important parameter. Due to increase of the task's complexity of the robots, modern offline programming methods became more essential than before. Offline programming means generation of a robot program without using a real robot. However, this is difficult to implement. The most important obstacle is the lack of accuracy of the robot's arm position. In addition to position and pose accuracy, the accuracy of the assembly or manufacturing cell plays an important role in offline programming. The deviation between the control and real model of the robot must be decreased after reconfiguration in order to decrease the pose accuracy which defines a convergence criterion to numerically determine the control model.

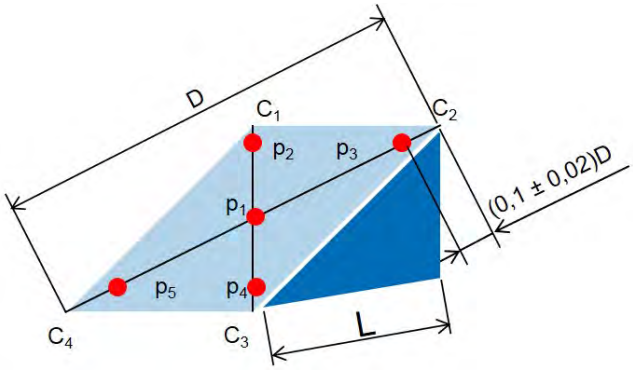


Figure 3 Measurement plane as defined in the ISO standard 9283

Thus, the pose repeatability and the pose accuracy are the most important characteristics of industrial robots. These two are defined in the ISO standard 9283 [16]. In addition, the standard defines other characteristics, e.g. path accuracy and path repeatability. It depicts the performance criteria and related testing methods to determine performance characteristics of manipulating industrial robots. It evaluates the suitability of a specific robot for a specific task.

In order to identify the pose accuracy and repeatability according to the ISO standard 9283, a cube must be defined in the workspace of the robot. A diagonal plane containing four edges on the cube must be chosen. Five positions will be located on the plane (see figure: 3). The robot will move in the following sequence p1, p5, p4, p3, p2, p1. The robot stops for a few seconds at each position. This sequence will be repeated 30 times.

Several publications are dealing with the ISO standard 9283. This is linked with the fact that the standard has several weak points [17], [18]. Slamani founds that the 30 cycles prescribed in the standard are not sufficient to determine the pose repeatability. The robot takes several hours to fully heat up so that the effect of the thermal expansion on the measurement can be minimized. The effect of the thermal expansion which has an influence on the repeatability will be decreased after 24 hours.

Hongyi [18] founds in his tests that the knowledge about the robot behavior within the ISO standard 9283 cube or the ISO standard 9283 plane is not sufficient for the evaluation of the robot. In reality the robot does not only work within this ISO cube, and it will have another accuracy outside this cube. Another problem is that the standard does not specify which measuring device the measurements are to be implemented. Each equipment has a different measuring method. It follows that different results are obtained depending on the measuring device, even if it always goes to the same 5 measuring points. Despite the weaknesses mentionned above, the ISO standard 9283 gives at least a clear definition of the robot characteristics. In the following, the mathematical formulas for calculation of pose accuracy are given.

$$AP_{pi} = \sqrt{(\tilde{x} - x_c)^2 + (\tilde{y} - y_c)^2 + (\tilde{z} - z_c)^2} \tag{1}$$

$$\tilde{x} = \sum_{j=1}^{n} x_j \tag{2}$$

$$\tilde{y} = \sum_{j=1}^{n} y_j \tag{3}$$

$$\tilde{z} = \sum_{j=1}^{n} z_j \tag{4}$$

$$AP = \frac{\sum_{i=1}^{m} AP_{pi}}{m} \tag{5}$$

Where AP_{pi} is the absolute accuracy of the position pi, and AP is the average of the absolute accuracy of the m tested positions. Also, $\tilde{x}$, $\tilde{y}$, $\tilde{z}$ are the coordinates of the center of gravity of the group of measured positions which are obtained after n-time repetition of the same position. Moreover, x_c, y_c, z_c are the coordinates of the desired position. Finally, x_j, y_j, z_j are the coordinates of the j-th measured position.

$$l_j = \sqrt{(x_j - \tilde{x})^2 + (y_j - \tilde{y})^2 + (z_j - \tilde{z})^2} \tag{6}$$

$$RP_{li} = \tilde{l} + 3S_{li} \tag{7}$$

$$\tilde{l}_i = \frac{\sum_{j=1}^{n} l_j}{n} \tag{8}$$

$$S_{li} = \frac{\sum_{(} l_j - \tilde{l})}{n-1} \tag{9}$$

$$RP_l = \frac{\sum_{i=1}^{m} RP_{li}}{m} \tag{10}$$

$$RP_l = \frac{\sum_{i=1}^{m} RP_{li}}{m} \tag{11}$$

Where RP_{li} is the pose repeatability of the position pi, and RP_l is the average of the pose repeatability of the m tested positions. The test method for the determination of these variables has been modified by several robot users to obtain a meaningful result. In the conducted experiments in the frame work of this paper, 30 positions distributed in the workspace of the robot have been measured in order to measure the pose accuracy.

3.3.1 Calibration of the industrial robot

Calibration offers the possibility to increase the pose accuracy of the robot without complex changes in its mechanical design. Robot calibration is a process for improving robot accuracy by parameter modification in the control software [13]. Calibration could be classified into two groups: Static calibration and dynamic calibration. The dynamic behavior of the robot can be improved by dynamic calibration. Thereby, the dynamic parameters such as weight, mass of inertia, friction as well as forces and moments will be dynamically identified as a function of the traversing speed. This has hardly any effect on the improvement of pose accuracy and is not discussed within the framework of these paper.

As an industrial robot increasingly used in the industry, the pose accuracy at the beginning was only partially at 20 to 30 mm [19]. In order to improve this pose accuracy to a few millimeters, simple calibration procedures were developed. A calibration method is developed by Trevelyan. This calibration method works as follows: a piece of millimeter paper was placed on a table, which was aligned with the robot. The robot was then programmed to run equidistant grid points. When the points were traced, the deviation was determined by means of a ruler or a slider. The advantage of this method was that it was very cost-effective. However, it was also very time-consuming. At the present time, the accuracy of industrial robots is in few millimeters range. For this reason, modern measurement methods with high enough resolutions are needed.

The lasertracker measuring system [20], is the most commonly used measuring instrument for robot calibration and accuracy determination in handling technology. The measurements are non-contact. A large area of the robot's work space can be detected, provided that a laser beam, deflected by a mirror fixed to a precise cardan joint, hits the retroflexror mounted on the robot end defector. The lasertracker can continuously measure every millisecond [17]. The accuracy of the lasertracker system is between 5 and 50 µm and are easy to handle. But they are expensive compared to other measurement systems. For the measurements carried out within the scope of this study, a lasertracker from the company API was used.

The deviation between the pose given by the offline programming and its real implementation depends on various errors. These errors can be divided into the systematic and stochastic errors. The systematic errors always occur with the same value with unchanged measuring conditions. They are determined by the system, that is, they can

neither be recognized nor switched off during repeating the measurements under the same measuring conditions. These errors have no effect on the pose repeatability of the robot, but it has a great and decisive influence on the pose accuracy of the robot.

Three different components of a assembly cell can be modeled in order to increase the accuracy. These are: robot, tool and workpiece. Within the scope of this paper, the robot and tool parameters are identified. The static calibration identifies the parameters that affect the static pose (position and orientation) of the TCP. When the static pose of the robot is considered, the static errors have to be determined and modeled. In order to reduce the complexity of the model and to make it as accurate as necessary and as simple as possible, not all parameters have to be modeled. Static errors combine geometric and non-geometric errors (see table: 1). If only the robot and its tool are considered, the geometrical errors are in the zero positions of the joints, parallelism, orthogonality, arm length and angle errors. The cause of these errors are adjustment errors, manufacturing and assembly tolerances and as a result of partial damage [19]. Schröer [13] has evaluated and arranged the relevant errors according to their importance and influence on pose accuracy. These are arranged according to this order: geometric parameters, joint and arm elasticity.

Tabelle 1 Geometric and non-geometrical errors.

Geometric errors	non-geometrical errors
Zero position error	Arm and joint elasticity (with and without load)
Arm length and angle errors	Gear fault (gear ratio, gear backlash, gear friction)
Parallelism	Tempreture influences
Orthogonality	Stochastic errors(eccentricity, resolution)

Therefore, the geometric quantities are considered as systematic errors. Thermal influences can also cause geometrical errors. These are not considered, however, since they have hardly any influence on the pose accuracy. Gear fault and elasticity are insignificant and classified as stochastic errors [13].

In general, each calibration procedure is to be carried out by five steps (see figure:4): First, the robot system has to be modeled. The model contains all possible parameters that have significant influence on the accuracy of the robot. The selection of the parameters depends on the application of the robot and the achieved accuracy. The measurement should be carried out with a suitable measuring system, frequently lasertrackers is used in the handling technique. The number of measurements affects the results of the calibration. The more measurements taken in the workspace, the more accurate the result of the calibration. The system parameters are to be identified from the carried out measurements. The technique offers various possibilities to identify the geometric parameters of a robot. The two common methods are geometric calibration of individual robot axes and numerical calibration of the entire system.

After comparison of both, numerical calibration of the entire system method was chosen to identify the parameters of an articulated arm robot in the context of this work. An intuitive tool has been developed, so that the end user can choose the parameter that has to be modified and modifies the parameters of the calibration process. This will be discussed in the next chapter.

Figure 4 Block diagram representing the calibration process.

3.3.2 Identification of the kinematic parameter

The aim of this method is to set up a system of equations for an error function with the aid of a parameterized description model of the robot as well as measured values and corresponding joint angles. The error function describes the deviations between the desired positions from the parameterized model and the measured positions as a function of the geometric parameters of the robot. A numerical optimization method is used to selectively modify parameters to reduce the error function.
The advantage of this method is that during the calibration of the whole system, the optimum of the parameters which best describes the overall model is determined by the numerical calculation method. The geometrical parameters will be described by the HD-parameters. The number of measuring positions depends directly on the number of parameters. Wiese [21] recommends to choose the number of positions to be measured by multiplying the number of unknown parameters with 1,5. This number can be further reduced if, in addition to the position, the orientation is also determined. This additional information provides twice as much information for each measured value [21].

A six degree of freedom robot has 24 HD-parameters, 6 from them represent the joints variable and they vary by changing the configuration of the manipulator. 18 parameters are constant and independent for the permanent configuration manipulator. Since an industrial robot is not useful without a tool, the latter must be mounted on it is flange. That the number of parameters that describe the kinematic of the manipulator increases to 27 where 6 parameters are variable. The above mentioned parameters are described by the parameter vector:

$$\underline{p} = \left(\theta_1 \ \dots \ \theta_f \ d_1 \ \dots \ d_f \ \alpha_1 \ \dots \ \alpha_n \ a_1 \ \dots \ a_n \ x_{TCP} \ y_{TCP} \ z_{TCP} \right)^T \tag{12}$$

After studying the kinematic of manipulator, linear dependent parameters can be eliminated and the number of parameters that must be identified can be decreased. The multidimensional Newton-Raphson method offers the possibility to determine the model parameters in a few iterations. For this purpose, the robot is moved to different positions. At each position, the permanent joint variables are noticed from the controller and the actual position is measured. Based on the direct kinematic model, the desired positon can be calculated. An error vector between the desired and actual positions is formed.The multidimensional Newton-

Raphson method attempts to minimize the error vector by varying the parameter that has to be identified. By doing so, the model parameter can be identified. The examined positions are distributed to cover the most of the volume of the workspace of the robot.

Multidimensional Newton-Raphson method The Newton-Raphson method is a standard method for the numerical solution of non-linear equations and equation systems in mathematics. Following, the procedure for identifying the kinematic parameter of a robot:
Steps 1: Definition of the desired vector, by measuring the current position with lasertracker

$$\underline{w}_{\text{desired}} = \left({}^0x_{f\,\text{desired}}\ {}^0y_{f\,\text{desired}}\ {}^0z_{f\,\text{desired}}\right)^T \tag{13}$$

Steps 2: Specification of the nominal parameter vector for each measured pose:

$$\underline{p}_{\text{start}} = \left(p_1\ \cdots\ p_n\right)^T \tag{14}$$

If the HD parameter is a variable parameter (rotation or displacement about z-axis), the joint variable is read from the control of the robot and given as an initial condition. If the HD parameter is a fixed parameter, its nominal value is given as the initial condition. For n measured positions:

$$\underline{p}_{\text{start}_{1:n}} = \begin{pmatrix} \underline{p}_{\text{start}_1} \\ \underline{p}_{\text{start}_2} \\ \vdots \\ \underline{p}_{\text{start}_n} \end{pmatrix} \tag{15}$$

The nominal parameter vector $\underline{p}_{\text{start}}$ of all poses serves as the start vector for the iteration calculation and has 24·n dimensions if the TCP coordinates are not taken into account and the retroreflector of the lasertracker is directly mounted on the flange of the robot. Otherwise, it has 27·n dimension.
Step 3: Calculate the HD transformation matrix for each measured position

$${}^0\mathbf{A}_f = \prod_{i=1}^{f} {}^{i-1}\mathbf{A}_i = \begin{pmatrix} b_{11}(\underline{p}) & b_{12}(\underline{p}) & b_{13}(\underline{p}) & b_{14}(\underline{p}) \\ b_{21}(\underline{p}) & b_{22}(\underline{p}) & b_{23}(\underline{p}) & b_{24}(\underline{p}) \\ b_{31}(\underline{p}) & b_{32}(\underline{p}) & b_{33}(\underline{p}) & b_{34}(\underline{p}) \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{16}$$

where

$${}^{i-1}\mathbf{A}_i = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & \cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i)) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{17}$$

θ_i, α_i, d_i and a_i represent the four HD-parameters defined by Hartenberg and Denavit [8].
Steps 4: Calculation of the actual position vector for each measured position based on the direct kinematic model

$$\underline{w}_{\text{actual}} = \left({}^0x_{f\,\text{actual}}\ {}^0y_{f\,\text{actual}}\ {}^0z_{f\,\text{actual}}\right)^T \tag{18}$$

for n positions:

$$\underline{w}_{\text{actual}_{1:n}} = \begin{pmatrix} \underline{w}_{\text{actual}_1} \\ \underline{w}_{\text{actual}_2} \\ \vdots \\ \underline{w}_{\text{actual}_n} \end{pmatrix} \tag{19}$$

The resulting actual vector has 3·n dimensions if position is taken into account and 6·n dimensions if the orientation is taken into account (pose).
Steps 5: Definition of the error function (if position is considered):

$$\underline{f}(\underline{p}) = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \left|\underline{w}_{\text{actual}} - \underline{w}_{\text{desired}}\right| \tag{20}$$

Where the error of each position is defined as max (f_1, f_2, f_3). The abort criterion is defined as maximum error (e.g. maximum error = 0.0001 m) of the deviation of all measured poses / positions. If this abort criterion is fulfilled, the iteration process is terminated and the current parameter vector $\underline{p}_m$ is determined as an ideal parameter vector. for n positions:

$$\underline{f}(\underline{p})_{1:n} = \begin{pmatrix} \underline{f}(\underline{p})_1 \\ \underline{f}(\underline{p})_2 \\ \vdots \\ \underline{f}(\underline{p})_n \end{pmatrix} \tag{21}$$

Where the resulting error vector has either 6·n dimensions, if orientation is taken into account (pose) or 3·n dimensions, if position is taken into account. Steps 6: Application of the multidimensional Newton-Raphson method. The assumption that $\underline{f}(\underline{p}_{m+1}) = \underline{0}$ implies:

$$\frac{\underline{f}(\underline{p})_{1:n_m}}{\partial \underline{p}_{1:nm}^T}(\Delta\underline{p}_m) = \underline{f}(\underline{p}_m)_{1:n} \tag{22}$$

$$\underline{p}_{m+1} = \underline{p}_m - \left(\frac{\underline{f}(\underline{p})_{1:n_m}}{\partial \underline{p}_{1:nm}^T}\right)^{-1} \cdot \underline{f}(\underline{p}_m)_{1:n} \tag{23}$$

The jacobian matrix for one pose is defined as follows:

$$\mathbf{F} = \frac{\underline{f}(\underline{p})}{\partial \underline{p}^T} = \begin{pmatrix} \frac{f(\underline{p})_1}{\partial p_1} & \cdots & \frac{f(\underline{p})_1}{\partial p_{24}} \\ \vdots & \vdots & \vdots \\ \frac{f(\underline{p})_6}{\partial p_1} & \cdots & \frac{f(\underline{p})_6}{\partial p_{24}} \end{pmatrix} \tag{24}$$

For n poses, the total Jacobian matrix is:

$$\mathbf{F}_{1:n} = \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \mathbf{F}_n \end{pmatrix} \tag{25}$$

The total Jacobimatrix (if pose is taken into account) has 6·n×24 dimensions without calibration of the TCP coordinates and 6·n×27 if TCP coordinates are to be calibrated. If the orientation is not considered, the number of rows of the Jacobian matrix will reduced to 3·n. The inverse of the Jacobimatrix is calculated using the Singular Value Decomposition (SVD). SVD allows the calculation of inverse of non-square matrices. After each iteration step m, the parameter vector will be update for each position:

$$\underline{p}_{1m} = \underline{p}_{1m-1} + \Delta\underline{p}_m \tag{26}$$

$$\underline{p}_{2m} = \underline{p}_{2m-1} + \Delta\underline{p}_m \tag{27}$$

$$\vdots \tag{28}$$

$$\underline{p}_{nm} = \underline{p}_{nm-1} + \Delta\underline{p}_m \tag{29}$$

Whereby $\Delta \underline{p}_m$ has 24 dimensions when only concerning the HD parameters and 27 dimensions when the TCP coordinates must be calibrated.

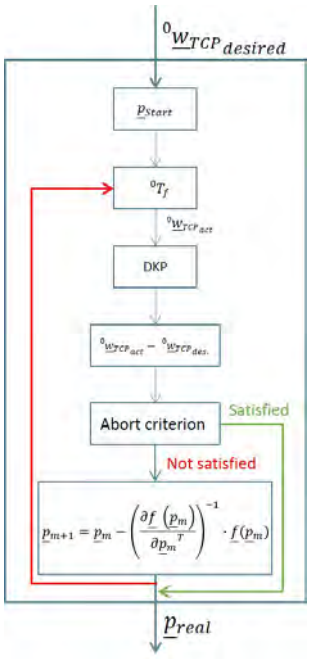


Figure 5 Calibraion algorithm.

$\underline{p}_{real}$ represent the real HD-parameter that best models the kinematic chain of the manipulator (see figure: 5).

4 Implementation and validation

To facilitate the commissioning and reconfiguration of the robot, a tool (see figure:6) has been developed at ZeMA to determine the real HD-parameters. Mathematics and robotic science knowledge are not essential for the end-user in order to calibrate the robot. It sufficient to measure a cloud of positions of the robot and save the result in an excel file with corresponding HD-Parameters. After which the parameters to be identified have to be selected in the tool and the abort criterion has to be defined. Its recommended to choose an abort criterion that is less than or equal than the repeatability of the robot, that is derived from the joint characteristics of the robot.

To validate the developed calibration tool, a serial robot that is developed at ZeMA (see figure:7) was analyzed. The nominal HD-parameters were derived from the CAD model of the manipulator, 30 positions that distributed the workspace of the robot were examined and the relevant information was taken as input to the calibration tool. The relative HD-parameter to be identified was selected in the tool. The position accuracy of the robot before calibration was measured as 2,4 mm. After two iterations, the tool support the modified value of the HD-parameters which reduces the position accuracy to 0.82 mm. To validate the result, the HD-parameters in the robot controller are updated with the modified HD-parameters identified with the algorithm. The robot is moved to positions that are measured before the calibration is occurred.

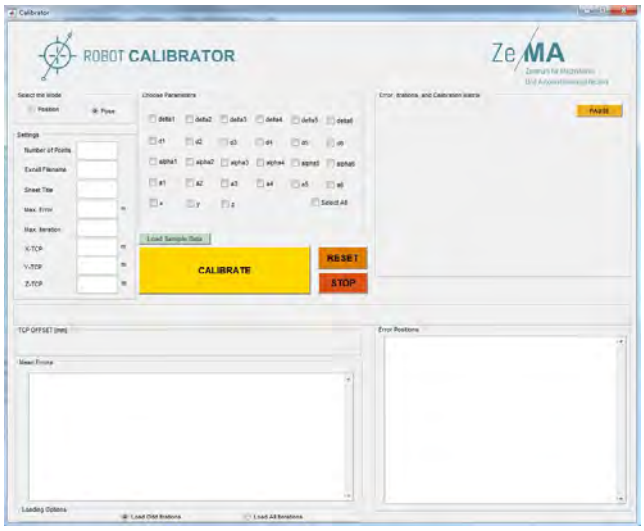


Figure 6 Calibration Tool developed by ZeMA.

The actual positon it measured and the position accuracy is determined as 0.82 mm which fulfils 100% of the value determined by the tool. An analogous experiment was implemented to UR10 robot from the company Universal Robot [22] and the position accuracy was reduced from several millimeters to few tenths of a millimeter.

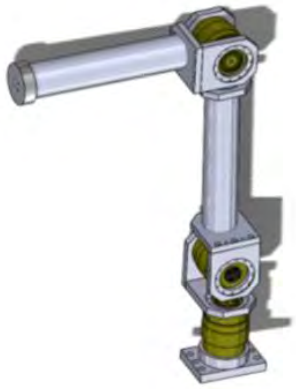


Figure 7 Three DoF robot developed by ZeMA.

5 Summary and outlook

Due to the modular design and the possibility of reconfiguration, the robot can easily respond to the requirements of the scenario. The development of a mechanical calibration gauge for a cost-effective integration and commissioning process of the robot system on the shopfloor are being integrated in future researches. The performance characteristics of the robot will be tested based on the process requirements. Furthermore, the modular concept of the FourBy-Three robotic system is transferred from the hardware to the software components. In this context, the identification of the robot parameters in a ROS compatible format is to be reproduced [23].

6 References

[1] DIN EN ISO 8373: *Industrieroboter - Wörterbuch*. Berlin. (1996), S. 2.

[2] Siciliano, B.; Khatib, O. (Hrsg.): *Handbook of Robotics.* Berlin, Heidelberg: Springer 2008.

[3] Bartscher, S.: *Mensch-Roboter-Kooperation in der Produktion - Chancen und Risiken.* Robot World 6 (2011) 1/2, S. 8-9.

[4] Krug, S.: *Automatische Konfiguration von Robotersystemen (Plug & Produce)* München, S. 1-3.

[5] Armbruster, H.; Kirner, E.; Kinkel, S.: *Neue Nutzungspotentiale für Industrieroboter - Ergebnisse einer Betriebsbefragung. wt Werkstattstechnik* online 96 (2006) 9, S. 631-636.

[6] igus® GmbH: *In: http://www.igus.de/.* Köln.(Stand: 30.01.2017).

[7] SCHUNK GmbH & Co. KG: *In: http://de.schunk.com/.* Lauffen/Neckar. (Stand: 12.09.2016).

[8] Hartenberg, R. S.: *Die Darstellung und Handhabung der niederen Elementenpaare in einer auf Matrizenrechnung basierenden Zeichensprache*. VDI-Berichte Bd 12. (1956).

[9] Craig, J. J.: *Introduction to Robotics Mechanics and Control 3rd edition*. Pearson. (2005), S. 100-106.

[10] Angerer, A.; Pfurner, M.; Brandstötter, M.; Hofbaur OVE, M.: *Modulare serielle Roboter* e & i Elektrotechnik und Informationstechnik 130.2 (2013): 48-53.

[11] Husty, M.; Pfurner, M.; Schröcker.; *A new and efficient algorithm for the inverse kinematics of a general serial 6R* In Proceedings of ASME 2005 29th mechanism and robotics conference, Long Beach. H.-P. (2005).

[12] Liu, S.; Zhu, S.: *An optimized real time algorithm for the inverse kinematics of general 6R robots. In Proceedings of the IEEE international conference on control and automation* Guangzhou, China, S. 2080-2084. S. (2007).

[13] Schröer, K.: *Theory of kinematic Modeling and Numerical Procedures for Robot Calibration*. Cambridge UK. (1993).

[14] Corves, B.; Müller, R.: *unpublished, Vorlesung: Kinematik, Dynamik und Anwendung in der Robotik*. Aachen. (2015).

[15] VDI Rechtlinie 2861 - Blatt 1 *Montage- und Handhabungstechnik; Kenngrößen für Industrieroboter; Achsbezeichnungen*. Düsseldorf. (1988).

[16] DIN EN ISO 9283: *Industrieroboter. Leistungskenngrößen und zugehörige Prüfmethoden*. Berlin. (1998).

[17] Slamani, M.; Nubiola A.; Bonev, A. I.: *Assessment of the Positioning Performance of an Industrial Robot* École de technologie supérieure, Montreal (Quebec), Canada. Industrial Robot, Vol. 39, No. 1, pp. 57-68, 2012.

[18] Hongyi, L.: *Enviroment for Industrial Robot Performance Evaluation* KTH Royal Institute of Technology Computer Science and Communication, 2012.

[19] Beyer, L.: *Genauigkeitssteigerung von Industrierobotern*. Hamburg (2004).

[20] Vincze, M.; Prenninger, J. P. und Gander H.: *A Laser Tracking System to Measure Position and Orientation of Robot End Effectors Under Motion* University of Technology Vienna 1994.

[21] Wiest, U.: *Kinematische Kalibrierung von Industrierobotern* Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH) 2001.

[22] Universal Robots A/S: *In: https://www.universal-robots.com/de/.* Köln.(Stand: 15.05.2017).

[23] Robot Operating System: *ROS. In: http://ros.org/.* (Stand: 21.10.2016).

The optimization of the deflection error of an industrial robot for a friction stir welding application

F.Dardouri[1,2], G.Abba[1] and W.Seemann[2]

***Abstract*— Today industrial robots are used in many manufacturing applications because of their versatility and easy applicability. Notwithstanding their performance these robots are not suitable for some manufacturing processes where uniform and high forces together with suitable precision of position are required. The present research is focused on one of the high thrust operations, the friction stir welding (FSW). This method for connecting two parts works while the connected materials are in the solid phase. For this reason a very high thrust force is needed to soften the material during the welding process. Due to these high forces the position of the tool of a serial robot deviates from the desired trajectory. In this paper the possibility of using a parallel structure device is investigated to improve the load capacity and stiffness of a heavy manipulator robot. Such a system may exert forces directly on the process tool. In this way the movement of the tool is mainly generated by the industrial serial robot, while the parallel structure ensures the generation of very high thrust forces.**

I. INTRODUCTION

Friction Stir Welding process is a comparatively new method of welding. This process invented in 1991 by Wayne Thomas in The Welding Institute (TWI) [Thomas et al., 1991]. It enables joining material in solid phase without reaching the melting temperature. Comparing with other joining processes, FSW has many benefits for welding metals. For this reason, after its invention several industries have shown great interest in this process, especially the aerospace industry, mainly due to the exceptional mechanical properties of welds, the absence of defects and typical porosities.

To use this process for welding linear joints, most industrial applications use specified FSW machines. One of the biggest specified machine was developed by ESAB in cooperation with Boeing for the project; space launch system (SLS), of NASA. The first application of FSW in the aeronautics was the Delta II in 1999, [ESAB, 2013]. These machines are characterized by high stiffness and thrust capacity, but they require significant investments and have a low manipulapility. It is possible also to use parallel robot (Tricept), see [Smith, 2007]. Major research on FSW using parallel structures were developed by HZG in Germany (former GKSS), with a first test in 1998, see [Strombeck et al., 2000]. Despite the fact that this type of structure is characterized by a high stiffness but it has a great ability to push only in one direction. To improve the thrust capacity in the other direction of a parallel robot, Palpacelli proposed to join it to a simple cable-driven device whose moving platform was rigidly attached to the robot end-effector. He applied this concept to a tricept after a static and kinematic modeling of the system [Palpacelli, 2016]. The application of FSW process using this solution may limit the manipulapility of the structure due to the cables use in the work plane. However, in industry, many applications require a large manipulapility to weld complex joints, demanding machines with several degrees of freedom. Because of that using an industrial robot broadens this flexibility of application.

Of course, Comparing serial robot with many other kinds of machines, they are characterized by low stiffness. For this reason, their use is usually limited to applications like packaging, assembling, or pick and pack, where the positioning accuracy of the tool is not a major factor and a large workspace is required. High force operations like welding or machining are preferably made by machines which have high stiffness, thrust capacity and accuracy, resulting in a better product quality. Nevertheless, their poor flexibility limits many operations, e.g. in cases where a complex path has to be realized and many places have to be reached by the tool. This flexibility of application is best done by a serial robot. Although In recent years, current researchers tried to replace the dedicated machines by serial robots due to their low cost. The idea in this paper is to use an industrial serial robot to perform the friction stir welding process (FSW).

Due to the high thrust force required for FSW, generally a heavy industrial robot uses to perform the process. Usually, the heavy industrial robots are considered among rigid structures, this condition remains valid unless the forces needed to be created by the robot do not exceed its carrying capacity, which is not the case such as this process requires a very high thrust force that must be provided by the serial robot to hold the contact between the tool and the workpiece surface. For example, to assemble two pieces of aluminum with 6 mm depth, we need $10\,\mathrm{kN}$. For that reason, its positioning accuracy becomes very low, when the effect of transmission elasticity is non-negligible. The deformation of the whole industrial robot needs to be considered especially, for applications that require a precise position of the tool as described by [Soron, 2008] and [Voellner et al., 2007], as in our case, it is necessary that the tool follow a defined trajectory. Therefore, this deformation makes deflection in each joint of the robot, which causes an error in following a given trajectory, see [Strombeck et al., 2000] and affects evidently

[1] F.Dardouri and G.Abba are with Design, Manufacturing and Control Laboratory (LCFC), Arts et Métiers ParisTech, 57078 Metz, France fawzia.dardouri@ensam.eu, gabriel.abba@ensam.eu

[2] F.Dardouri and W.Seemann are with Institut of Technical Mechanics (ITM), Karlsruhe Institut of Technology-KIT, 76131 Karlsruhe, Germany wolfgang.seemann@kit.edu

the quality of the weld. The magnitude of this error depends on the robot position in the work space and the direction of welding [Voellner et al., 2007], [Zaeh and Voellner, 2010] and [Qin et al., 2014]. The deviation error can be reduced by solving this difficulty.
The static performance and the stiffniss of an industrial robot can be greatly improved by adding a device in form of a parallel structure.This assisting device has the architecture of a parallel structure but it is not an existing parallel robot. It is just tree thrust branches able to create together very high thrust forces on the tool which can be controlled easily. In this paper we develop an optimization algorithm to minimize the deviation error. For the first time this algorithm allows the optimization of the parameters defined the parallel structure. In the second time, the optimization of the three forces created by this structure. Thereafter, we simulated the workspace of the whole structure to have an idea about its flexibility of the applications.

II. FSW Process

As shown in Fig. 1 the rotating tool used for the FSW process consists of a probe and a shoulder. Moreover, the external forces exerted on it are the thrust force F exerted by the FSW machine, F_x the axial force during welding when the tool advances along the joint line the material's resistance generates a force along x_0 and Fy, then during tool rotation the flow of material pushes the tool which generates a force along y_0.

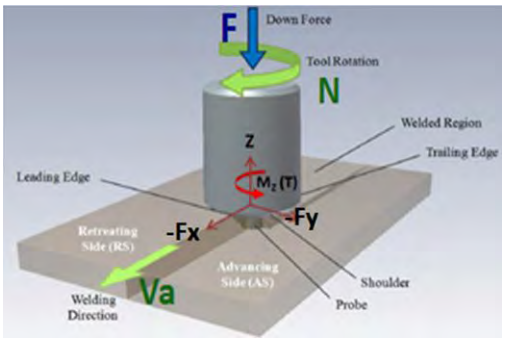


Fig. 1. Princible of the FSW Process [Gibson et al., 2014]

FSW is typically described in three steps. The first step is plunging. Here, the probe penetrates into the weld joint between the two parts to be assembled. This step stops when the shoulder touches the surface. The second step is welding. It involves the rotation of the tool and needs a high force which locally softens and mixes the material. This process assists the advance of the tool along the welding line [Fuller 2007, 2007]. The third step is retraction; in this step the probe is pulled out vertically from the material.

The FSW process is defined by four parameters, the thrust force F, the welding speed Va, the rotation speed N and the tilt angle B. The force is necessary for maintaining contact between the tool and the pieces to be welded. Further, it softens the material in order to assist the penetration of the probe into the joint. N and Va describe how fast the tool traverses along the interface and rotates, respectively. These two tool speeds have considerable importance, and therefore they need to be correctly set to get a successful welding cycle. There is a relationship between the heat input, the rotation speed and the welding speed during welding. It is arguable that decreasing the lateral speed or increasing the rotation speed will cause a hotter weld. FSW is characterized by a slower Va comparing to other welding processes. The last parameter is B, it tilts the tool mostly between $1.5°$ and $3.5°$ degrees such that the front of the tool is higher than the rear. This inclination assists to forge the material. See [Balasubramanian, 2009]. To use this process, these four parameters must to be set dependly on the material of the parts to be assembled the depth of the joint and the geometry of the tool and its material.

III. The wrench created on the platform

A. Description

In the optimization of the deviation error, an industrial robot KUKA KR 500-2MT was used. This is an articulated manipulator with 6 degrees of freedom (DOF) consisting only of revolute joints. This a heavy robot is able to carry in its end a load of $500\,\mathrm{kg}$. Moreover, the first three axes of this robot were modified, such that the transmission are twice as rigid as the transmission of the standard KR 500 Robot. This modification doubles the torque of three motors. However, owing to the high normal force needed for FSW the compliance of the robot remains important and the error due to the deformation in the joints cannot be neglected.
Fig. 2 shows a schematic of the parallel structure that we propose to investigate and which is optimized. The fixed base is connected to the moving platform by three identical limbs. Each limb consists of a prismatic joint P , an universal joint U at point M_i and a spherical joint S at point P_i, for i = 1, 2 and 3. The prismatic joint is driven by actuator which creates a force. Together the three limbs provide an additional force required to minimize the positioning error of the tool. Finally, to summarize This UPS parallel manipulators produces only three forces F_1,F_2 and F_3 on the moving platform but not its actuation and it is controlled in force by retracting or extending the actuators. As shown in the figure r_b is the distance $O_M M_i$ and r_p is the distance $O_6 P_i$.

B. Modelling of the system

The reference frame (O_0, x_0, y_0, z_0) is fixed to the ground, the reference frame (O_M, x_M, y_M, z_M) is fixed to the base of the parallel structure while the reference frame (O_6, x_6, y_6, z_6) is moving with the platform. In order to be able to locate the end-effector of an industrial robot, it is necessary to specify its positions and its orientation in the workspace. There are several methods to definie the rotation angles of the transformation between two frames. So in this work to calculate the orientation of the serial robot tool in R_0, the Euler method was used. Three successive rotations are defined as follows:

$$R(z_0, A), R(y_0, B), R(x_0, C) \tag{1}$$

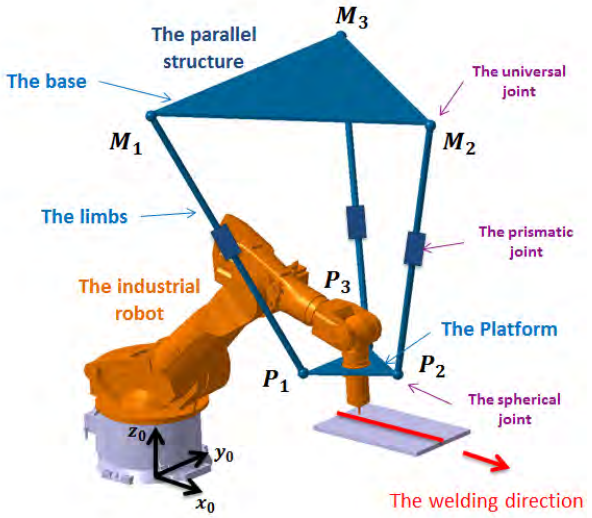


Fig. 2. Definition of the parallel structure

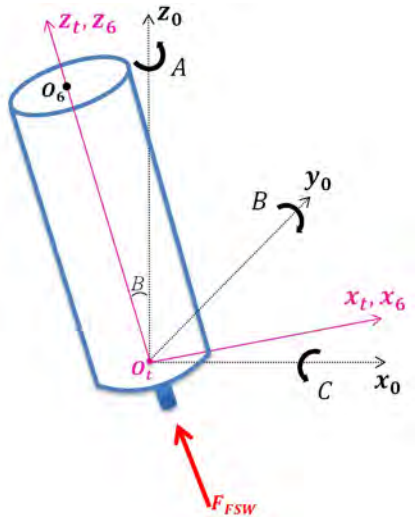


Fig. 3. Cordinite to define the tool orientation

The designated angles A, B and C shown in Fig. 3 describe the angles of roll, pitch and yaw. Each new rotation is carried out with respect to one of the fixed frame axes.

C is the rotation around the axis x_0, B is the rotation around the axis y_0 and A is the rotation around the axis z_0. In this work, the orientation of the end-effector is represented to obtain the rotation matrix depending these angles:

$$ {}^0R_6 = \begin{bmatrix} CACB & CASBSC - CCSA & SASC + CACCSB \\ CBSA & CACC + SASBSC & CCSASB - CASC \\ -SB & CBSC & CBCCi \end{bmatrix} $$

Where CA, SA, CB, SB, CC and SC represent cos(A), sin(A), cos(B), sin(B), cos(C) and sin(C) respectively.

The direct geometric model defines the set of relations which express the situation of the object j in the space in terms of the articular variables vector of the robot q.

$$ q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{bmatrix}^\top \quad (2) $$

q_i is the rotation angle of joint i. The situation of the end-effector j in the frame R_0 is defined by:

$$ T = \begin{bmatrix} {}^0P_j & R_j \end{bmatrix}^\top \quad (3) $$

$$ T = \begin{bmatrix} x_j & y_j & z_j & A_j & B_j & C_j \end{bmatrix}^\top \quad (4) $$

There are several conventions to calculate these relations. Modified Denavit-Hartenberg (MDH) is usually used to model robots consisting of revolute or prismatic joints including the manipulator used in this research work. The MDH parameters defined for this robot and are used thereafter with the support of the software SYMORO+ [Khalil and Creusot, 1997] to determine the relationship:

$$ T = f(q) \; i \in [1...6] \quad (5) $$

Where x_6, y_6 and z_6 of the equation (4) represent the coordinates of the point O_6 in the fixed frame R_0. This point was noted by:

$$ {}^0P_6 = \begin{pmatrix} x_6 \\ y_6 \\ z_6 \end{pmatrix} $$

Then the transformation matrix of the end-effector in R_0 is given as :

$$ {}^0T_6 = \begin{bmatrix} {}^0R_6 & {}^0P_6 \\ 0 & 1 \end{bmatrix} $$

The FSW tool is in direction of the 6th axis of the manipulator with the same orientation of the frame R_6, The position of O_t is

$$ {}^6P_t = \begin{pmatrix} 0 \\ 0 \\ L_t \end{pmatrix} $$

Where L_t is the length of the tool, see Fig. 3.

$$ {}^0P_t = {}^0P_6 + {}^0R_6 \begin{pmatrix} 0 \\ 0 \\ L_t \end{pmatrix} $$

And the transformation matrix of the tool in R_0 is :

$$ {}^0T_t = \begin{bmatrix} {}^0R_6 & {}^0P_t \\ 0 & 1 \end{bmatrix} $$

To calculate the vector P_iM_i it is necessary to calculate first the contact position between the platform and limbs P_i and the contact position between limbs and the fixed base of the parallel structure M_i in the fixed frame R_0 of the industrial robot. As shown in Fig. 4, the orientation of the frame of the

parallel structure is the same as of the fixed frame R_0 of the robot. The coordinates of point O_M in the robot frame are $x_M$in x_0, $y_M$in y_0 and H in z_0 direction. Then its position is defined by:

$$ {}^0P_M = \begin{pmatrix} x_M \\ y_M \\ z_M \end{pmatrix} $$

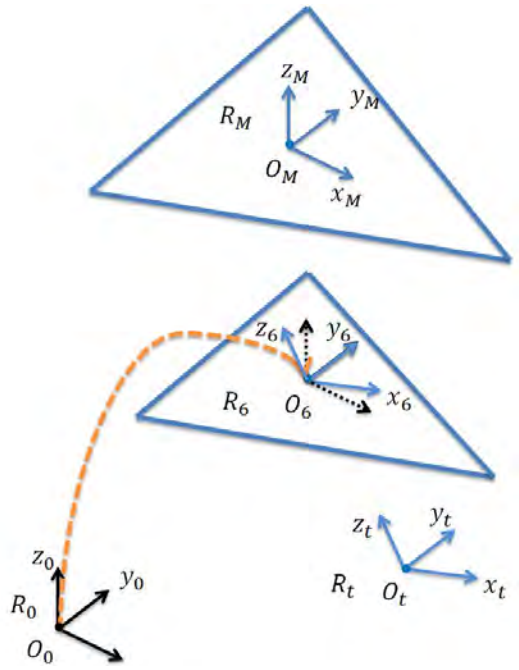


Fig. 4. Frames of the robot, the parallel structure and the tool

Therefore, the transformation matrix of the base of the parallel structure is given by:

$$ {}^0T_M = \begin{bmatrix} I_3 & {}^0P_M \\ 0 & 1 \end{bmatrix} $$

According to the geometry of the chosen base shown in Fig. 2, the coordinates of point M_i in frame R_M is expressed by:

$$ {}^MP_{M_1} = \begin{bmatrix} -\frac{r_M}{2} & -\frac{\sqrt{3}}{2}r_M & 0 \end{bmatrix}^\top $$

$$ {}^MP_{M2} = \begin{bmatrix} r_M & 0 & 0 \end{bmatrix}^\top $$

$$ {}^MP_{M_3} = \begin{bmatrix} -\frac{r_M}{2} & \frac{\sqrt{3}}{2}r_M & 0 \end{bmatrix}^\top $$

The three contact positions between limbs and the base of the parallel structure in the fixed frame R_0 were defined by:

$$ M_i = {}^0P_{M_i} + I_3 \, {}^MP_{M_i} \tag{6} $$

The coordinates of the point P_i in frame R_6 are given by:

$$ {}^6P_{p_i} = \begin{pmatrix} x_{p_i} \\ y_{p_i} \\ z_{p_i} \end{pmatrix} $$

Such as the platform geometry was chosen like the base geometry, for this reason the coordinates of the points expressed in the end-effector frame R_6 are:

$$ {}^6P_{p_1} = \begin{bmatrix} -\frac{r_p}{2} & -\frac{\sqrt{3}}{2}r_p & 0 \end{bmatrix}^\top $$

$$ {}^6P_{p_2} = \begin{bmatrix} r_p & 0 & 0 \end{bmatrix}^\top $$

$$ {}^6P_{p_3} = \begin{bmatrix} -\frac{r_p}{2} & \frac{\sqrt{3}}{2}r_p & 0 \end{bmatrix}^\top $$

To calculate the contact positions between limbs and the platform p_i in the fixed frame R_0 the homogeneous transformation is defined by:

$$ P_i = {}^0P_6 + {}^0R_6 \, {}^6P_{p_i} \tag{7} $$

As mentioned, in this work the industrial robot allows to move and to position the tool while the parallel structure allows to improve its stiffness. Then, to correct the deviation of the end effector, we need to calculate torques created by the parallel structure on the platform. Each thrust limb of this parallel structure produces a wrench τ_i on the moving platform in point P_i defined by:

$$ \tau_i = \begin{bmatrix} F_i^\top C_i^\top \end{bmatrix}^\top \tag{8} $$

As shown in Fig. 2 the forces exerted by the limbs are:

$$ F_i = f_i \, Z_i \tag{9} $$

f_i is the force created by the actuator i of the parallel structure according to the direction Z_i. According to the transformation matrices expressed in equations (6) and (7), it is possible to calculate the positions of the points P_1, P_2, P_3, M_1, M_2, M_3 during the movement of the industrial robot and for all these configurations in the space. Using the coordinates of these points expressed in the frame R_0 the vector Z_i is obtained as

$$ Z_i = \frac{P_i M_i}{\|P_i M_i\|} \tag{10} $$

The vector of moment Ci created by the actuator i in point O_6 is:

$$ C_i = O_6P_i \times F_i \tag{11} $$

O_6P_i is the vector from the origin of the coordinate of the platform center defined to the point where the force is exerted

$$ O_6P_i = r_p U_{Pi} \tag{12} $$

where r_p is the distance between O_6 and P_i along the direction U_{Pi}.

$$ U_{Pi} = \frac{O_6P_i}{\|O_6P_i\|} \tag{13} $$

The spindle of the FSW tool is located at the 6th axis of the manipulator, which is supposed to remain inclined with the small tilt angle M during the process, fig. 3. The FSW force is exerted on the robot tool along its axis z_t as shown in fig. 3. This force can be defined in frame R_0 by:

$$F_{FSW} = \begin{pmatrix} -F_{FSW} sin(B) \\ 0 \\ F_{FSW} cos(B) \end{pmatrix}$$

The wrench at point O_6 created by the FSW force expressed in the fixed frame is:

$$\tau_{FSW} = \begin{bmatrix} -F_{FSW} sin(B) & 0 & F_{FSW} cos(B) & 0 & 0 & 0 \end{bmatrix}^\top \quad (14)$$

Here, we neglet the torque created by the FSW force on the robot tool C_{FSW}.

The total wrench applied on the platform at its center O_6 in the fixed frame R_0 is:

$$F_{ext} = \tau_{FSW} + \textstyle\sum_{i=1}^{3} \tau_i \quad (15)$$

IV. Deviation error

This paper presents an approach to improve the quality of friction stir welding using serial robot. The idea is to minimize the deviation error of the tool due to the application of the external forces. The positioning accuracy is depending on the tool orientation and location [Deblaise et al., 2006].

A. The error in the joint space

In the industrial robot flexibility is originated from links and joints. However, many researchers claim that, because of the larger stiffness in links, its flexibility could be ignored. [Dumas, 2011] had analysed in her research work that link flexibilities contribute to 25% of the global deflection. For this reason, it is generally admitted that in this kind of machine the flexibility of the structure originates mainly from the joints. The error in the joint space is typically associated to the gearbox flexibility, including motors and transmissions. This source of rotation is the most important contributor to positional inaccuracies [Duelen and Schröer, 1991], [Schröer, 1993]. Therefore, the links of serial robots used in this work are condidered as rigide and only the joint errors are considered, and they are modelled by linear torsional springs [Bres et al., 2010]:

$$\tau = K \Delta q \quad (16)$$

Then, τ is the vector of joint torques, and K is a diagonal matrix for the joint stiffness. The difference $\Delta q = (q_d - q)$ is the source of errors in the joint space, where q_d is the desired angle and q is the actual angle. During the application of a load, this model is used to calculate the deformation created in the joints of the robot.

Dynamic model Dynamic modeling of the robot is required for mechanical design, controls and simulations. A corresponding model shows the relationship between positions of the manipulator joint, torques, speeds, accelerations, friction and external forces. This leads to a set of nonlinear differential equations of order two. It gives the state of the robot at each moment as it can analyse the stability of the control and performance trajectory. The Euler-Lagrange formulation for the joint space dynamic model is written as follows [Khalil and Kleinfinger, 1986] , [Wernholt and Östring, 2003]:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + F_s sgn(\dot{q}) + G(q) = \tau - \tau_{ext} \quad (17)$$

$D(q)$ is the robot inertia matrix, $C(q,\dot{q})\dot{q}$ represents Coriolis and centrifugal terms, F_v is the viscous friction matrix, F_s is the static friction matrix, $G(q)$ is the gravitational torque vector, $J^\top \tau_{ext}$ is wrench vector due to force and torque on the end effector.

During FSW the welding speed Va is always constant and very low, for this reason the acceleration can be assumed to be zero. In this contribution, we use a KuKa robot, which is characterized by a small static friction, and therefore F_s was neglected. Equation (17) can be reduced to equation (18)

$$\tau = G(q) + \tau_{ext} \quad (18)$$

B. The error in the operational space

Using the direct kinematic model the relationship between the error in the joint and in the operational spaces can be written as follows:

$$\Delta X = J(q)\Delta q \quad (19)$$

Where $J(q)$ is a 6×6 matrix which is known as a manipulator Jacobian that relates Cartesian velocities to joint velocities. this jacobian matrix is composed of two parts:

$$J(q) = \begin{pmatrix} J_v \\ J_w \end{pmatrix}$$

J_v calculates the cartesin velocity vector which is obtained from the derivation of the platform position vector

$$J_v = \frac{d^0 P_t}{dq_i} \quad (20)$$

$$J_v = \begin{bmatrix} \frac{dx_6}{dq_1} & \frac{dx_6}{dq_2} & \cdots & \frac{dx_6}{dq_6} \\ \frac{dy_6}{dq_1} & \frac{dy_6}{dq_2} & \cdots & \frac{dy_6}{dq_6} \\ \frac{dz_6}{dq_1} & \frac{dz_6}{dq_2} & \cdots & \frac{dz_6}{dq_6} \end{bmatrix}$$

J_w calculates the vector of rotation speed of the tool that is obtained by this expression:

$$J_w = T_r \frac{dR_t}{dq_i} \quad (21)$$

T_r is a 3×3 matrix defined in [Siciliano and Khatib, 2016]

$$T_r = \begin{bmatrix} -SB & 0 & 1 \\ CBSC & CC & 0 \\ CBCC & -SB & 0 \end{bmatrix}$$

Now combining (16) and (19) provides:

$$\Delta X = JK^{-1}\tau \quad (22)$$

Equations (18) and (22) provide the deviation error in the operational space:

$$\Delta X = JK^{-1}(G(q) + \tau_{ext}) \tag{23}$$

Among the applications of the Jacobian matrix is to define a linear relationship between the external forces exerted at the tool and torques required at the joints of the industrial robot to support these forces. But we cannot use the same Jacobian matrix calculated in the point because the forces exerted on the patform in the point. For this reason, it is necessary to calculate the jacobien in O_6

$$\tau_{ext} = J_6^{\top} F_{ext} \tag{24}$$

This expression allows to project all end-effector forces at the joints. This important relationship not just for the development of static forces but also can be used for robot control. But we cannot use the same Jacobian matrix calculated in point O_t because the forces were applied to the platform and not to the welded parts. For this reason, it is necessary to calculate the new Jacobian in O_6

$$J_6(q) = \begin{pmatrix} J_{v_6} \\ J_{w_6} \end{pmatrix}$$

R_6 and R_t have the same orientation what gives us this equality

$$J_{w_6} = J_w \tag{25}$$

But,

$$J_{v_6} = \frac{d^0 P_6}{dq_i} \tag{26}$$

Integrating equation (15) in (24) and equation (24) in equation (23) provides:

$$\Delta X = JK^{-1}(G(q) + J_6^{\top}(\tau_{FSW} + \textstyle\sum_{i=1}^{3} \tau_i)) \tag{27}$$

$$\Delta X = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta C \\ \Delta B \\ \Delta A \end{pmatrix} \tag{28}$$

$$\Delta X = \begin{pmatrix} \Delta P_{(3\times 1)} \\ \Delta R_{(3\times 1)} \end{pmatrix} \tag{29}$$

with the deviation errors ΔP along the axes x_0, y_0 and z_0. Similarly ΔR contrains the orientation errors about the axes x_0, y_0 and z_0.

V. The optimization of the error

A. The position error (EP)

The EP occurs when the tool deviates laterally of the desired weld line and contributes to positional inaccuracies. The error created along the normal of the workpiece during FSW does not affect the positional accuracy of the tool in the welding plane. But can be affect the welding quality then, the deviation errors creted along the three axes in the space have been minimized using the following expression:

$$EP = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{30}$$

With Δx, Δy and Δz are the deviation along the axis x_0, y_0 and z_0 respectively.

B. The orientation error (ER)

Among the FSW parameters we defined the tilt angle. In some applications, it is different from zero. And a higher value of this angle generates a blocking in advance, and a smaller value causes weld defects. Therefore, this angle has to be chosen with care, before starting the welding. In addition, it is very important that it remains constant during welding. But the action of the external loads changes the tool orientation which leads to an undesired tilt angle value and results an orientation error. Consequently, we get an incorrect welding. The orientation error reads:

$$ER = \sqrt{\Delta A^2 + \Delta B^2 + \Delta C^2} \tag{31}$$

C. Optimization results

For a good quality of welding an optimization algorithm in MATLAB is developed that minimizes ER and EP. This has been done by optimization the following function:

$$E = (\lambda EP + ER)^2 \tag{32}$$

Where, λ is the Lagrange multiplier.
The position and the orientation errors are in terms of the parameters of the parallel structure which are defined with the matrix:

$$V = \begin{pmatrix} r_p \\ r_M \\ x_M \\ y_M \\ z_M \end{pmatrix}$$

Then, as previously mentioned r_p is the dimention of the platform, r_M is the dimention of the base, x_M, y_M and z_M are the coordinates of the base center O_M (see Fig. 2). In addition, these errors depend on the forces created by the three actuators of the parallel structure:

$$F_p = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix}$$

To minimize E, we developed an optimization algorithm in MATLAB. This algorithm calculated the optimal parameters which correspond to the minimum deviation. Thereafter this algorithm gives us the optimal three forces which have to be realized by the parallel structure as a fonction of the position of the tool during welding. These parameters and forces then correspond to a minimum error.

It is revealed that F_2 is totally different from F_1 and F_3. However, F_1 has approximately the same value as F3 (see

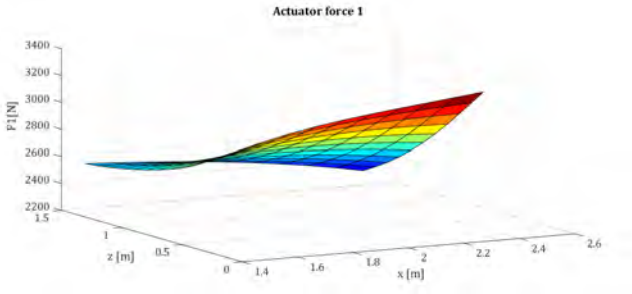


Fig. 5. External force created by the limb 1 of the parallel structure

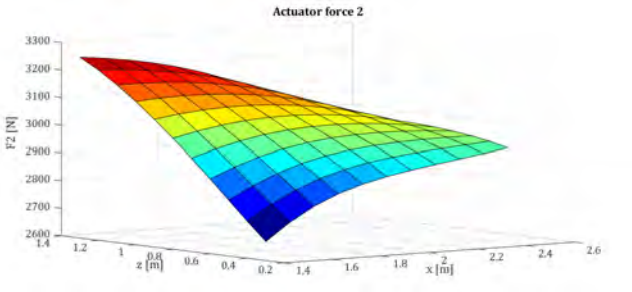


Fig. 6. External force created by the limb 2 of the parallel structure

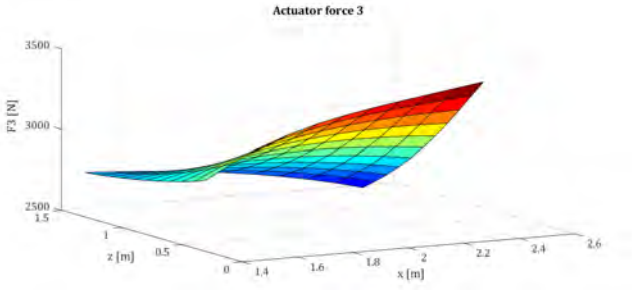


Fig. 7. External force created by the limb 3 of the parallel structure

Fig. 5 and Fig. 7). As expected this is because F_2 is applied directly on the welding line while F_1 and F_3 are applied symmetrically on both sides of the welding line. Moreover, it has been observed that to correct the error, forces created by the parallel structure are between 2500 N and 3500 N. Therefore the parallel structure is technically feasible.

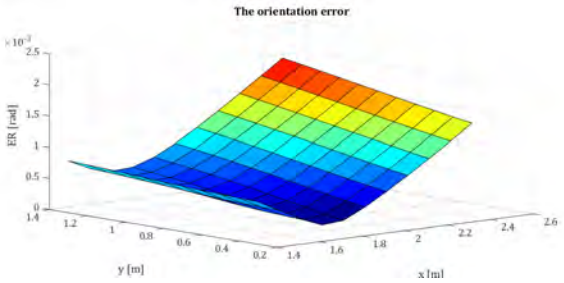


Fig. 8. The orientation error (ER)

The dimensioning parameters of the parallel structure and the three forces estimated to minimize the function E were

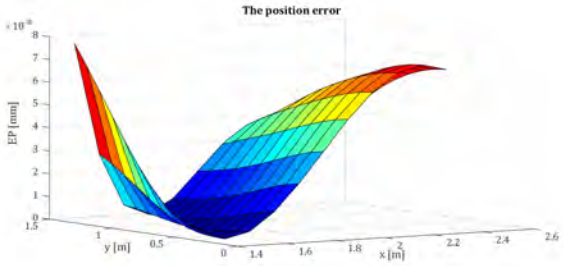


Fig. 9. The position error (EP)

used to calculate the position and rotation error. As shown in Fig. 8, the maximum rotational error is equal to 2.5×10^{-3} rad. Similarly, as shown in Fig. 9 the maximum position error is equal to 8×10^{-8} mm. The deviation of the tool is very well corrected by adding the parallel structure.

D. The workspace of the structure

The idea to add a parallel structure allows improving the rigidity of the robot. However, it limits the workspace and the index of the manipulability [De Backer, 2014]. The workspace of the hybrid structure was not calculated in this work. However, it has been simulated using the software Catia. From this simulation it has been observed that the tool can go up to $y_0 = \pm 1.2\mathrm{m}$ (because of the structure symmetry on the plane (x_0, z_0)). Further, different configurations have been tested in the plane $y_0 = 1.2\mathrm{m}$ using Catia as shown in Fig. 10. It has been revealed that there is no collision between the serial and the parallel structure in each configuration. At an initial stage, the workplace can be estimated by a parallelepiped of the following dimensions. the large length of the base equal $2.5\mathrm{m}$, the small length of the base equal $2.5\mathrm{m}$ and the height equal $1\mathrm{m}$. To know the exact limits of the tool before collision it is necessary to calculate the workspace correctly.

VI. CONCLUSIONS

The elasticity of industrial robots and limited force capability are barriers for achieve a successful robotic FSW which limits its use for high quality welding. Usually, this elacticity causes errors to follow the desired trajectory. There are two types of error; position error (EP) and orientation error (ER). These two errors are calculated in Cartesian space using the joint stiffness model, the kinematic and dynamic models of the manipulator. In this work a parallel structure was added to polyarticulated robot to minimize them. Optimization has been carried out using an algorithm developed in Matlab. The algorithm calculates the parameters of the parallel structural V and the three forces generated by this structure F_p to minimize the position and the orientation error during welding. For ours example we obtained a small orientation error of $0.1°$ and a very small position error of approximately zero.

Using the software Catia we estimated approximately the workspace of the serial and parallel robots by a parallelepiped. The volume of the parallelepiped equals $5.76\mathrm{m}^3$.

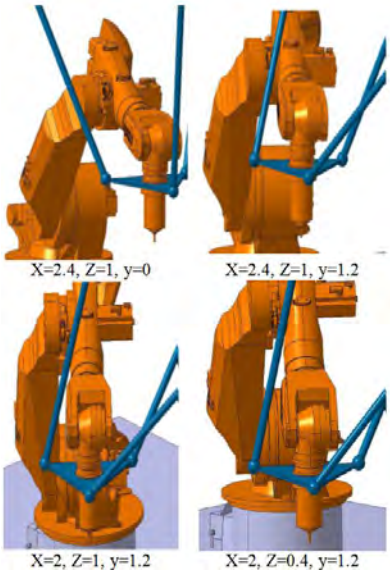


Fig. 10. the workspace of the hybrid structure

With the proposed solution, the accuracy of positioning the tool and the FSW welding performance can be greatly improved. This assist device allows to improve the stiffness of an industrial robot. This approach allows us to gain in terms of stiffness, however, it limits the workspace. Moreover the main advantage of a serial robot is its workspace. But, despite this limitation of the application flexibility, this solution can ensure a process with good accuracy for simple paths and also for complex paths as the angle between the axis of the tool and the axis z_0 stays in the workspace of the hybrid structure. So finally, the structure can be useful for many industrial applications with the advantage of high welding quality.

ACKNOWLEDGMENT

The authors would like to thank the Doctoral School SMI of Arts et metiers ParisTech and the Institut of Technical Mechanics of KIT for their support.

REFERENCES

[Balasubramanian, 2009] Balasubramanian, B. Gattu, R. S. M. (2009). Process forces during friction stir welding of aluminium alloys. 14:141–145.

[Bres et al., 2010] Bres, A., Monsarrat, B., Dubourg, L., Birglen, L., Perron, C., Jahazi, M., and Baron, L. (2010). Simulation of friction stir welding using industrial robots. *Industrial Robot*, 37(1):36–50.

[De Backer, 2014] De Backer, J. (2014). *Feedback control of robotic friction stir welding*. PhD thesis, University west.

[Deblaise et al., 2006] Deblaise, D., Hernot, X., and Maurine, P. (2006). A systematic analytical method for pkm stiffness matrix calculation. In *Robotics and Automation, 2006*, pages 4213–4219. Proceedings of the 2006 IEEE International Conference on Robotics and Automation.

[Duelen and Schröer, 1991] Duelen, G. and Schröer, K. (1991). Robot calibration: method and results. *Robotics and Computer-Integrated Manufacturing*, 8. 4.

[Dumas, 2011] Dumas, C. (2011). *Développement de méthodes robotisées pour le parachèvement de pièces métalliques et composites*. PhD thesis, Université de Nantes.

[ESAB, 2013] ESAB (2013). Boeing selects esab for space launch system project. (2013, 2013/09/06).

[Fuller 2007, 2007] Fuller 2007 (2007). *Friction Stir Welding and Processing*, chapter Chapter 2 Friction Stir Tooling: Tool Materials and Designs, Friction Stir Welding and Processing. ASM International. ISBN-13 978-0-87170-840-3.

[Gibson et al., 2014] Gibson, B., Lammlein, D., Prater, T., Longhurst, W., Cox, C., Ballun, M., Dharmaraj, K., Cook, G., and Strauss, A. (2014). Friction stir welding: process, automation, and control. *Journal of Manufacturing Processes*, 16(1):56–73.

[Khalil and Creusot, 1997] Khalil, W. and Creusot, D. (1997). Symoro+: A system for the symbolic modelling of robots. *Robotica*, 15:153–161.

[Khalil and Kleinfinger, 1986] Khalil, W. and Kleinfinger, J. (1986). New geometric notation for open and closed-loop robots. pages 1174–1179.

[Palpacelli, 2016] Palpacelli, M. (2016). Static performance improvement of an industrial robot by means of a cable-driven redundantly actuated system. *Robotics and Computer-Integrated Manufacturing*, 38:1–8.

[Qin et al., 2014] Qin, J., Leonard, F., and Abba, G. (2014). Nonlinear discrete observer for flexibility compensation of industrial robots. *IFAC Proceedings Volumes*, 47(3):5598–5604.

[Schröer, 1993] Schröer, K. (1993). Theory of kinematic modelling and numerical procedures for robot calibration. *Robot Calibration, Chapman & Hall, London*, pages 157–196.

[Siciliano and Khatib, 2016] Siciliano, B. and Khatib, O. (2016). *Springer handbook of robotics*. Springer.

[Smith, 2007] Smith, B. C. (2007). *Friction Stir Welding and Processing*, chapter 11 Robots and machines for friction stir welding / processing, Friction Stir Welding and Processing. ASM International.

[Soron, 2008] Soron, M. (2008). Towards multidimensionality and flexibility in fsw using an industrial robot system. *Welding in the World*, 52(9-10):54–59.

[Strombeck et al., 2000] Strombeck, A., Schilling, C., and Santos, J. (2000). 2D and 3D friction stir welding with articulated robot arm. In *Proc. of 2nd International Friction Stir Welding Symposium*, Gothenburg, Sweden.

[Thomas et al., 1991] Thomas, W. M., Nicholas, E. D., Needham, J. C., Murch, M. G., Temple-Smith, P., and Dawes, C. J. (1991). International patent application pct/gb92/02203. Technical report, The Welding Institute, London, UK.

[Voellner et al., 2007] Voellner, G., Zaeh, M., Silvanus, J., and Kellenberger, O. (2007). Robotic friction stir welding. Technical report, SAE Technical Paper.

[Wernholt and Östring, 2003] Wernholt, E. and Östring, M. (2003). Modeling and control of a bending backwards industrial robot. *Department of Electrical Engineering,Linköping, Sweden*, page 38.

[Zaeh and Voellner, 2010] Zaeh, M. F. and Voellner, G. (2010). Three-dimensional friction stir welding using a high payload industrial robot. *Production Engineering*, 4(2-3):127–133.

Measurement strategy for flexible assembly for adhesive bonding

Prof. Dr.-Ing. R. Müller;
Dipl.-Wirt.-Ing. (FH) M. Vette-Steinkamp, M.Eng.;
Dipl. Ing (RWTH) O. Mailahn; S. Andres, B.Eng.;
P. Becker, M.Sc. *

* *ZeMA - Zentrum für Mechatronik und Automatisierungstechnik gemeinnützige GmbH, Eschberger Weg 46 Gewerbepark, Gebäude 9, 66121 Saarbrücken, Deutschland (e-mail: p.becker@zema.de).*

Abstract: This publication focuses on the development of measurement concepts for adhesive bonding in flexible assembly cells. The quality of adhesive bonding is influenced by boundary conditions like constant velocity and distance to the surface. Classic assembly processes, like off-line robot path planning, are not reliable enough for new developed light-weight materials like CFRP because of the production tolerances. To use off-line programmed industrial robots it is required to measure the deviation from reality to the model and adapt the path before applying the adhesives.
The research project "Entwicklung **i**nnovativer **Pro**duktionstechnologien für **Groß**bauteile" (IProGro) is researching strategies to rise the flexibility of assembly cells by using model based controlling, sensitive robotic and worker assistance systems.
The project contains the following steps: planning and modelling of the cell, (re-)configuration, commissioning and optimization of the robot path. To be able to use off-line robot programming, the simulation has to be adapted to the reality. A method to align model and reality has to be developed. After the initial set-up of the cell, the position of the production equipment relative to a reference coordinate system needs to be determined. Following, the position and orientation of the robot to the part feature is measured. The last step is the determination of the surface to adapt to manufacturing deviation.

Keywords: Robot simulation, Robot control, Sensor data

1. INTRODUCTION AND PROBLEM

The development of intelligent and flexible assembly processes in times of a high number of variants, small quantities and long development circles are quite important. The research project IProGro is developing production technologies for the assembly of carbon-fiber-reinforced polymer (CFRP) for large parts. Focus is on the increase of flexibility while cost and quality (and the reliability of the process) stay constant. The improved flexibility is reached by simplifying the process of (re-)configuration of the cell. To reduce the downtime while (re-)configuration, the production equipment is only positioned roughly according to a simulation and the real position is detected. This is called **global measurement**. The assembly cell contains production equipment (like robots) which are mounted with tools to perform the process. The process is performed on a part feature, which will be measured relative to the robot to detect deviation in position and rotation (and to compensate the inaccuracy of the robot calibration). The referencing of the feature relative to the robot flange is called **regional measurement**. Classic assembly processes like off-line planned robot paths are not reliable enough for new developed light-weight materials like CFRP because of the production tolerances. Effects like spring-in make CFRP prone to deviation [Stei (2015)]. The application of adhesive bonding on such a surface is quite challenging and sometimes not reliable. In the body production of the BMW i3, nearly every second CFRP-part is scrapped because of those deviations [Mana (2014)]. To improve the process reliability, the deviation of the surface needs to be detected and measured. This **local measurement** is the outlook of this publication. All three concepts (global, regional, local) will be explained in the following chapters. All considerations will be done with regard to a robot assembly cell.

2. CLASSIFICATION OF THE MEASUREMENT CONCEPTS

Fig. 1 gives an overview of the three measurement principles (global, regional, local).

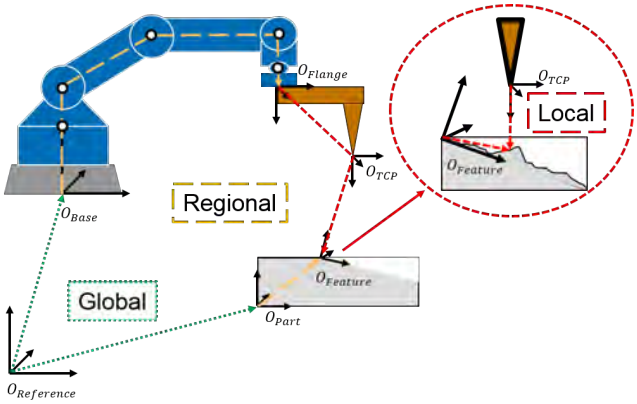


Fig. 1. Overview of the measurement principles
Global: Where is the robot relative to the work piece?
Regional: Where is the flange relative to the feature?
Lokal: Where is the TCP relative to the surface?

By setting up the robot cell, the following tolerance chain (fig. 2) can be formed. The chain can be split into a robot influenced section and a part influenced section.

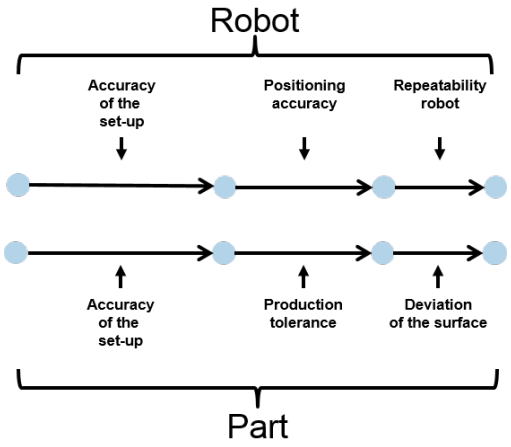


Fig. 2. Example of the tolerance chain

The purpose of the following measurement concepts is to shorten the tolerance chain to be able to perform the task reliably.

2.1 Global measurement

A robot cell contains different production equipments like the robot, working table, tools as well as the part that has to be assembled. In the work scheduling, the cell is planned.In the next step, the cell will be set up. During this process deviations between planning and reality can occur. Those deviations contain tolerances and accuracy of the set-up itself. Instead of improving the accuracy of the set-up, global measurements adapt the planned model to the real set-up. This makes the set-up process easier and reduces idle time. The Global measurement sets all production equipments in the same reference coordinate system (fig. 3).

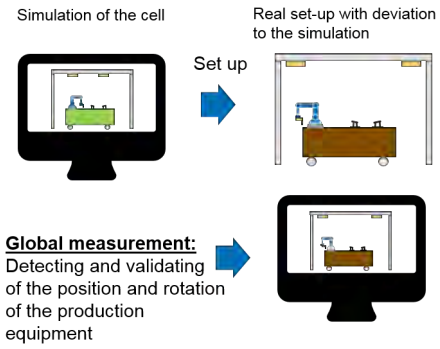


Fig. 3. Model of global measurement

By measuring and adapting the simulation the chain can be shortened. The link doesn't vanish completely since the measuring equipment has an accuracy of its own.

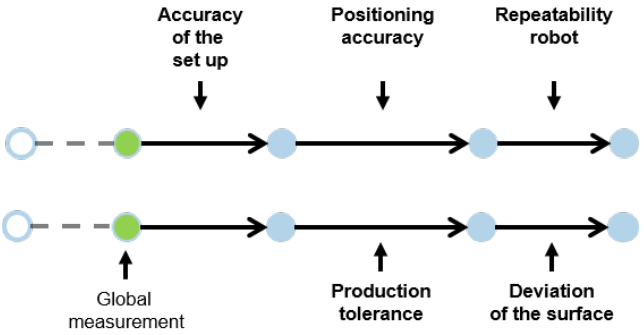


Fig. 4. Tolerance chain after global measurement

2.2 Regional measurement

The position of the part feature relative to the robot is important for the process itself. The regional measurement is needed because the position inaccuracy of the robot as well as the production tolerance of the part influence the process.

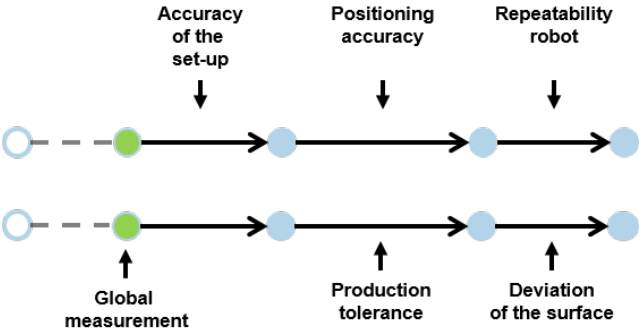


Fig. 5. Tolerance chain after regional measurement

The position accuracy of the robot is influenced severely by the calibration [Muel (2017b)]. Instead of regional measurement the tolerance chain could be shortened by improving those calibrations.

2.3 Local measurement

The local measurement is a method to optimize the robot path. It detects local deviations of the surface and adapts the programming. fig. 6 shows the general concept of the local measurement.

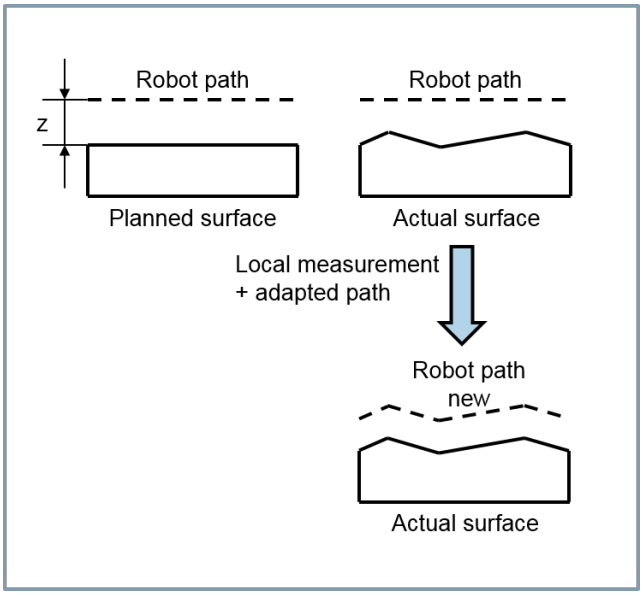


Fig. 6. Local measurement

Fig. 7 shows the influence on the tolerance.

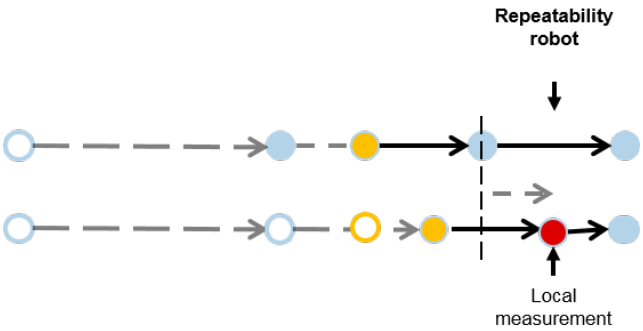


Fig. 7. Tolerance chain after local measurement

3. IMPLEMENTATION IN THE RESEARCH PROJECT IPROGRO

These concepts are being developed in the research project IProGro.

3.1 Global measurement - forward incision

The global measurement, which defines the position between the robot coordinate system and the part coordinate system is done with two projection lasers. They also serve to assist the worker with the set-up by showing where to place the production equipment in the cell (Fig. 8 top). The measurement is based on the principle of forward incision. The measurement reflectors are mounted on the production equipment (fig. 8 bottom) [Muel (2017)].

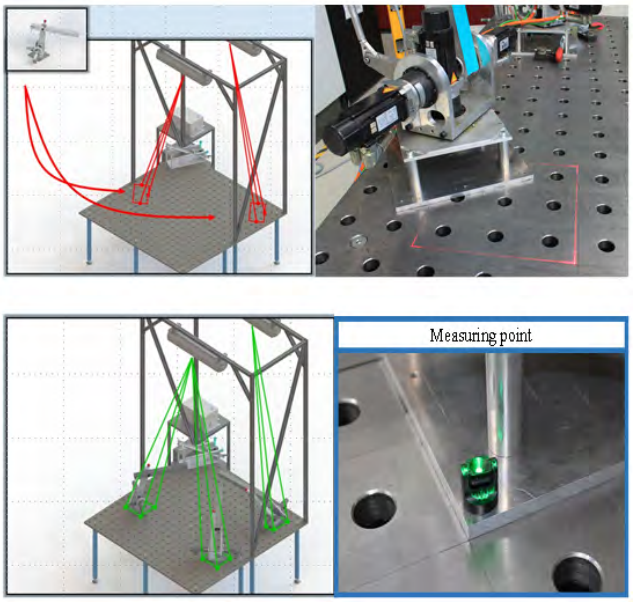
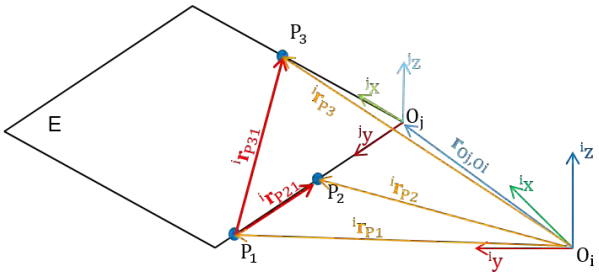


Fig. 8. Implementation global measurement

To calculate the position and orientation of a rectangular plate, three points, two on the same edge, the third on a perpendicular one, are required. When the plate is parallel to one of the three main planes of the base coordinate system, two of the three equations are not linear independent. It is recommended to always use the equation with the highest denominator (see (1),(2),(3)).



$$ {}^{i}\mathbf{r}_{P1} = \begin{pmatrix} P1_x \\ P1_y \\ P1_z \end{pmatrix} \quad {}^{i}\mathbf{r}_{P2} = \begin{pmatrix} P2_x \\ P2_y \\ P2_z \end{pmatrix} \quad {}^{i}\mathbf{r}_{P3} = \begin{pmatrix} P3_x \\ P3_y \\ P3_z \end{pmatrix} $$

Fig. 9. Flying measurement of a rectangular plate

$$ \lambda_{1a} = \frac{(P1_y - P3_y) \cdot ie_{(jx_x)} + (P3_x - P1_x) \cdot ie_{(jx_y)}}{(ie(jy_x) \cdot ie_{(jx_y)}) - (ie_{(jy_y)} \cdot ie_{(jx_x)})} \quad (1) $$

$$ \lambda_{1b} = \frac{(P1_z - P3_z) \cdot ie_{(jx_x)} + (P3_x - P1_x) \cdot ie_{(jx_z)}}{(ie(jy_x) \cdot ie_{(jx_z)}) - (ie_{(jy_z)} \cdot ie_{(jx_x)})} \quad (2) $$

$$ \lambda_{1c} = \frac{(P1_z - P3_z) \cdot ie_{(jx_y)} + (P3_y - P1_y) \cdot ie_{(jx_z)}}{(ie(jy_y) \cdot ie_{(jx_z)}) - (ie_{(jy_z)} \cdot ie_{(jx_y)})} \quad (3) $$

$$ O_j = P1 + \lambda_1 \cdot \frac{P1 - P2}{|P1 - P2|} \quad (4) $$

3.2 Regional measurement - flying measurement

Regional measurement detects part features. This can be done by using a robot guided distance sensor. By detecting edges on a plate, a reference can be calculated (see chapter 3.1 Global measurement).

3.3 Local measurement - outlook

Surface deviations effect performing tasks on a surface. Effects like arch in and tolerances or different fixation points on non-rigid parts can lead to deviations which interfere with the quality of the process. Local measurement is needed to detect those deviations and adapt the robot path accordingly. There are different methods to detect the distance. A selection can be seen in the following table (fig. 10).

Process	Range	Application	Pro	Con
Ultrasonic	3500 mm	Distance sensor, Car manufacturing	Small and cheap	Reliant on surface
Laser-autofocus	< 1mm	Semi conductor production	Can measure reflecting surfaces	Clean surface with less than 10° tilt
Chromatic	0,02 mm – 25 mm	Fill level controlling in medical applications	Small measurement errors, can measure transparent surfaces	Sensitive to tilt on the surface
Triangulation	1 mm – 1 m	In process measurement	Can measure non flat surfaces	Depending on the color of the surface

Fig. 10. Overview of measurment principles [Oubb (2012),Kefe (2015)

4. SUMMARY AND OUTLOOK

The research project IProGro is about improving the flexibility of an assembly cell. This is realized by a three step measuring concept. First, projection lasers assist the worker by setting up the assembly cell and using those lasers to measure the position of the production equipment (global measurement). Second, the position of the part feature is detected by a robot guided distance sensor to reference the feature to the robot (regional measurement). In the last step, the surface is scanned to determine deviations and adapt the robot path if necessary (local measurement).

REFERENCES

[Mana(2014)] Managermagazin 2014 *In: http://www.bimmertoday.de/2014/02/20/bmw-i3-produktion-carbon-probleme-leipzig-ausschuss.* (Stand: 30.05.2017).

[Plas(2017)] AirbusA350XWB Kohlefaser-Verbundwerkstoff *http://www.plastverarbeiter.de/15818/airbus-a350-xwb-besteht-zu-53-aus-kohlefaser-verbundwerkstoffen/.* (Stand: 30.05.2017).

[Stei(2015)] Toleranzmanagement für Bauteile aus kohlenstofffaserverstärktem Kunststoff - Ursachen der geometrischen Streuung, präventive Vorhersagen der Maßhaltigkeit und der Einsatz des Werkstoffes im Rohbau *KIT-Bibliothek.*

[Muel (2017)] Müller, Rainer and Ball, Jan Assembly assistance and position data feedback by means of projection lasers *19.06.2017.*

[Pfei (2010)] Pfeifer, Tilo and Schmitt, Robert Fertigungsmesstechnik. *19.06.2017.*

[Kefe (2015)] Keferstein, Claus P and Marxer, Michael Fertigungsmesstechnik Praxisorientierte Grundlagen, moderne Messverfahren *19.06.2017.*

[Oubb (2012)] Oubbati,Mohamed Einführung in die Robotik: Sensoren, *19.06.2017.*

[Muel (2017)] Müller,Rainer and Kanso, Ali Methodology for design of mechatronic robotic manipulators based on suitability for modern application scenarios *19.06.2017.*

ACKNOWLEDGMENTS

This paper was written in the framework of the research project IProGro which is funded by the European Union (EU). The project is part of initiative “Investment in growth and jobs”, which takes place within the structural fund for development (EFRE). It is supervised by the state of Saarland.

EUROPÄISCHER FONDS FÜR REGIONALE ENTWICKLUNG
Investition in Ihre Zukunft
Saarland
Ministerium für Wirtschaft und Wissenschaft
Gefördert mit Mitteln des Saarlandes und der Europäischen Union

www.robotix.academy