INTELLI 2020

The Ninth International Conference on Intelligent Systems and Applications

ISBN: 978-1-61208-798-6

October 18 – 22, 2020

INTELLI 2020 Editors

Gil Gonçalves, University of Porto, Portugal

INTELLI 2020

Foreword

The Ninth International Conference on Intelligent Systems and Applications (INTELLI 2020), held between October 18–22, 2020, was an inaugural event on advances towards fundamental, as well as practical and experimental aspects of intelligent and applications.

The information surrounding us is not only overwhelming but also subject to limitations of systems and applications, including specialized devices. The diversity of systems and the spectrum of situations make it almost impossible for an end-user to handle the complexity of the challenges. Embedding intelligence in systems and applications seems to be a reasonable way to move some complex tasks form user duty. However, this approach requires fundamental changes in designing the systems and applications, in designing their interfaces and requires using specific cognitive and collaborative mechanisms. Intelligence became a key paradigm and its specific use takes various forms according to the technology or the domain a system or an application belongs to.

We take here the opportunity to warmly thank all the members of the INTELLI 2020 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to INTELLI 2020. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the INTELLI 2020 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that INTELLI 2020 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of intelligent systems and applications.

INTELLI 2020 Chairs

INTELLI 2020 Steering Committee

Carsten Behn, Schmalkalden University of Applied Sciences, Germany
Leo van Moergestel, HU University of Applied Sciences Utrecht, Netherlands
Stefano Berretti, University of Firenze, Italy
Gil Gonçalves, University of Porto, Portugal

INTELLI 2020 Publicity Chair

Jose M. Jimenez, Universitat Politecnica de Valencia, Spain
Jose Luis García, Universitat Politecnica de Valencia, Spain

INTELLI 2020 Industry/Research Advisory Committee

David Greenhalgh, University of Strathclyde, Glasgow, UK
Paolo Spagnolo, National Research Council, Italy
Luca Santinelli, ONERA Toulouse, France
Lars Braubach, Hochschule Bremen, Germany
Maiga Chang, Athabasca University, Canada
Chin-Chen Chang, Feng Chia University, Taiwan

INTELLI 2020

Committee

INTELLI 2020 Steering Committee

Leo van Moergestel, HU University of Applied Sciences Utrecht, Netherlands
Stefano Berretti, University of Florence, Italy
Carsten Behn, Schmalkalden University of Applied Sciences, Germany
Gil Gonçalves, University of Porto, Portugal

INTELLI 2020 Publicity Chair

Jose M. Jimenez, Universitat Politecnica de Valencia, Spain
Jose Luis García, Universitat Politecnica de Valencia, Spain

INTELLI 2020 Industry/Research Advisory Committee

David Greenhalgh, University of Strathclyde, Glasgow, UK
Paolo Spagnolo, National Research Council, Italy
Luca Santinelli, ONERA Toulouse, France
Lars Braubach, Hochschule Bremen, Germany
Maiga Chang, Athabasca University, Canada
Chin-Chen Chang, Feng Chia University, Taiwan

INTELLI 2020 Technical Program Committee

Azizi Ab Aziz, Universiti Utara Malaysia, Malaysia
Ari Aharari, SOJO University, Japan
Bilal Ahmad, University of Warwick, UK
Zaher Al Aghbari, University of Sharjah, UAE
Sarra Alqahtani, Wake Forest University, USA
Rachid Anane, Coventry University, UK
Benjamin Aziz, University of Portsmouth, UK
Arvind Bansal,Kent State University, USA
Suzanne Barber, The University of Texas at Austin, USA
Carmelo Bastos-Filho, University of Pernambuco, Brazil
Carsten Behn, Schmalkalden University of Applied Sciences, Germany
Fayçal Bensaali, Qatar University, Qatar
Lyes Benyoucef, Aix-Marseille University, France
Giuseppe Berio, IRISA | Université de Bretagne Sud, France
Stefano Berretti, University of Firenze, Italy
Jonathan Berrisch, University of Duisburg-Essen, Germany
Francisco Bonin Font, University of the Balearic Islands, Spain
Frederic Bousefsaf, LCOMS | Université de Lorraine, France

Lars Braubach, Hochschule Bremen, Germany
Ramon F. Brena, Tecnologico de Monterrey, Mexico
Simeon C. Calvert, Delft University of Technology, Netherlands
Valérie Camps, Paul Sabatier University | IRIT, Toulouse, France
Laura Carnevali, University of Florence, Italy
Carlos Carrascosa, Universitat Politècnica de València, Spain
Cesar Castelo-Fernandez, Institute of Computing | University of Campinas, Brazil
Dario Cazzato, Centre for Security, Reliability and Trust | University of Luxembourg, Luxembourg
Chin-Chen Chang, Feng Chia University, Taiwan
Guilherme Conde, Federal University of Western Pará, Brazil
Angelo Croatti, University of Bologna, Italy
Edgar Giovanny Cuzco Silva, Universidad Nacional de Chimborazo, Ecuador
Mohammed Dahane, Université de Lorraine, France
Chuangyin Dang, City University of Hong Kong, Hong Kong
Angel P. del Pobil, Jaume I University, Spain
Alessandra Scotto di Freca, Università di Cassino e del Lazio Meridionale, Italy
Andrea D'Ariano, Roma Tre University, Italy
Arianna D'Ulizia, National Research Council - IRPPS, Italy
Ahmed Ewais, Arab American University, Jenin, Palestine / Vrije Universiteit Brussel, Belgium
Ana Fernández Vilas, School of Telecommunication Engineering | University of Vigo, Spain
Manuel Filipe Santos, University of Minho, Portugal
Marta Franova, CNRS | LRI & INRIA, France
Todorka Glushkova, Plovdiv University "Paisii Hilendarski", Bulgaria
Gil Gonçalves, University of Porto, Portugal
Sérgio Gorender, Federal University of Bahia, Brazil
Emmanuelle Grislin-Le Strugeon, LAMIH | Université Polytechnique Hauts-de-France (UPHF), France
Ibrahim A. Hameed, Norwegian University of Science and Technology (NTNU), Norway
Wladyslaw Homenda, Warsaw University of Technology, Poland
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Wei-Chiang Hong, School of Education Intelligent Technology - Jiangsu Normal University, China
Martin Horauer, University of Applied Sciences Technikum Wien, Austria
Matin Hosseini, University of Louisiana at Lafayette, USA
Christopher-Eyk Hrabia, Technische Universität Berlin | DAI-Labor, Germany
Shah Rukh Humayoun, San Francisco State University, USA
Chih-Cheng Hung, Kennesaw State University - Marietta Campus, USA
Syed Muhammad Zeeshan Iqbal, BrightWare LLC, Riyadh, Saudi Arabia
Zahid Iqbal, University of Porto, Portugal
Ajune Wanis Ismail, Universiti Teknologi Malaysia, Malaysia
Raheleh Jafari, School of Design | University of Leeds, UK
Anubhav Jain, Telstra, India
Juergen Jasperneite, Fraunhofer IOSB-INA, Germany
Andrés Jiménez Ramírez, University of Seville, Spain
Maria João Ferreira, Universidade Portucalense, Portugal
Janusz Kacprzyk, Systems Research Institute - Polish Academy of Sciences, Poland
Ryotaro Kamimura, Tokai University, Tokyo, Japan
Stelios Kapetanakis, University of Brighton, UK
Mehdi Kargar, Ted Rogers School of Management | Ryerson University, Canada
Alexey Kashevnik, SPIIRAS, Russia

Simeon Keates, Edinburgh Napier University, UK
Sotiris Kotsiantis, University of Patras, Greece
Tobias Küster, DAI-Labor / Technical University of Berlin, Germany
Victoria Lapuerta, Universidad Politécnica de Madrid, Spain
Antonio LaTorre, Universidad Politécnica de Madrid, Spain
Frédéric Le Mouël, Univ. Lyon / INSA Lyon, France
Deok-Jin Lee, Kunsan National University, South Korea
George Lekeas, City Universty - London, UK
Chanjuan Liu, Dalian University of Technology, China
Daniela López De Luise, CI2S Labs, Argentina
René Meier, Hochschule Luzern, Germany
António Meireles, GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Portugal
Jérôme Mendes, Institute of Systems and Robotics (ISR-UC), Portugal
Jair Minoro Abe, Paulista University & Institute of Advanced Studies | University of São Paulo, Brazil
Dusmanta Kumar Mohanta, Birla Institute of Technology, India
Jose M. Molina, Universidad Carlos III de Madrid, Spain
Vítor Monteiro, University of Minho, Portugal
Ceci Morales, iRobot, USA
Fernando Moreira, Universidade Portucalense Infante D. Henrique, Portugal
Debajyoti Mukhopadhyay, Mumbai University, India
Filippo Neri, University of Naples, Italy
Pranav Ajeet Nerurkar, NMIMS University, Mumbai, India
Dinh-Luan Nguyen, Michigan State University, USA
Thanh-Tuan Nguyen, HCMC University of Technology and Education, HCM City, Vietnam / University of Toulon, CNRS, LIS, Toulon, France
Alex Norta, Tallinn University of Technology, Estonia
Cyrus F. Nourani, akdmkrd.tripod.com, USA
Kenneth Nwizege, Ken Saro-Wiwa Polytechnic, Nigeria
Michel Occello, Université Grenoble Alpes, France
Krzysztof Okarma, West Pomeranian University of Technology in Szczecin, Poland
Ana Oliveira Alves, CoimbraPolytechnic- ISEC & Centre of Informatics and Systems of the University of Coimbra - CISUC, Portugal
Joanna Isabelle Olszewska, University of West Scotland, UK
Marcin Paprzycki, Systems Research Institute / Polish Academy of Sciences - Warsaw, Poland
Goharik Petrosyan, International Scientific-Educational Center of the National Academy of Sciences, Yerevan, Armenia
Agostino Poggi, Università degli Studi di Parma, Italy
Marco Polignano, University of Bari "Aldo Moro", Italy
Filipe Portela, University of Minho, Portugal
Catia Prandi, University of Bologna, Italy
Dilip Kumar Pratihar, Indian Institute of Technology Kharagpur, India
Radu-Emil Precup, Politehnica University of Timisoara, Romania
Shahnawaz Qureshi, National University of Computer and Emerging Sciences, Pakistan
Ahmed Rafea, American University in Cairo, Egypt
Giuliana Ramella, National Research Council (CNR) -Institute for the Applications of Calculus "M. Picone" (IAC), Italy
Aurora Ramírez, University of Córdoba, Spain

Rabiâ Riad, Ibn Zohr University, Morocco
Fátima Rodrigues, Institute of Engineering | Polytechnic of Porto, Portugal
Daniel Rodriguez, University of Alcalá, Spain
Amirreza Rouhi, Politecnico di Milano, Italy
Alexander Ryjov, Lomonosov Moscow State University | Russian Presidential Academy of National Economy and Public Administration, Russia
Fariba Sadri, Imperial College London, UK
Mohammad Saeid Mahdavinejad, Kansas StateUniversity, USA
Lorenza Saitta, Università del Piemonte Orientale, Italy
George E. Sakr, St Joseph University of Beirut, Lebanon
Bilal Abu Salih, Curtin University, Australia
Demetrios Sampson, Curtin University, Australia
Christophe Sauvey, LGIPM | Université de Lorraine, France
Francisco Souza, Oncontrol Technologies, Portugal
Nick Taylor, Heriot-Watt University, UK
Mark Terwilliger, University of North Alabama, USA
Supphachai Thaicharoen, Srinakharinwirot University, Bangkok, Thailand
Pei-Wei Tsai, Swinburne University of Technology, Australia
Berna Ulutas, Eskisehir Osmangazi University, Turkey
Paulo Urbano, Universidade de Lisboa - BioISI, Portugal
Leo van Moergestel, HU University of Applied Sciences Utrecht, Netherlands
Jan Vascak, Technical University of Kosice, Slovakia
Yifei Wang, Georgia Institute of Technology, USA
Kanoksak Wattanachote, Guangdong University of Foreign Study, China
Dietmar Winkler, TU Wien | Institute of Information Systems Engineering | CDL-SQI, Austria
Stefanie Wuschitz, Miss Baltazar's Laboratory, Vienna, Austria
Mudasser F. Wyne, National University, USA
Maria Gabriella Xibilia, University of Messina, Italy
Peng Xu, Technical University of Munich (TUM), Germany
Longzhi Yang, Northumbria University, UK
George Yee, Aptusinnova Inc. / Carleton University, Canada
Leila Zemmouchi-Ghomari, Ecole Nationale Supérieure de Technologie, ENST, Algiers, Algeria
Shengxin Zhu, Xi'an Jiaotong-Liverpool University, Suzhou, China

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Roadmap-based Planning in Human-Robot Collaboration Environments

Zahid Iqbal, João Reis, Gil Gonçalves

SYSTEC, Research Center for Systems and Technologies
Faculty of Engineering, University of Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
Email: {zahid,jpcreis,gil} @fe.up.pt

***Abstract*—Enabling humans and robots to work together allows for cost savings and workplace efficiency. The robot must be equipped to perceive humans, and redirect its actions for co-operative tasks or under hazardous situations. Thus, dynamic motion planning appears as an essential exercise. We use dynamic roadmaps for online motion planning in changing environments combined with a voxel-based grid. The presented approach can answer path planning queries efficiently. Visualized simulation is an important technique for rapid verification of algorithms or prototypes. We present an architecture that implements a simulation of the robotic manipulator using the Robot Operating System (ROS) and MoveIt.**

Keywords–robotics technology, intelligent robotics, sampling-based planning, perception and sensing, dynamic environments, voxel-based grid.

I. INTRODUCTION

Today, there are opportunities for automation in many industries, such as automotive assembly lines, material transport, space exploration. Frequently, robots support this automation, improving thus the production volumes, bringing down costs, and improving the precision and accuracy of the production process. Additionally, as robots capabilities increase, they can take on jobs that might be impossible or dangerous for humans [1]. The ability to calculate and execute motion plans is central to the operation of these robots. Fundamental motion planning problem is to compute a set of inputs to the robot that drive it from a start to a goal position while avoiding collisions with the environment obstacles. Early works on robot motion planning assumed a robot's world to be known and fixed at the time of planning. During the execution of the plan, however, the robot might discover a deviation from the assumed world state, such as a human entering its operational area. Typical industrial assembly environments, that confined robots to separate operation spaces isolated from human workers, would bring robot operation to a halt under such scenario [2], safety being the primary concern. In other cases, the robot must replan its trajectory from scratch, which might be computationally intensive for real-time operation.

Random sampling is an important technique employed by popular planning algorithms such as Probabilistic Roadmaps (PRM) [3], Rapidly Exploring Random Trees (RRT) [4] to build network of feasible paths. For static environments, these methods can efficiently make and execute plans, even when there are obstacles. With some of these algorithms, such as PRM that are *multi-query*, we can delegate roadmap building as an offline exercise and run multiple queries online on the available roadmap. However, we can realize a better potential of the robots by operating them in unstructured open environments that change dynamically. A robot would only possess partial information of surroundings before it commences operation; obstacles could appear or go away at any time during its operation. This situation poses several challenges for motion planning [5]. With efficient planning, we can leverage the benefits of cooperation by assigning specific production tasks to robot and humans, as well as make the robot more autonomous. Specifically, it requires the ability to account for new obstacles or changes in the environment and continue to plan on the fly and quickly with some accuracy.

This work is an incremental exercise concerning building a collabarative planning solution. We define collaboration as working as a team to reach a common goal. In the collaborative environment, physical contact between human and robot may be inevitable and indeed desirable, enabling the robot to learn from experience. To achieve accurate and efficient collaboration, both robot and human need to perceive each other's intentions and must know their task set. With that knowledge, the robot can plan and adapt its actions accordingly, ultimately leading to the achievement of the mutual objective, all the while assuring safety. A typical scenario could be the robot as a work assistant that can hand in the next tool piece to the human. Robot predicts that this piece is the one needed for the person to continue its job and is shelved and outside human's access. Collaboration scenario primarily comprises a dynamic environment, and any motion planning strategy that tackles the dynamic obstacles must allow dealing with the collaboration episodes accordingly. To this end, different approaches exist, such as Incremental Path Planning [6] and Experienced based Motion Planning (EBMP) [7]. The later employs a database of previous searches and recalls an experience from the database when it encounters a similar planning exercise. This may involve *repairing* the previous plan and reduce the time of the current query as opposed to planning from scratch. Likewise, incremental search methods use previous search results to solve similar path planning problems faster.

Our work concerns path planning with obstacle avoidance in dynamic collaboration scenarios. For efficient collaboration, an intelligent monitoring system is inevitable [8, 9]. The present work uses the monitoring solution in [8] and combines with a PRM based approach, dynamic roadmaps. We organize the work presented here in two distinct parts. The first part explains dynamic motion planning for robots and reviews some strategies that address dynamic environments (Section II). In the second part (Section IV), we present the tools that allow developing motion planning algorithms on a simulated model

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

of the robot. In Section III, we describe how we integrate the monitoring solution with planning algorithm for efficient motion planning. In Section V, we present the preliminary path planning architecture using the popular tool support and based on a voxelized grid. Finally, in Section VI, we conclude the paper and present some future work directions.

II. MOTION PLANNING IN DYNAMIC AND UNKNOWN ENVIRONEMNTS

The work presented here concerns industrial robotic arms or so-called manipulators. In particular, we do not consider mobile robots; proposed ideas will be applicable to trajectory planning of a Universal Robot with a lift ability of 5 kg (UR5) [10]. Mounted on a stationary platform, its links with revolute joints and end-effector can move with a certain degree of freedom.

A. Definitions

Fundamental to the path planning problem is the concept of *configuration* and *configuration space*. The configuration of a robot is a set of independent parameters that characterize the position of every point in the object whereas configuration space (denoted $\mathcal{C}$) is the space of all possible configurations [11]. The dimension of the configuration space is determined by the degrees of freedom of the robot. A configuration q is a vector of robot joint values. For instance, a robotic arm with six revolute joints has six degrees of freedom and its configuration is denoted $q = \{q_1, q_2, q_3, q_4, q_5, q_6\}$ where q_i denotes i^{th} joint angle. The configuration space $\mathcal{C}$ is a space of these configurations, i.e., for 6 dimensions, we have $\mathcal{C} = R^6$. Any path finding strategy places configurations in $\mathcal{C}$ into two categories, those that are free $\mathcal{C}_{free}$, and others that are in collision $\mathcal{C}_{forb}$, i.e., occupied by obstacles. A configuration $q \in \mathcal{C}_{free}$ if the robot placed at q does not intersect with workspace obstacles. A *path* is a continuous sequence of configurations in $\mathcal{C}_{free}$ connecting initial and goal configurations. For dynamic scenarios, a notion of time is incorporated in $\mathcal{C}$, and resulting space may be termed as *state-time* space [12]. It consists of pair (x, t) where $x \in \mathcal{C}$ and t denotes the time. A path obeying dynamic constraints is termed as *trajectory*.

B. PRM based approaches for dynamic environments

PRM approach, in the planning phase, randomly samples configurations in the free space and builds a roadmap, i.e., $\mathcal{C}_{free}$. In the query phase, it finds paths in the roadmap to guide robot from start to the goal. Dynamic obstacles in the environment can invalidate parts of the previously constructed roadmap in what concerns free space. Therefore, we need methods to update the roadmap taking into consideration the current state.

The works in [13] and [12] use PRM-based approach to handle dynamic scenarios. The work in [12] extends PRM by augmenting configuration space with a dimension of time. The preprocessing step is like regular PRM. Online, checking for static obstacles is not required for collision testing. It incorporates the notion of *free interval*, the maximum continuous segment of time that a configuration (a roadmap node) is collision-free. Arriving earliest in a free interval allows reaching the goal faster. Implicitly, this approach builds a free interval graph that corresponds to the roadmap built in the preprocessing phase, i.e., each vertex in free interval graph maps to one node in the roadmap. The approach builds a local trajectory first, i.e. traversing a single arc between two roadmap nodes, and then extends to the complete interval graph. Robot movement velocity is given maximum v_{max}. The approach interpolates motion along a single arc using a two-dimensional state-time grid where distance travelled along the arc and time are two dimensions. The state-time space is discretized by time step τ and velocity value $v_p < v_{max}$; l is the length of the arc. From a given state (x, t), three states are reachable (Figure 1).

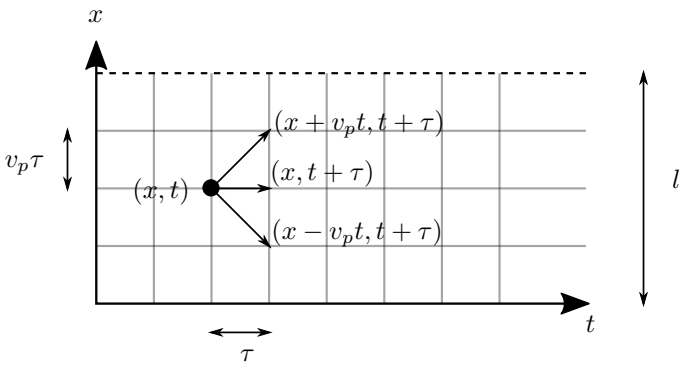


Figure 1. A state-time grid of a single roadmap arc [12].

For an arc a with start state-time (x_s, t_s) and destination state x_d, the approach maintains a stack where initially (x_s, t_s) is present. In a loop, it pops a stack element. If it is collision-free and not visited before, it pushes the reachable grid points in an order that favours movement towards the goal. The algorithm runs until the stack is empty or goal state x_d has been reached. Backpointers and the visit status of each state are maintained. So if a certain movement encounters an obstacle, we move back to the next unexplored state on the stack. The algorithm chooses a short time step τ for better accuracy of the algorithm. For finding a global trajectory, the algorithm sends $probes$ corresponding to single arcs and evaluates a function, $f(p) = g(p) + h(p)$, to determine how promising a given probe p is. $g(p)$ is the cost so far and $h(p)$ is the estimate until the goal state. Assuming (x_u, t_u) to be the top state on the stack for probe p, the trivial estimate for $g(p)$ is t_u. To estimate $h(p)$, it uses a roadmap distance function $D(x_u.g)$ from the current state x_u until the goal state g. This estimate could come from running a simple Dijkstra's algorithm on the roadmap before the query phase, giving $h(p) = D(x_u, g)/v_{max}$. The probe with minimum $f(p)$ value has the highest priority. For a start state s, the algorithm initializes probes on all arcs starting from s and puts them on the priority queue, from which it examines probes one by one. If the local trajectory found by a prob reaches the goal, we have a solution. Otherwise, it starts probes on outgoing edges and continues. The algorithm terminates either when it finds the goal node or if no probes are remaining.

The work in [13] start their approach alike the [12], i.e., building the roadmap in the preprocessing phase, accounting for static obstacles. In the query phase, at first, the approach connects query nodes to the roadmap. Next, it divides roadmap nodes into three categories, those that are reachable from the start node, those that are reachable from goal node, and those that are not reachable from query nodes. A first attempt to

find path labels some edges as blocked; edges that are not collision-free due to dynamic obstacles. An essential element of this approach is the connectivity extension in two ways, from a query node to a node reachable from the other query node, and from query nodes to the roadmap portion which is not yet connected to query nodes. The approach searches for a path with enhanced connectivity. When the approach cannot retrieve path from the roadmap, it uses RRT-connect [14] which incrementally builds two random trees one rooted at the start node and the other rooted at the goal node, and tries to find a path. Finally, it can create new nodes if the existing roadmap does not allow any path. This method can store the key positions of the moving obstacle in a database. For subsequent edge labelling, it compares the current position of an obstacle to the stored value. If it's the same, the previous exercise (collision testing, and pathfinding) is still valid, thus, avoiding new effort. Since dynamic obstacles can move at any time different positions, so, practically, there can be infinite positions that must be stored. For efficient memory usage, the approach stores only the last checked positions. However, this approach is beneficial in certain situations; for example, when moving obstacles are doors that could be closed or open, then storing a few positions might save significant future recalculations.

C. Experienced based motion planning

Experience-based approach to motion planning builds on knowledge from past planning exercises, i.e., [7]. It maintains a database of past experiences from which it can retrieve a partial or complete solution to the current problem. Such an approach may be particularly beneficial in situations where a robot is supposed to be operational in a fixed environment for long periods. In such a situation, the robot would frequently encounter similar obstacles and environments. Thus, planning from scratch is not desirable because of redundant computation effort. Repairing an available solution, instead, is faster. The approach maintains a sparse roadmap for efficient memory usage. Further, for speed up, it delays the collision checking until after it has found a candidate path. Later, it tests the path for collisions and removes any invalid segments. When the path gets broken, the search is commenced. In a post-processing step, it smoothes repaired paths before sending these to the robot. The approach discretizes the solution path into an ordered set of vertices and incorporates it into the experience roadmap. In the query phase, it follows the general approach, connecting query nodes to the graph, if this is successful, then A* algorithm can return the optimal path between start and goal nodes. As stated, it checks for possible collisions with environmental obstacles after it has found the candidate path. For this case, invariant constraints do not lend any recomputation. Moving obstacles, however, can appear and invalidate path segments. For this case, a new A* search is run to find the path. The approach to finding a path through disconnected components is similar to that of [13].

D. Incremental path planning

Incremental search methods reuse recent solutions to speed up searching for similar problems. Assuming that we have incomplete information about the world, replanning occurs from the current state until the goal whenever an obstacle encounters. When we have to execute the next action in the plan, we receive changes, if any, in the graph. This information could come from different sensors outside of the robot, such as laser scanners or RGB cameras. Using this information, we update the previous map instead of replanning from scratch. We continue to perform this loop until we reach the goal (Figure 2).

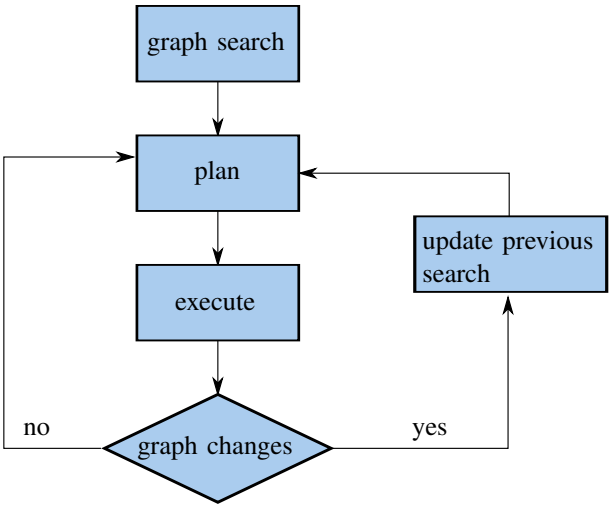


Figure 2. General methodology of incremental path planning.

In the following, we briefly discuss an incremental planning algorithm D* [15].

1) D algorithm:* D* is an incremental search algorithm, considered to be the dynamic version of A* algorithm [16] since it permits edge costs to change as the algorithm runs. The algorithm begins at the goal state donated G and works its way back to the start state; on its way, it keeps on reiterating and updating costs to reflect optimal paths. Each graph node can be in one of the following states NEW, CLOSED, OPEN, RAISE or LOWER. The OPEN list maintains nodes needing expansion. The algorithm runs until the current robot state is CLOSED. The algorithm maintains two cost estimates for each node, namely path cost function $h(G, X)$ and key function $k(G, X)$. Path cost function gives the current estimate of the sum of arc costs from state X to goal state G, and key function denotes the minimum of $h(G, X)$ which we can regard as the previous estimate of $h(G, X)$. The algorithm sorts the OPEN list according to nodes key values. When expanding a node on the OPEN list, the key function allows changing the current node state as RAISE or LOWER. When $k(G, X) < h(G, X)$, current estimate indicates an increased cost, so X state is changed as RAISE, otherwise as LOWER. Next, it propagates changes from expansion to the neighbours that are now placed on the OPEN list and expanded in turn. Each node has a back pointer which leads to the target. Algorithm reports a solution path when the next node for expansion is the start node. We follow back pointers from the start until the goal.

There have been different improvements to the original D* such as D* Lite [17], and Focussed D* [18]. Focussed D* is an informed incremental heuristic search algorithm that combines ideas from A* and D*. D* Lite, though not directly based on D*, implements similar ideas to path planning in unknown environments as D*. With lower implementation overhead, it yields better performance.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

III. Proposed approach

In this section, we describe the approach that makes use of the voxel-based grid for motion planning. We envisage using a roadmap based approach for its simplicity and ease of development. This is an extension of previous work [19], now addressing the dynamic environment.

A. Dynamic roadmap with voxel-based occupancy grid

A voxel represents a value on a regular grid in three-dimensional space, useful for many applications such as Dynamic Roadmaps [20] to create a mapping from an area in the workspace to states in $\mathcal{C}$. The work in [8] presents a monitoring approach that outputs a voxel grid. The voxel-based grid renders the entire scene of the collaborative environment as a grid composed of cubes with a given dimension, known as voxels. We can describe the granularity of the grid with the size of one side of the cube. The resolution can be set higher or lower by choosing the dimension of the unit cube in the grid, i.e., for a higher resolution, we choose a smaller size of the voxel, thus corresponding to more voxels in the grid, and vice versa. The resolution of the grid presents a tradeoff between the accuracy of the planning queries and computation effort. The monitoring solution can also label the workspace objects. When voxels in the grid are labelled, we can identify, at first, if an object in the collaborative environment occupies the voxel, and secondly, which type of object, i.e., either a robotic part itself, human, or an obstacle object. A labelled voxel-based grid can efficiently solve planning queries, in particular, in the presence of static obstacles. In the query phase, before the search commences, occupied voxels would indicate $\mathcal{C}_{forb}$ and the remaining nodes would make the $\mathcal{C}_{free}$.

In our system, we need an accurate mapping of workspace objects from the voxel grid to the coordinates that planning algorithms can work with. Monitoring solution considers the origin $(0, 0, 0)$ to be at the base-center of the robotic arm. For the tools used in this work (section IV), every point on the robotic arm, and in the workspace can be calculated in reference to the planning-frame, which is the frame of the "base-link" of the arm, and its origin is at position $(0, 0, 0)$. When we have a common reference point, and with some additional information from the voxel grid such as dimensions of individual voxels as well as of the hyper-cube that represents the complete voxel grid, we can map the goal positions and obstacle positions from the voxel grid into the planning solution, which allows solving subsequent path-planning queries efficiently. Figure 3 shows an example voxel grid with extreme-points and dimensions identified.

However, for dynamic obstacles, we incorporate further information, such as input from prediction algorithms dictating the temporal occupancy status of voxels. Accounting for dynamic scenarios, here we stipulate two possible ways to handle it. We can have an additional dimension of time to the voxel-based grid. And, we can associate a probability of occupancy pr_{occ} for any voxel c at any given time t, i.e., $pr_{occ}(c, t) \in [0, 1]$. This input, might come from the perception sensor, and aid in redirecting paths of the manipulator in the presence of dynamic obstacles.

Secondly, we employ ideas from the reviewed approaches, in particular, those based on random roadmaps in combination with voxel-based grid for dynamic path planning of robotic manipulator. Consider the roadmap R and the voxel-based

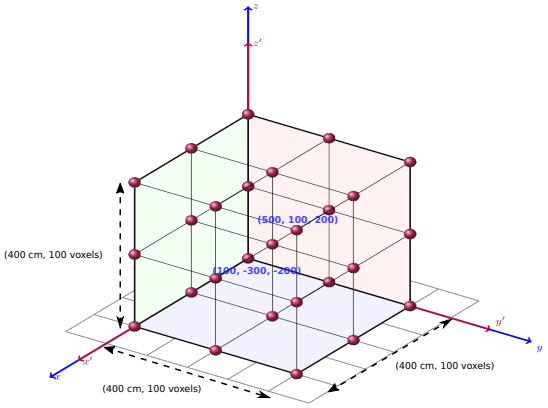


Figure 3. An example voxel grid.

grid G. The nodes in R are configurations q_i in C_{free}, and cells of grid G are denoted c. For each c in G, we check configurations q_i representing a position of the end-effector and that fall in c. With a large number of cells in G having many such configurations, we can stop computing the roadmap R, lending a uniform distribution of end-effector positions over the workspace. An occupied c invalidates the respective q in R. Obstacles appearing, disappearing, or changing their positions represent dynamic scenarios. The grid G can inform us when a cell gets occupied or becomes free. The scene can update based on actions of an external agent, e.g., a human that shares the workspace with the robot. Whenever one or more objects occupy a cell, a reference counter keeps increasing in respective node or edge in R, denoting invalid nodes. The approach keeps renewing the state of some nodes or edges as these become occupied or free. An RRT algorithm is used when the path gets broken, and the algorithm cannot compute the query.

IV. Tool Support for developing motion planning algorithms

In this section, we briefly describe the tools that we have used to develop motion-planning solution.

A. Robot Operating System (ROS)

ROS [21] provides a framework to develop and test robot software as well as to deploy such software to the real robots. As the scope of the robots has significantly grown, a fully functional code must contain software for hardware drivers, perception algorithms, abstract reasoning, trajectory planning, control and more. For any single user, it might not be possible to cover all aspects of software development for a robot. For the same reason, the final software architecture would incur a significant integration effort. The main objective of ROS is to manage the complexity of software development for robots. It contains helpful tools and open software packages for perception, navigation, transforms and simulations, allowing rapid prototyping of software for experiments. Emulating the motion of robots and the world in a virtual workspace without physically depending on an actual robot saves cost and time. ROS supports development in different programming languages, mainly C++ and Python.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

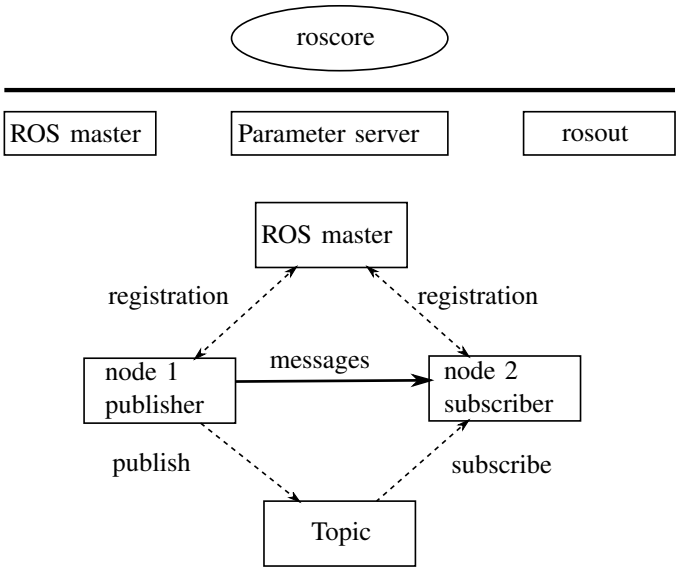


Figure 4. ROS core communication system and an example of nodes publish & subscribe to topics.

Conceptually, the computation graph within ROS is the peer-to-peer network of ROS processes that are processing data together. *roscore* is the core of the system providing a collection of nodes and programs. roscore contains three main functional module suits, which are *ROS Master*, *Parameter Server* and *rosout* logging node. ROS Master provides naming and registration to nodes in the ROS system; it allows nodes to find each other. A *node* is an executable unit of code that implements a specific functionality, similar to a software module in traditional usage. A typical system within ROS comprises many nodes and is a graph where links represent nodes communication. ROS has a publish-subscribe model of communications. Nodes communicate with each other using *topics* as communication channels. Nodes publish to a topic to send *messages* and subscribe to it for receiving messages. There can be multiple publishers and subscribers of a given topic (Figure 4). The basic message types include boolean, float, integer or string; users may use custom types as well.

B. URDF

Unified Robot Description Format (URDF) is an XML format for describing a robot model in ROS. Different ROS programs can interpret the URDF model of physical robot, e.g., tools such as *rviz* that allows to visualize it. In the most basic form, URDF describes the topology of the robot, listing its joints and links, where a joint connects two links. Topology makes a tree structure, where each link has precisely one parent, but it can have multiple child nodes. The geometry of the robot is represented through coordinate frames and transforms between these frames. Transforms have a translation and rotation components; translation measured in meters is specified in coordinate axes x, y and z whereas rotations in radians along x, y and z-axis are known as roll, pitch and yaw respectively. For ROS system to know about robot model, a launch script loads URDF file to the parameter server. The same script publishes joint state values. The kinematics libraries within ROS can read joint states, and carry out respective forward and inverse kinematics analysis to produce the relative transforms. The model also allows specifying visuals for robot parts using notations as a cylinder, sphere or box, and colour values with material tags. More complex information, such as collision information, inertia, joint and velocity limits, can also be given here. Specific utilities, e.g., *urdf_parser* can check for the validity of the URDF file.

C. MoveIt

MoveIt [22, 23] is a set of software packages with specific capabilities for mobile manipulation, such as kinematics, motion planning and control, 3D perception and navigation. *MoveIt* is integrated with standard ROS. It uses plugins for most of its functionality; motion planning (Open Motion Planning Library (OMPL) [24]), collision detection (default: Fast Collision Library (FCL) [25], and kinematics (default: OROCOS Kinematics and Dynamics Library (KDL) for forward and inverse kinematics for generic arms.

1) MoveIt setup assistant (SA): has the objective to reduce the entry barrier for developing robot software. It can automatically set up the task pipeline for producing an initial configuration quickly. Robot model URDF (section IV-B) is a prerequisite for MoveIt setup assistant. The configurations set by the assistant include self-collision matrix, planning group definitions, robot poses, end effector semantics, virtual joints list, and passive joints list. The first step of the SA is the generation of a self-collision matrix for the robot that can be used in future planning to speed up collision checking. This collision matrix encodes pairs of links on a robot that can safely be discarded from collision checking due to the kinematic infeasibility of there actually being a collision. During the step-by-step process user can provide information on different motion planning aspects. Virtual joints attach the robot to the world. Planning groups semantically describe parts of the manipulator, i.e., what constitutes a gripper or which joints and links comprise an arm. SA allows adding certain fixed poses of the manipulator. We can define query positions such as the initial and the goal configurations of the manipulator, and end-effectors can be labelled. In the last step, SA generates several configurations and launch files that can be used inside a ROS package.

V. Proposed solution architecture

In this section, we present the general architecture for developing a dynamic motion planning approach using the tools described in the last section. For a simulation of the robotic manipulator, we identify three key functional modules in the architecture, perception, modelling and planning (Figure 5).

The central component *Motion planning framework* is developed as a ROS node and integrates different modules into the final motion-planning solution. Perception of the world is available through the voxel-based grid (discussed in section III-A). This grid is formed using data from vision sensor ZED 2K stereo vision camera [8].

The model of the manipulator, a prerquisite for the MoveIt setup assistant is given as a URDF file; central node looks for the *robot_description* parameter on the ROS param server to get the URDF. We used the model available in the UR5 repositories [26] that we have installed and built. An initial set of configurations generated by MoveIt setup assistant defining query poses, self-collision matrix, planning groups, is also

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

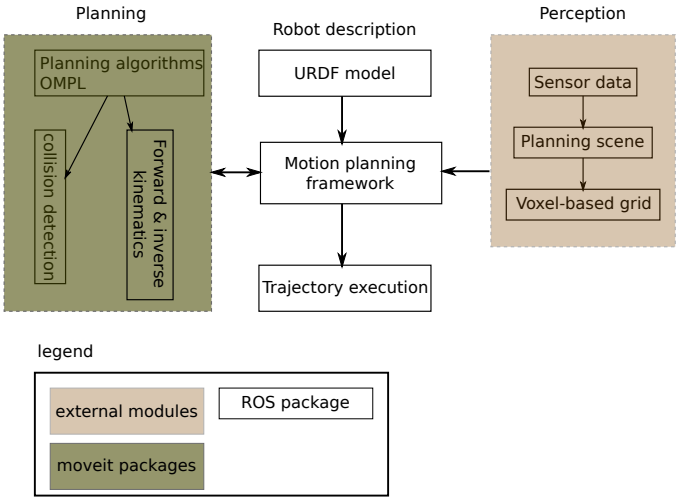


Figure 5. Robot manipulator motion planning - simulation architecture and module relationship.

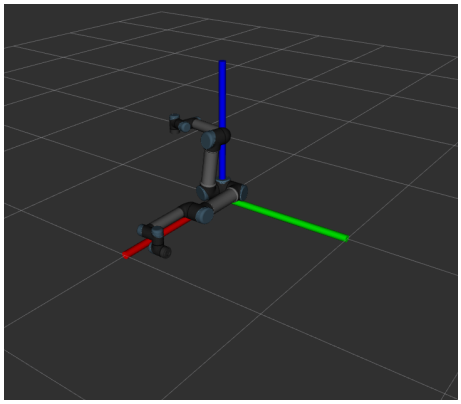
available on the param server, that Motion planning framework can look.

Concerning motion planning, MoveIt has a plugin based on the OMPL library that primarily implements randomized motion planners, such as PRM (Probabilistic RoadMaps), RRT (Rapidly Exploring Random Trees), RRTConnect etc. The central node offers an interface to the motion planner through ROS actions or services (a communication paradigm in ROS). Collision checking in MoveIt happens inside the planning scene. In particular, we can specify either pose goals or joint-space goals. Pose goals define a position of the end-effector in 3-d cartesian coordinates, whereas a joint-space goal identifies a distinct final configuration for the joints (given by individual joint angles). For both cases, we can plan the movement of the robotic arm to the desired goal. These tests have been done within graphical simulator Rviz as well as on the UR5 robotic arm. The framework allows us to add collision objects (obstacles) to the workspace. Collision objects are geometric primitives such as a box or a cylinder, and can be easily specified through their key dimensions and 3d position. In this case, the planned trajectory will avoid the obstacle or fail to find a solution when the target configuration cannot be achieved in the presence of the obstacle.

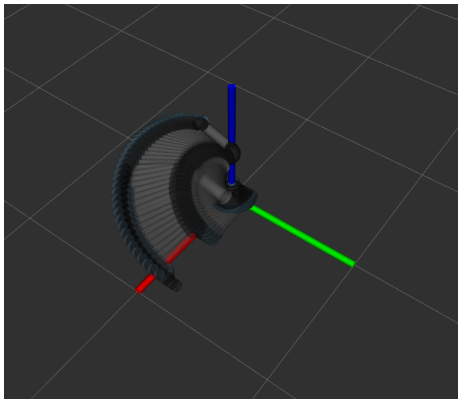
A. *Preliminary test*

We perform a simple test with regards to chosen tools. In this experiment, we use PRMstar [27] algorithm. We consider a joint-space goal given by the following configuration vector $\{-1.83, -1.732, 1.8, -1.634, -1.57, 2.88\}$ where joint angles are listed in radians. We test the motion of UR5 arm to this goal configuration in the absence of a collision object. Figure 6 shows the results for this test. For reference, see Figure 6 (a) that considers the case with no obstacle object and start and goal positions of the robotic arm. Start position is the manipulator lying in the x-y plane. In Figure 6 (b) we see the path trail that UR5 follows to reach the goal position. For the cases (a) and (b), the algorithm creates 785 roadmap states and solution is found in 5.009850s.

It is important to note that monitoring solution that outputs



(a)



(b)

Figure 6. Motion planning of UR5 manipulator.

the voxel-grid is a complementary component of intelligent motion planning. Its main objective is to provide online information on the workspace occupancy by different objects in the scene. Path planning component availabe via MoveIt is a standalone component. The reported experiment, validates the later.

VI. Conclusion

Robot path planning is a classic problem and is complicated in human-robot collaborative environments that present dynamic scenarios for motion planning. In this paper, we have reviewed some motion planning approaches that address such environments, i.e., where complete information is not available at planning time. We combine a voxel-based grid with a roadmap-based approach, for its simplicity and ease of development. Random sampling might result in roadmaps with disconnected components, and thus, it fails to find a path when start and goal configurations lie in disconnected components. For such cases, an online planner such as RRT would be helpful as it incrementally builds the complete path to the target. The software development for robots requires a breadth of knowledge and steep learning curve. For this reason, tools such as ROS and MoveIt with available functional packages can improve development time. We briefly described some essential concepts from these tools. And, presented a simulation architecture based on these tools.

The temporal validity of free spaces is not apparent in the

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

reviewed approaches. The cell marked as occupied may get free before the execution, but the algorithm has rendered it as occupied. It can be marked free only in the following planning episode. In this case, the planner might find an alternative path without the high cost. The contrary is more costly when space gets occupied between planning episodes, and the algorithm has not yet marked it occupied; leading to a failed query. Our future work will investigate this scenario. To efficiently handle dynamic scenarios, we can input fresh voxel grids to the planning program, with a certain frequency. The refresh rate of the voxel grid depends on the response time of previous planning query. Currently, we are looking into solutions to address these issues.

ACKNOWLEDGMENTS

INDTECH 4.0 – New technologies for intelligent manufacturing. Support on behalf of IS for Technological Research and Development (SI à Investigação e Desenvolvimento Tecnológico). POCI-01-0247-FEDER-026653

REFERENCES

[1] R. I. Association, "How Robots Are Taking on the Dirty, Dangerous, and Dull Job," https://www.robotics.org/blog-article.cfm/How-Robots-Are-Taking-on-the-Dirty-Dangerous-and-Dull-Jobs/209, accessed: 2020-05-19.

[2] P. Anderson-Sprecher, "Intelligent Monitoring of Assembly Operations," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-12-03, June 2011.

[3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," IEEE Transactions on Robotics and Automation, vol. 12, no. 4, August 1996, pp. 566–580.

[4] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Department of Computer Science, Iowa State University, Tech. Rep. TR 98-11, October 1998.

[5] D. Katz, J. Kenney, and O. Brock, "How Can Robots Succeed in Unstructured Environments," in Workshop on Robot Manipulation: Intelligence in Human Environments at 4th Robotics: Science and Systems Conference (RSS 2008). Citeseer, June 2008.

[6] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong Planning A*," Artificial Intelligence, vol. 155, no. 1-2, 2004, pp. 93–146.

[7] D. Thornton Coleman IV, "Methods for Improving Motion Planning Using Experience," Ph.D. dissertation, University of Colorado, 2017.

[8] L. Antão, J. Reis, and G. Gonçalves, "Voxel-based Space Monitoring in Human-Robot Collaboration Environments," in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2019, pp. 552–559.

[9] R. Nogueira, J. Reis, R. Pinto, and G. Gonçalves, "Self-adaptive cobots in cyber-physical production systems," in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2019, pp. 521–528.

[10] U. Robots, "UR5 collaborative robotic arm," https://www.universal-robots.com/products/ur5-robot/, 2015, accessed: 2019-12-20.

[11] T. Lozano-Pérez, "Spatial Planning: A Configuration Space Approach," in Autonomous Robot Vehicles. Springer-Verlag, 1990, pp. 259–271.

[12] J. P. Van Den Berg and M. H. Overmars, "Roadmap-based Motion Planning in Dynamic Environments," IEEE Transactions on Robotics, vol. 21, no. 5, 2005, pp. 885–897.

[13] L. Jaillet and T. Siméon, "A PRM-based Motion Planner for Dynamically Changing Environments," in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), vol. 2. IEEE, 2004, pp. 1606–1611.

[14] J. J. Kuffner Jr and S. M. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning," in Proceedings of the 17th IEEE International Conference on Robotics and Automation (ICRA 2000), vol. 2. IEEE, April 2000, pp. 995–1001.

[15] A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments," in Intelligent Unmanned Ground Vehicles. Springer, 1997, pp. 203–220.

[16] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, July 1968, pp. 100–107.

[17] S. Koenig and M. Likhachev, "Fast Replanning for Navigation in Unknown Terrain," IEEE Transactions on Robotics, vol. 21, no. 3, 2005, pp. 354–363.

[18] A. Stentz et al., "The Focussed Dˆ* Algorithm for Real-Time Replanning," in IJCAI, vol. 95, 1995, pp. 1652–1659.

[19] Z. Iqbal, J. Reis, and G. Gonçalves, "Path planning for an industrial robotic arm," in The Eighth International Conference on Intelligent Systems and Applications (INTELLI 2019). IARIA, 2019, pp. 30–36.

[20] M. Kallman and M. Mataric, "Motion planning using dynamic roadmaps," in IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, vol. 5. IEEE, 2004, pp. 4399–4404.

[21] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in ICRA workshop on open source software, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[22] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," IEEE Robotics & Automation Magazine, vol. 19, no. 1, 2012, pp. 18–19.

[23] S. Chitta and I. Sucan, "Moveit," https://moveit.ros.org/, 2012, accessed: 2019-10-31.

[24] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," IEEE Robotics & Automation Magazine, vol. 19, no. 4, 2012, pp. 72–82.

[25] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 3859–3866.

[26] I. GitHub, "Universal Robot," https://github.com/ros-industrial/universal_robot, 2019.

[27] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," The international journal of robotics research, vol. 30, no. 7, 2011, pp. 846–894.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

Preliminary Evaluation of Speech-to-Text Query Application for Parts Database

Yuichiro Aoki
Research and Development Group, Center for Technology Innovation - Digital Technology
Hitachi, Ltd.
Tokyo, Japan
email: yuichiro.aoki.jk@hitachi.com

Tadashi Takeuchi
Research and Development Group, Data Science Laboratory
Hitachi, Ltd.
Tokyo, Japan
email: tadashi.takeuchi.dt@hitachi.com

***Abstract*— To ensure the safety and security of workplaces and homes, parts replacement of mechanical/electric/electronic devices is inevitable, because they will gradually break down. Users have to search for the appropriate part numbers from a parts database to order and replace the parts. In many cases, users make phone calls to the support center of the vendors instead of searching by themselves, which can be inconvenient for the support center staff. In this paper, to reduce the number of phone calls to support centers, we propose a prototype of an automatic part number answering system from natural language. This system consists of open-source speech-to-text conversion and Structured Query Language (SQL) generation from the natural language. Preliminary evaluation results show that 83% of the voice questions returned the correct part numbers. In addition, the search with our system was executed an average of 3.86 times faster than with the conventional manual keyword search.**

Keywords-speech-to-text; SQL generation; natural language processing.

I. INTRODUCTION

To ensure the safety and security of workplaces and homes, parts replacement of mechanical, electric, and/or electronic devices in factories, construction sites, and homes is inevitable, as such devices will gradually deteriorate and break down. Before a malfunction occurs, device users have to search for the part numbers on a parts database to order and replace parts. However, generally speaking, users do not know the correct part names, thus they cannot use the conventional manual keyword search. Instead, they make phone calls to the support center of the device vendors, which inconveniences the support center staff, who invariably have little time to improve their productivity or the quality of their support. To reduce the number of phone calls or even completely eliminate them, one potential solution is an automatic answering system for the part number.

In this paper, we propose an automatic part number answering system using speech recognition (speech-to-text) and SQL generation from natural language and make a preliminary evaluation of its functionalities and performance. We use an open-source software for speech-to-text and develop a SQL generation algorithm. More specifically, we assume that "ask back" processing of the system is new and more pragmatic than previous studies in real industry fields.

In Section II of this paper, we review related work. In Section III, we describe the architecture of our system and Graphical User Interface (GUI). The preliminary evaluation results are reported in Section IV. In Section V, we discuss the functionalities and the performance, followed by a conclusion and future study in Section VI.

II. RELATED WORK

In this section, we review speech-to-text technology and SQL generation technology from natural language. First, we focus on speech-to-text. Table I shows the comparison of speech-to-text technologies offered by various providers. Amazon Transcribe on Amazon Web Services (AWS) handles 14 languages, but does not handle Japanese and has no sound model customization [1]. Cloud Speech-to-Text on Google Cloud Platform (GCP) can recognize 120 languages and dialects, including Japanese, and can customize the sound model. However, it does not have a user dictionary [2]. Watson Speech to Text on IBM Cloud handles Japanese, has a user dictionary, and can customize the sound model, but is proprietary [3]. Speech-to-Text on Microsoft Azure handles Japanese and can perform sound model customization. However, it does not have a user dictionary [4].

TABLE I. COMPARISON OF SPEECH-TO-TEXT TECHNOLOGIES.

Name	*Japan-ese*	*User Dictionary*	*Sound Model Custom-ization*	*Open Source Software*
Amazon Transcribe		✓		
Google Cloud Speech-to-Text	✓		✓	
IBM Watson Speech to Text	✓	✓	✓	
Microsoft Speech-to-Text	✓		✓	
Julius (This Study)	✓	✓	✓	✓

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

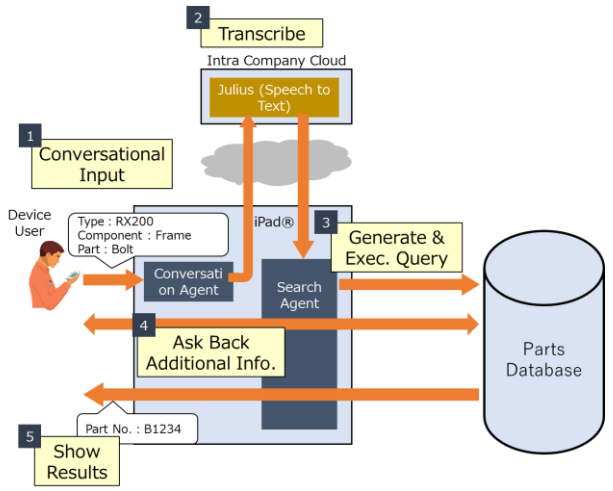


Figure 1. Overview of our system.

Julius is an open-source, speech recognition engine with a large vocabulary. It supports statistical N-gram model and rule-based grammars as a language model and Hidden Markov Model as a sound model [17]. Julius translates Japanese speech into Japanese text, has a user dictionary, can customize the sound model, and is open-sourced [5]—in fact, no speech-to-text technology other than Julius is open-sourced. In addition, Julius can be deployed on the intra-company cloud and has less risk of information leak. For these reasons, we opt to utilize Julius for our system.

Second, we examine SQL generation technologies from natural language [6][7]. Precise [8] constructs systematic natural language queries that are easy to translate into SQL queries. Rao et al. [9] convert inquiries that are written in natural language and whose words are in the user-defined dictionary into SQL queries. They cannot make queries interactively. Safari and Patrick [10] transform general natural language queries into restricted natural language queries first and then convert them into SQL queries. NaLIR [11] makes a syntax analysis of the natural language query and shows its result to the user. Next, the user manipulates the result manually and modifies the syntax tree to one that is easy to convert into an SQL query. Sukthankar et al. [12] generate complex SQL queries that include WHERE clauses. Seq2SQL [13] converts natural language queries into SQL queries by using reinforcement learning. However, the accuracy of the generated SQL queries is about 50–60%. All the systems above lack interactive functionalities on mobile devices or have insufficient accuracy of the generated SQL queries. In contrast, our system interactively converts the natural language questions into SQL queries on mobile devices such as an iPad® and does not show the details of the analysis or the transformation. Moreover, it can generate complex SQL queries using WHERE clauses. Preliminary evaluation results show that 83% of the generated SQL queries are correct.

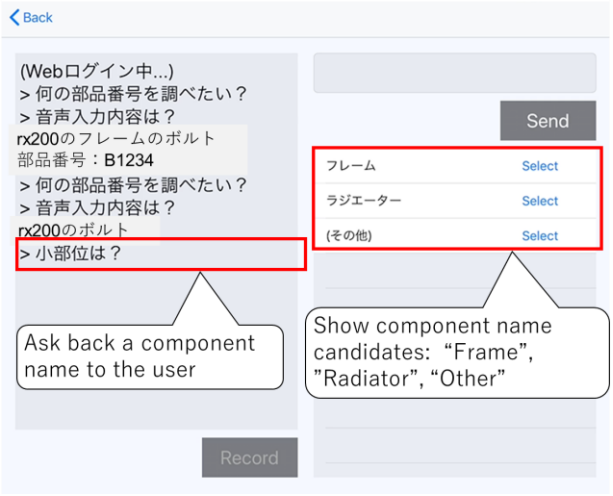
Real world applications, such as Apple Siri [14], Google Assistant [15], and Amazon Alexa [16], resemble our system. However, they cannot use intra-company proprietary databases. Our system can deal with such databases.

III. Overview of Our System

In this section, we provide an overview of our system. The device user makes a conversational input consisting of the part information, such as the machine type name (RX200), the component name (Frame), and the part name (Bolt), to the system. The system analyzes the information and returns the corresponding part number (B1234).

This process is shown in Figure 1. First, in step 1, the user talks to the conversation agent on the iPad®, saying (for example) “RX200, Frame, and Bolt”. In step 2, the conversational agent accepts the utterance, formats it as Moving Picture Experts Group (MPEG)-1 Audio Layer-3 (MP3) and sends it to the speech-to-text engine (Julius) on the intra-company cloud.

(a) Complete question



(b) Incomplete question

Figure 2. GUI images on an iPad®.

Julius transcribes the utterance to text and sends it back to the search agent on the iPad®. In step 3, the search agent generates the SQL query. Words from the text are embedded in WHERE clause. Finally, the search agent executes the query on the parts database. The example SQL is as follows:

SELECT part_number FROM parts_database WHERE type='RX200' AND component='Frame' AND part='Bolt'.

The parts database returns the part numbers to the search agent. If the search agent does not get a unique part number from the SQL query, in step 4, it generates an additional SQL query from the answer, as follows:

SELECT component FROM parts_database WHERE type='RX200' AND part='Bolt'.

The answer of the additional SQL query is displayed as selection buttons on the iPad®. The device user taps the appropriate button for the component name, that is, the search agent asks back the device user what the component name is. Finally, the search agent shows the part number, which in this case is "B1234", on the iPad® in step 5. Manual correction of the speech recognition result can be performed if the user finds incorrect results.

Next, we present the GUI of our system. Figure 2(a) shows an example where a complete question is asked, which includes a type name, a component name, and a part name. The user verbally inputs the complete question and the system transcribes it (red box in the figure). Then, the user selects the appropriate text and the system replies with the part number.

The case of an incomplete question is shown in Figure 2(b), where some of the required information is lacking in the question—in this example, a component name. The system analyzes this incomplete question and shows candidates for necessary information that is missing in the voice input ("ask back" processing, red boxes in the figure). The user taps the appropriate candidate button and the system replies with the part number.

IV. PRELIMINARY EVALUATION

In this section, we report our preliminary evaluations of the functionalities and performance of our system. All evaluations are done in Japanese. We use the Apple® iPad® (6th Generation) and the intra-company cloud for the evaluation.

A. *Functional Evaluation*

We manually make 100 artificial oral questions to test the coverage of the system. Each question is composed of a machine type name, a component name, and a part name. That is, all questions are complete question cases. All combinations of the names in the questions are different from each other. Examples of questions are "RX200, Frame, and Bolt" or "RX78, Generator, and Chain". One of the authors orally inputs these questions in the office.

Table II lists the functional evaluation results. As shown, the system delivered the correct part numbers for 83 of the questions. However, 12 questions exhibited inappropriate speech recognition, and there were five error results.

TABLE II. FUNCTIONAL EVALUATION RESULTS.

Results	Number of Questions
Correct part number	83
Inappropriate speech recognition	12
Other errors	5
Total	100

TABLE III. BREAKDOWN OF 83 SUCCESSFUL RESULTS

Voice Input	Additional Info Needed	Manual Correction	Number of Questions
✓			0
✓	✓		2
✓		✓	40
✓	✓	✓	41

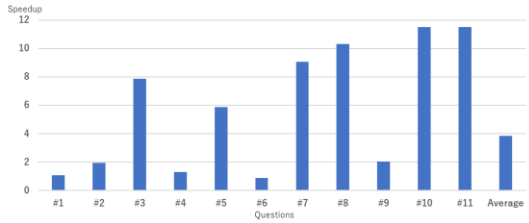


Figure 3. Speedup over the conventional search.

Table III shows the breakdown of the successful 83 results. In the case of voice input only, there were no results. When we used the voice input and the additional information input required by the system ("ask back" processing), there were two results. Using the voice input and the manual correction of the small speech recognition mistake, there were 40 results. When all three items were used, there were 41 results.

B. *Performance Evaluation*

Next, we evaluate the performance of the system. We prepare 11 additional oral questions from the device users. These 11 questions are different from the 100 questions in the previous section in that they were actually used by the device users in real engineering fields. Some questions are incomplete, such as "RX78 and Generator". One of the authors orally inputs these questions in the office, too. Figure 3 shows the speedup of the search time over the conventional search. The conventional search specifies the correct part name manually using a keyboard and then searches for it. If the search does not get the part number within 300 seconds, we assume the search time to be 300 seconds. When we tallied the results, we found that our system had a speedup larger than 1 in all questions except #6 (0.88 times). The average speedup was 3.86.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

V. DISCUSSION

In this section we will discuss the functional and performance evaluation results.

A. Functional Evaluation Results

First, our system could not transcribe 12 questions properly due to a deficiency of the synonym dictionary for terms such as “urea”. We need to expand the synonym dictionary to include domain-specific terms. The other five errors are currently under investigation.

Second, 81 of the 83 questions that led to the correct part number required manual correction of the transcribed text. This is because Julius failed to detect the pauses between words, as the background noise of the evaluation environment was not exactly small. A noise-resistant system is a future consideration. Parameter tuning of Julius may improve the situation.

Third, 43 questions required additional information to reach the correct part number. This is because the same part and component names appeared in a device type. In such cases, the system asks back the device user which information is needed. To automate this issue, we propose to display probable combinations of names and the user to select proper one.

B. Performance Evaluation Results

In the conventional search, the device user needed to know the correct type/component/part name before the search. In addition, some questions lacked information (e.g., component name). As a result, the user could not input the correct search keywords and six of the 11 questions did not lead to correct part numbers within 300 seconds. In contrast, our system could obtain all the correct part numbers within 300 seconds. In several cases, questions about the missing information were automatically asked back by the system. In question #6, the tester coincidentally knew the correct part name, so the conventional search was executed faster than our system.

Additionally, our system acquired the part numbers an average of 3.86 times faster than the conventional search. We assume that the automatic “ask back” processing in our system helped reduce the search time.

VI. CONCLUSION

To reduce the number of phone calls to support centers, we have developed an automatic part number answering system consisting of an open-source speech-to-text software and SQL generation from natural language (Japanese).

The functional evaluation results demonstrated that 83% of the questions returned the correct part numbers. Moreover, the performance evaluation results showed that our system executed the search an average of 3.86 times faster than the conventional search.

In future work, we will examine how to improve the accuracy of voice recognition under noisy environments and decrease latencies of the system for speedup. In addition, we are planning to evaluate the performance and robustness of the SQL generation algorithms.

ACKNOWLEDGMENT

The authors thank Dr. Tsuyoshi Tanaka and Dr. Hiroaki Shikano for supporting this research.

REFERENCES

[1] Amazon Web Services, “Amazon Transcribe” [Online]. Available from: https://aws.amazon.com/transcribe/ 2020.03.26

[2] Google Cloud, “Cloud Speech-to-Text” [Online]. Available from: https://cloud.google.com/speech-to-text 2020.03.26

[3] IBM, “Watson Speech to Text” [Online]. Available from: https://www.ibm.com/cloud/watson-speech-to-text 2020.03.26

[4] Microsoft, “Speech-to-Text” [Online]. Available from: https://azure.microsoft.com/en-au/services/cognitive-services/speech-to-text/ 2020.03.26

[5] Julius, “Open-Source Large Vocabulary Continuous Speech Recognition Engine” [Online]. Available from: https://github.com/julius-speech/julius 2020.03.26

[6] B. Sujatha, S. V. Raju, and H. Shaziya, “A survey of natural language interface to database management system,” International Journal of Science and Advanced Technology, vol. 2, no. 6, pp. 56–61, 2012.

[7] H. S. Dar, M. I. Lali, M. U. Din, K. M. Malik, and S. A. C. Bukhari, “Frameworks for Querying Databases Using Natural Language: A Literature Review,” arXiv:1909.01822, 2019.

[8] A.-M. Popescu, O. Etzioni, and H. Kautz, “Towards a theory of natural language interfaces to databases,” in Proc. of the 8th International Conference on Intelligent User Interfaces, pp. 149–157, 2003.

[9] G. Rao, C. Agarwal, S. Chaudhry, N. Kulkarni, and D. S. H. Patil, “Natural language query processing using semantic grammar,” International Journal of Computer Science and Engineering, vol. 2, no. 2, pp. 219–223, 2010.

[10] L. Safari and J. D. Patrick, “Restricted natural language based querying of clinical databases,” Journal of Biomedical Informatics, vol. 52, pp. 338–353, 2014.

[11] F. Li and H. V. Jagadish, “Constructing an interactive natural language interface for relational databases,” in Proc. of the VLDB Endowment, vol. 8, no. 1, pp. 73–84, 2014.

[12] N. Sukthankar, S. Maharnawar, P. Deshmukh, Y. Haribhakta, and V. Kamble, “nQuery: A Natural Language Statement to SQL Query Generator,” in Proc. of ACL2017, Student Research Workshop, pp. 17–23, 2017.

[13] V. Zhong, C. Xiong, and R. Socher, “Seq2SQL: Generating Structured Querues from Natural Language using Reinforcement Learning,” arXiv:1709.00103, 2017.

[14] Apple, “Siri” [Online]. Available from: https://www.apple.com/siri/ 2020.08.14

[15] Google, “Google Assistant” [Online]. Available from: https://assistant.google.com/ 2020.08.14

[16] Amazon, “Alexa” [Online]. Available from: https://developer.amazon.com/en-US/alexa 2020.08.14

[17] A. Lee and T. Kawahara, “Recent Development of Open-Source Speech Recognition Engine Julius,” Asia-Pacific Signal Information Processing Association Annual Summit and Conference, 2009.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

Clustering Techniques for On-Demand Transport Data: A Case Study

Carlos Afonso[1]
[1]Department of Informatics and Systems Engineering
ISEC, Polytechnic Institute of Coimbra
Coimbra, Portugal
e-mail: a21240004@isec.pt

Ana Alves[1,2]
Department of Informatics and Systems Engineering
[1]ISEC, Polytechnic Institute of Coimbra
[2]CISUC, University of Coimbra
e-mail: ana@dei.uc.pt

Abstract— **The on-demand transportation request requires a quick and efficient response to satisfy customers and also make the system a viable option. Clustering techniques were used to group transport requests, i.e., the starting points of vehicles that have been requested to optimize the service with the benefit of reducing the number of vehicles needed and, consequently reduce the amount of pollution produced. The objective is to compare the two main clustering techniques from two distinct categories: partitioned and density-based to evaluate which one is best suited for defining start zones. The quality of the generated clusters is defined by calculating the silhouette related to the generated clusters. Using previous references, the two clustering methods were compared based on the desired characteristics. The analysis demonstrates that DBSCAN is best suited for the problem and is then applied over a sample dataset. The manner in which the DBSCAN algorithm can generate random shapes, which fit well into the geographic distribution of points and how the number of necessary clusters do not need to be defined in advance makes it the ideal choice for defining starting zones.**

Keywords- On-Demand Transport; Transport Requests; Partition-based Clustering; Density-based Clustering; K-Means; DBSCAN

I. INTRODUCTION

Clustering techniques are essential since they allow grouping something, whether they are objects or people, according to their degree of similarity. In this way, clustering will play a key role in this study focused on smart mobility, and the objective is to optimize the collection points of people in a city or on a route to it. It involves aggregating transport requests, grouping people together and simultaneously ensuring route optimization [1]. This article will compare two of the various clustering techniques that exist, more specifically K-Means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN**)**, since they are part of the core of the most popular clustering techniques [2]. Clustering techniques are essential for problems such as people search. Throughout this paper, several key points of each of the mentioned techniques will be described, namely, a description, advantages and limitations, a comparison between the two and a case study that will use the best technique in comparison.

This paper is structured as follows: in section 2, several study scenarios are covered with the clustering techniques that will be compared; in section 3, where various techniques for encouraging the grouping of persons are addressed; in section 4 is made a brief description of Clustering and K-Means and DBSCAN are analyzed and compared. In addition, the silhouette coefficient is also addressed; in section 5, a comparison is made between the two clustering techniques (K-Means and DBSCAN); in section 6, the case study carried out with the application of the best clustering technique resulting from the comparison made in the previous section is described. Finally, in section 7, the conclusions and future work are presented, followed by the thanks and their references.

II. STATE OF THE ART

The importance of clustering applied to mobility has been demonstrated [3] exploring travel patterns and target transit passengers only using smart card data. These smart cards have individual transactions, that is, per user and each working day was used to build travel itineraries. In this research work, DBSCAN and K-Means were used as clustering algorithms. For this, it was necessary to resort to DBSCAN, which was used to group the last landing stops made by users. In addition, the same algorithm also groups the starting stops and the time a passenger normally boards a particular vehicle. Finally, through K-Means, users were classified and divided into several groups, such as “transit passengers”, “regular passengers”, “habitual time passengers” and “irregular passengers”.

Another research work used data-mining tools and presented several measures related to the variability of travel behavior by public transport users. These analyses were carried out using, once again, the smart card and were collected over a period of ten months allowing to understand the difference in terms of boarding per day and new frequent stops with the travel days on the public transport network. For this, the authors used data-mining techniques to build clusters of days that present similar time patterns of boarding on the public transport network in order to understand whether passengers have regular travel behaviors and whether the days differ significantly. The experiments have shown that the behavior of habitual transport users evolves over time, both in terms of frequent transport stops and in relation to boarding

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

hours, leading to the conclusion that the change in behaviors varies according to the various types of existing users [4].

Another study focuses on taxi transport in Taiwan [5], helping taxi drivers to find high-density locations. This was due to the fact that drivers generally do not know where the passengers are, making them spend a lot of time driving unoccupied vehicles, thus revealing a huge loss to the business. Therefore, it was essential to look for a solution for taxi drivers to find potential customers and the locations of future customers. Based on a given reference position, weather condition, current weather, order history, hotspots have been calculated that can be provided and recommended derived from the use of data-mining techniques. In this case, DBSCAN served to group the coordinates of customer locations according to the spatial distance, leading to each identified cluster, the corresponding roads properly associated. With the result of this analysis, taxi drivers can help their strategies and decided where to go in order to pick up more passengers.

Another research paper published in GeoInfo [6], came up with the development of an application based on the DBSCAN clustering algorithm with the aim of reducing the daily loss of time in moving a large number of people to a common place. Clusters are created based on various attributes, such as the departure time of each person in his home, the final destination and their departure and arrival locations. The people who make up a given cluster are transported in an allocated vehicle according to the size of the same. A case study was then conducted with a certain group of people moving to campus II of the Federal Center for Technological Education of Minas Gerais using a traffic simulator to measure the individual and global time people need to move. Using the concepts of clustering on the latitude, longitude and time of movement of individuals, data were collected and processed in order to group people with similar movement time and location of origin and destination, making it possible to find routes that can transport these groups of people effectively. In order to find a solution that would improve people's daily movement time, the problem was divided into two parts. First, people were brought together in groups, according to the DBSCAN algorithm. For each cluster, a centroid was calculated and defined as a starting point for people in that particular cluster. So, in the first part of the route, each person walks to that central point, spending a certain initial time of the walk, having established that the time of the walk had to be less than twenty minutes. The distance of each individual to the center of the cluster was calculated using the haversine function and the estimated movement time to the centroid was defined as the average walking time of 4.82 kilometers / hour for each person. Subsequently, in the second part of the problem, after all people walked to the centroid of their respective cluster, the route would be established from one or more vehicles. The type of vehicle was chosen taking into account the number of people allocated to each cluster, with up to 5 people being displaced by car, between 6 to 15 people would be in a van, between 16 to 40 would be by bus or more than 40, people would be distributed among several vehicles, depending on their total number. After choosing the most suitable vehicle, the rest of the route is carried out without any kind of stop to the final destination and the movement time of each vehicle to the final destination was calculated using data from Google Maps, thus obtaining the total time that it would take [7].

Another study proposed an on-demand transportation system using clustering, in the book “Data-Driven Solutions to Transportation Problems” [8] had as main objective to help taxi drivers to collect and transport passengers from their departure location to their intended destination location and simultaneously walk the road in order to find the next customer. Data was collected from 1100 drivers in Harbin, a city located in China. This data contains various information, such as: **time,** which indicates the date and time when the records were recorded; **latitude and longitude** that specify the location of the taxi vehicle; -**speed**, which indicates the average speed of the vehicle; **orientation,** which represents the direction in degrees; **travel status,** which is a Boolean value that indicates whether the taxi is occupied by passengers. This study had as main objective to help taxi drivers to collect and transport passengers, from their departure location to their intended destination location and simultaneously walk the road in order to find the next customer. It was then relevant to understand which area of the city can attract a greater number of people and which is the area of the city. Thus, they decided to use DBSCAN as a clustering technique in order to group passengers' embarkation and disembarkation places. Their use has brought several benefits, such as the fact that they are able to classify locations in a cluster with high density, are able to find specific locations on the road network for each cluster and are able to deal well with noise points. Although DBSCAN was already used to group other types of things apart from transport requests, it was found to be the most used for the on-demand transport domain.

III. GROUPING OF PEOPLE

Before moving on to the description of clustering and some of its techniques, it is important to run a little context to understand why this unsupervised machine learning technique will be used. One of the causes of great mobility difficulties, especially in the city, is the volume of private vehicles circulating daily. A high percentage of them travel with only one passenger, which represents reduced efficiency. This inefficient use has a major impact on the flow of traffic routes, difficulties related to parking problems and increased pollution.

Grouping people into vehicles that will make the same trip or similar routes represents one of the ways to reduce the impact mentioned. If only one person is in the vehicle, the cost will be much higher, leading to greater congestion on the roads, whereas there will be a greater number of vehicles on them and a greater impact on environmental pollution in society. In addition, there is a financial impact resulting from the reduction in the cost of travel per element.

There are financial incentives and methods of road organizations to group as many people in a given vehicle as ***High-occupancy vehicle lanes, High-Occupancy toll lanes*** and ***Slugging lines.*** The use of a car has a relatively high cost,

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

maintenance, consumption, wear at the pneumatic level, among others.

The ***High-occupancy Vehicle lane (HoV)*** method [9] encourages drivers to bring together as many people as possible, offering specific transit routes. On roads with heavy vehicle traffic with a single occupant, this special route facilitates circulation, saving time for those sharing trips. In Portugal, there are traffic corridors for public transport (*buses* and *taxis)*, but in this case, the HoV allow its use by private vehicles.

The ***High-occupancy toll lane*** method [10] encourages users to gather as many people in a given vehicle, reducing the toll price. A smaller number of people in a given vehicle leads to an increase in toll prices.

Another method is ***slugging*** [11], which is an organized system in which people travel to a city to pick up other passengers, who may even be strangers, and share the cost of the trip among them.

IV. CLUSTERING

Clustering or **Data Grouping Analysis** is the set of data-mining techniques that aim to create automatic data groupings according to their degree of similarity. The similarity criterion is part of the problem definition and is dependent on the algorithm that will be used. Each collection resulting from the process is given the name of the group or grouping (cluster) [12].

This data analysis technique is increasingly being used to classify elements into groups, in order to understand which elements within a dataset are similar (are in a cluster*)* and which elements of the set are distinct from each other (they are in distinct clusters*)* [13].

The data grouping algorithms that will be described below are the most popular within the categories of Partitioned (K-Means) and Density-Based (DBSCAN). According to the state of the art, K-Means and DBSCAN are the two techniques that are most used to group people.

A. K-Means

The K-Means algorithm, also known as Hard *C-means* [14], organizes the elements of a dataset in a given number of *clusters* defined by the user. This data should be organized in the different clusters according to the similarity between them. The K-means algorithm organizes the data in k clusters and determines the organization of the clusters through an iterative process. The iterative aspect of the algorithm exists to look for the best result with each iteration with some mechanisms to avoid ending in local optimums.

The operation of the K-means algorithm can be divided into four steps. In the first step, the centers of each cluster are initialized. Typically, these centers are initialized by choosing *k* points randomly in the dataset, with k being the number of clusters defined at the beginning by the user.

After cluster centers have been initialized, each of the dataset elements is assigned to each of these clusters. This assignment is performed based on the shortest distance between an element and the centers of the defined clusters, that is, the cluster to which the element is closest.

In the third step, the value of a cost function is calculated using the equation (1).

$$WCSS_1 := \sum_{c_i \in C} \sum_{j=1 \ldots d} \sum_{x,y \in c_i} (x_{ij} - y_{ij})^2 \qquad (1)$$

The cost **J** of the clustering iteration consists is the sum of the costs of each cluster **C** that consist of the sum of the distances of each element $\mathbf{X_k}$ to the center of the cluster $\mathbf{C_i}$ to which the element was assigned to. In this case, euclidean distance is used although it is possible to use other types of distances between vectors. After calculating the cost function, it is possible to see whether there was an improvement in the result in relation to the value calculated in the previous iteration. If the value has improved, the algorithm moves on to the next iteration. If this value has increased the algorithm ends its execution and returns the results obtained in the previous iteration.

In case the algorithm does not finish its execution in the third step, there is a fourth and final step consisting of the renewal of the centers of all clusters using equation (2). This equation calculates the new centers $\mathbf{C_i}$ of each cluster by averaging between the elements $\boldsymbol{X_k}$ of that cluster.

$$c_i = \frac{1}{|G_i|} \sum_{k, x_k \in G_i} x_k \qquad (2)$$

Once the new centers have been calculated, the algorithm returns to the second step and continues to iterate until the stop criterion is met. Many implementations of this algorithm also present a variant that consists of its repetition, choosing at the end the best result among all repetitions, that is, the one that has a lower cost function value. This variant was introduced due to two factors. The first because the quality of the *K-means* algorithm is dependent on the initialization of cluster centers, because if the startup was always performed with the same centers the result obtained for a dataset would always be the same. Another factor is the possibility of the algorithm reaching quite easily local optimums. Thus repeating the algorithm, a certain number of times and using different centers at the beginning of the algorithm, the probability of achieving better results increases [15].

B. DBSCAN

The **DBSCAN** [16] algorithm is based on the density at which clusters grow according to a given density [17] threshold, where the number of clusters is decided depending on the data that is provided. Since it is a density-based clustering algorithm, some points in the data may not belong to any cluster. Again, this is quite different from the algorithm that was mentioned earlier (K-means), where all points of the data are considered and necessarily belong to some cluster. This can be explained with the help of two parameters ***epsilon*** and ***min_points*** that are used in the algorithm. Two points are considered neighbors if the distance between them is below the ***epsilon*** threshold. The minimum number of neighbors that a given point must have to be classified as a main point is defined by ***min_points***. This type of algorithm is widely used at the level of population density and there are some aspects that are relevant and worth mentioning:

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

- DBSCAN is a flexible algorithm, since it is dynamic relative to the data;
- The parameters required to execute the algorithm can be obtained from the data themselves, using **adaptive DBSCAN;**
- It provides a more intuitive grouping as it is based on density that leaves out points that do not belong anywhere (outliers).
- DBSCAN has $O(n^2)$ Complexity [18] where *n* is the number of data points, and so it is much faster when compared to traditional clustering techniques such as K-means that has a complexity of O(IKn), where *n* is the number of data points, *k* is the number of clusters and *I* is the number of iterations [19].

C. Silhouette Coefficient

The silhouette coefficient is a metric used to calculate the quality of a clustering technique. This will be used to evaluate the quality of the DBSCAN algorithm that uses the average distance between points of the same cluster and the average distance between nearby clusters. The value of the silhouette varies between -1 and 1, where -1 is the worst possible score and 1 the best. The value of 1 represents high density clusters (internal points very close to each other) and clusters far between each other. A value of 0 suggests overlapping clusters and a value of -1 means that clusters are assigned in the wrong way [20].

V. COMPARISON OF THE TWO CLUSTERING TECHNIQUES

In [21], the author made a comparison of several clustering techniques where the metrics analyzed were as follows: **Feature A-** ability to identify clusters with random shapes; **Feature B-** ability to identify clusters in datasets with high data volume; **Feature C-** good performance in obtaining results, mainly also in datasets with considerable volume of data; **Feature D-** ability to deal with noise and achieve that its presence has no impact on the results obtained; **Feature E-** parameterization/initial configuration of the algorithm (the non-obligation to indicate the number of final clusters that will have to be generated and estimation of the initial parameters); **Feature F:** handle numeric values.

For the current case under study we do not know the number of *clusters* that should be generated, which makes Characteristic E relevant. The ability of the algorithm to be able to identify *clusters* with multiple random shapes is also a characteristic that is intended, given the fact that we do not know what type of form the *clusters* will have though we predict that the adopted forms will correlate with geographic planning. It will also be interesting to analyze whether the algorithm to choose can handle noise and whether it does not affect the results. The ability to handle numeric values is key, as the *clustering* of transport requests will be done and a grouping of a radius will have to be done. Another metric that will also be important to analyze is a good performance in achieving results and the ability to identify *clusters,* as it will use datasets with a large volume of data.

TABLE I. COMPARISON BETWEEN THE TWO CLUSTERING TECHNIQUES [21]

Algorithm	A	B	C	D	E
K-Means	x	x	x	x	✔
DBSCAN	✔	✔	✔	✔	✔

Through the Table I, we can perceive that *K-means,* which belongs to partitioned methods, is sensitive to noise which influences the final results and it cannot identify clusters with random shapes, being only able to identify circular *clusters.* One of the main limitations of the algorithm mentioned above is that it cannot find the best partitions for clusters with different sizes and with different densities, and for these reasons and those mentioned above, a decision was made to not use it. Therefore, the algorithm to be used for this paper will be DBSCAN, since it meets all the requirements necessary, namely the fact that it is able to deal with *clusters* with different sizes and shape and the fact that it is needed to group *n* transport requests around a geographic coordinate, which are based on numerical values.

To our knowledge, DBSCAN has not yet been used to group transport orders, although it has already been used to group other very similar types of data, as already mentioned in the state of the art.

VI. CASE OF STUDY

In view of the result of the comparison made earlier, it was concluded that the best technique to use would be DBSCAN. A dataset [22] consisting of a set of taxi trips to the city of Chicago was used. This dataset contains several columns that were essential to the study, such as the "*Pickup Centroid Latitude*" and the "*Pickup Centroid Longitude*" that refer to the location of a taxi's starting point, as well as the "*Dropoff Centroid Latitude*" and the "*Dropoff Centroid" Longitude*" that refer to the location of a destination point of a taxi.

Thus, to perform the case study, a Jupiter notebook was produced importing the ***scikit-learn*** library. First, for the case of starting points, the dataset was loaded containing 278000 points. A sample with first 1000 points of the full dataset was used for the remaining steps to simplify visualization. The data relating to the "Pickup Centroid Longitude" and "Pickup Centroid Latitude" were extracted, having drawn them on a map taking into account the "corners" of the map so that the dots fit inside.

DBSCAN execution requires *epsilon* and *min_points* to be defined. The value for *epsilon* is obtained by executing the *nearest neighbors algorithm* with the dataset. The value for *epsilon* is the *y* coordinate of the point on the generated graph that presents the greatest inflection. The choice for the *min_points* value was made by comparing the results of the DBSCAN algorithm and choosing the value that produce the most satisfying results. Based on the data from the pickups, the Nearest Neighbors algorithm was executed to obtain an epsilon value to use in the DBSCAN algorithm. Then, the DBSCAN algorithm and the removal of points with noise were executed. Thus, as optimal values for *epsilon,* the value of 0.0000001 was obtained and for *min_points* the value of 3

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

was obtained. DBSCAN's results were 56 clusters with a silhouette coefficient of 0.907. This coefficient corresponds to a metric for measuring DBSCAN performance and is calculated using the average intracluster mean distance [15], and a value close to 1 is desired that indicates a high density of the points within each cluster and far from the remaining clusters. The case of study is stored on GitHub in a public way and can be accessed through the following url: https://github.com/pedroafonsoo/clustering_case_study_industrial_seminars/blob/master/dbscan_case_study_dbscan.ipynb .

The image in Figure 1 shows the locations of the mapped dataset, the image in Figure 2 demonstrates the same but colored locations representing the clusters and the Figure 3 demonstrates the representation of the Figure 2 but in detail with the largest area populated by points. The images contain two axes: x- longitude; y- latitude. Each of the clusters created, on average, have 18 points, and the cluster that presented the highest number of points was the cluster with label 3 with 109 points. To limit the number of points per cluster, it should be necessary to apply in a second phase a second clustering technique with this capacity as it will be presented in the future work.

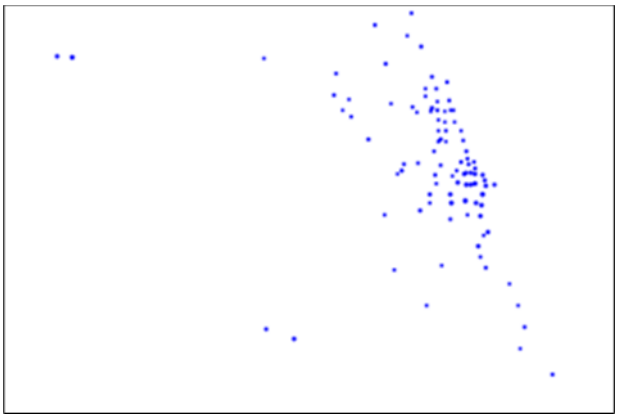


Figure 1. Spatial representation of Pickup points.

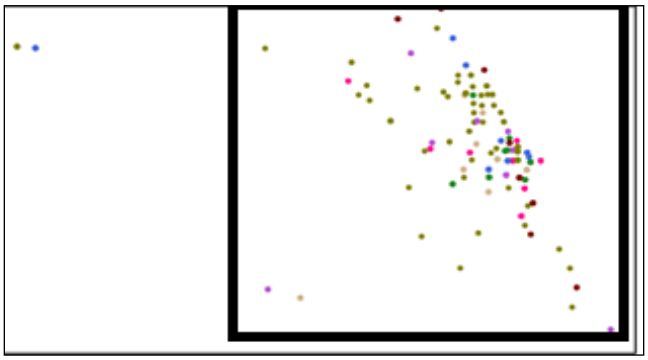


Figure 2. Spatial representation of Pickup points and clusters to which they belong (Each cluster is represented by a color which could be repeated across the 56 clusters generated)

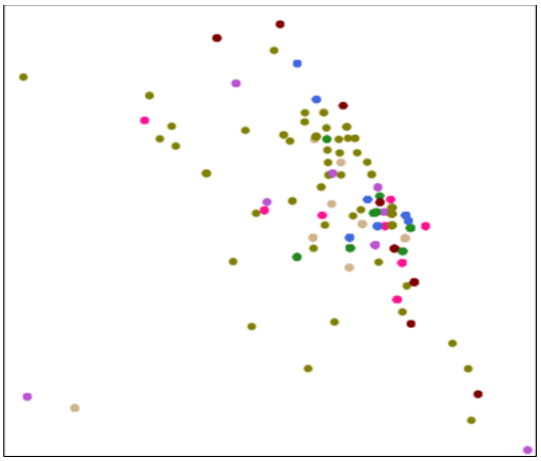


Figure 3. Representation of the Figure 2 but in detail with the largest area populated by points

VII. CONCLUSION AND FUTURE WORK

For the case of on demand transport study, the DBSCAN algorithm is the most appropriate compared to K-means, since it complies with all evaluated characteristics. By applying the algorithm with a real data subset, we obtained the set of associated clusters that define pickup points, with a strong silhouette value, indicating quality in the result. The generated clusters confirmed that there are a set of common pickup zones that can be explored in future steps of the study, which is publicly available. As a future work, it may be to combine the DBSCAN clustering technique with Constrained K-Means [23]. Constrained K-Means implementation modifies the cluster assignment step by formulating it as a Minimum Cost Flow (MCF) linear network optimisation problem. This is then solved using a cost-scaling push-relabel algorithm and uses Google's Operation Research tools's SimpleMinCostFlow which is a fast C++ implementation. With the application of a second phase of Constrained K-Means it will be possible to restrict the capacity of the minimum and maximum number of points for each cluster and at the same time guarantee the optimization of the distance between the points. In this way it will be possible to group a minimum and a maximum number of passengers, depending on the capacity of a given vehicle. In this way, Constrained K-means would be applied to each cluster resulting from DBSCAN.

ACKNOWLEDGMENT

I thank Ubiwhere, primarily to André Duarte for the opportunity, help and suggestions he gave for the preparation of this paper. I thank my advisor for ISEC, teacher Ana Alves, for her professionalism, both as a teacher – where in classes she excelled in pedagogical and didactic excellence- as a counsellor, and exemplary follow-up, which she has been performing during the development of the internship and finally to teacher João Cunha, not only for everything he taught in Industrial Seminar classes as well as the preparation he demanded with several presentations and debates throughout the semester.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

REFERENCES

[1] Cordeau, J. F., Laporte, G., Potvin, J. Y., & Savelsbergh, M. W. (2007). Transportation on demand. *Handbooks in operations research and management science*, *14*, 429-466.

[2] Seif, G. Towards Data Science, 5 February 2018. [Online]. Available: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68. [Accessed November 2019].

[3] Bhaskar, A., & Chung, E. (2014). Passenger segmentation using smart card data. *IEEE Transactions on intelligent transportation systems*, *16*(3), 1537-1548.

[4] Morency, C., Trépanier, M., & Agard, B. (2006, September). Analysing the variability of transit users behaviour with smart card data. In *2006 IEEE Intelligent Transportation Systems Conference* (pp. 44-49). IEEE.

[5] Chang, H. W., Tai, Y. C., Chen, H. W., Hsu, J. Y. J., & Kuo, C. P. (2008). iTaxi: Context-aware taxi demand hotspots prediction using ontology and data mining approaches. *Proc. of TAAI*.

[6] "GEOINFO," [Online]. Available: http://www.geoinfo.info/geoinfo2020/. [Accessed July 2020].

[7] Andrade, T. C., de Arruda Pereira, M., & Wanner, E. F. (2014). Development of an Application Using a Clustering Algorithm for Definition of Collective Transportation Routes and Times. In *GeoInfo* (pp. 13-24).

[8] Tang, J. (2019). Urban Travel Mobility Exploring With Large-Scale Trajectory Data. In *Data-Driven Solutions to Transportation Problems* (pp. 137-174). Elsevier.

[9] High Occupancy Vehicle (HOV) Lanes, [Online]. Available: http://www.its.leeds.ac.uk/projects/konsult/private/level2/instruments/instrument029/l2_029summ.htm. [Accessed April 2020]. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[10] High-Occupancy Toll Lanes (Partial Facility Princing), 2 March 2020. [Online]. Available: https://ops.fhwa.dot.gov/congestionpricing/strategies/involving_tolls/hot_lanes.htm. [Accessed April 2020].

[11] About Slugging, 31 January 2020. [Online]. Available: http://slug-lines.com/Slugging/About_slugging.asp. [Accessed April 2020].

[12] Clustering in machine learning, [Online]. Available: https://www.geeksforgeeks.org/clustering-in-machine-learning/. [Accessed April 2020].

[13] Oliveira, B. *O que é a análise de cluster?*, 2 October 2019. [Online]. Available: https://operdata.com.br/blog/analise-de-cluster/. [Accessed December 2019].

[14] Goktepe, A. B., Altun, S., & Sezer, A. (2005). Soil clustering by fuzzy c-means algorithm. *Advances in Engineering Software*, *36*(10), 691-698.

[15] Oliveira, J. *Classificação de Literatura Biomédica*, December 2014. [Online]. Available: http://files.isec.pt/DOCUMENTOS/SERVICOS/BIBLIO/Tees/Tese_Mest_Joao-Santos-Oliveira.pdf. [Accessed December 2019].

[16] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, No. 34, pp. 226-231).

[17] Castro, M. *Agrupamento-Clustering*, July 2003. [Online]. Available: http://www.dei.isep.ipp.pt/~paf/proj/Julho2003/Clustering.pdf. [Accessed December 2019].

[18] Agrawal, S. Machine Learning - DBSCAN, 29 May 2013. [Online]. Available: https://algorithmicthoughts.wordpress.com/2013/05/29/machine-learning-dbscan/. [Accessed December 2019].

[19] Clustering Algorithms: K-means, 2006. [Online]. Available: https://www.cs.princeton.edu/courses/archive/spring08/cos435/Class_notes/clustering2_toPost.pdf. [Accessed December 2019].

[20] Lutins, E. DBSCAN: What is it? When to Use it? How to use it., 6 September 2017. [Online]. Available: https://medium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818. [Accessed December 2019].

[21] Simões, J. *Repositório Comum- Análise de dados e Machine Learning na Mobilidade Urbana*, April 2019. [Online]. Available: https://comum.rcaap.pt/bitstream/10400.26/29858/1/Joao-Pedro-Fernandes-simoes.pdf. [Accessed December 2019].

[22] Taxi Trips-Chicago Data Portal, 11 December 2019. [Online]. Available: https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew. [Accessed December 2019].

[23] Bradley, P. S., Bennett, K. P., & Demiriz, A. (2000). Constrained k-means clustering. *Microsoft Research, Redmond*, *20*(0), 0.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

Surface Sensing of 3D Objects Using Vibrissa-like Intelligent Tactile Sensors

1st Lukas Merker
Dept. of Mech. Engineering
Technische Universitt Ilmenau
Ilmenau, Germany
e-mail: lukas.merker@tu-ilmenau.de

2nd Moritz Scharff
Dept. of Mech. Engineering
Technische Universitt Ilmenau
Ilmenau, Germany

3rd Klaus Zimmermann
Dept. of Mech. Engineering
Technische Universitt Ilmenau
Ilmenau, Germany

4th Carsten Behn
Dept. of Mech. Engineering
Schmalkalden University of Applied Scienes
Schmalkalden, Germany
e-mail: c.behn@hs-sm.de

***Abstract*—Interacting with the environment, many robots would benefit from advanced tactile sensors complementing optical sensors in particular when operating under poor visibility. In nature, rats exhibit a prominent tactile sense organ, the so-called vibrissae. For instance, these enable rats to detect shape and texture information of objects based on few contacts. Since vibrissae consist of dead tissue, all sensing is performed in the support of each vibrissa, the follicle-sinus complex. Inspired by this characteristic measuring principle, we set up a mechanical model, consisting of a cylindrical, one-sided clamped bending rod, which is swept along a 3D object surface undergoing large deflections. In doing so, the focus is on both simulating the scanning sweep in order to determine the support reactions of the rod during object scanning and subsequently using these quantities in order to reconstruct a sequence of contact points as a basis for shape reconstruction. The simulated scanning sweeps include tip and tangential contacts, as well as longitudinal, lateral and axial slip. The object reconstruction reveals that simple scanning kinematics, e.g., passive dragging of the tactile sensor on a mobile robot, are sufficient in order to capture fragments of object shapes and thus to complement data gathered by optical sensors.**

Keywords–Vibrissa; tactile sensor; surface sensing; surface reconstruction;

I. INTRODUCTION

A prominent and particularly well researched sense organ are the mystacial vibrissae in the snout region of the rat, enabling the animals to gather information about object distances, orientations, shapes and textures, as well as fluid flows [1]–[4]. Basically, a vibrissa consists of a long and slender hair-shaft with no receptors along its length, which is conical and pre-curved [5][6] and supported by the Follicle Sinus Complex (FSC) [7][8]. Making contact with an object of interest, mechanical stimuli are transmitted through the hair-shaft to the FSC, where the actual sensing is realized by a wide variety of mechanoreceptors, which are radially and longitudinally distributed along the follicle [9]. Despite the fact that it is not conclusively clarified how exactly animals manage to determine object features, e.g., object shapes [10], the basic measuring principle of a vibrissa often serves as a paragon for developing tactile sensors.

In literature, a large number of vibrissa-inspired sensor principles with a focus on object scanning and reconstruction can be found. These approaches differ considerably in the modulation of the vibrissa-shaft and its support, the evaluated signals (observables) and the procedure of localizing contact points, as a basis for shape reconstruction. Usually, the focus is on the actual object reconstruction based on different measured signals at the base of a rod-shaped structure. However, a theoretical generation of the support reactions during object scanning is rarely taken into account. For the process of object scanning, two different approaches have been established in literature [11]: Firstly, an object can be scanned using a tapping strategy, i.e., tapping the artificial vibrissa against various points of the objects surface by small pushing angles. In doing so, the artificial vibrissa is retracted from the object right after the very first contact resulting in only small deformations of the artificial vibrissa. Therefore, many approaches use linear bending theory to accomplish the localization of the contact points [12]–[15]. For that purpose, curvature or torque information at the base of the artificial vibrissa [15] or even its base angles and/or moments [12]–[14] are evaluated. Secondly, there is the sweeping strategy. In this case, the rod is pushed against an object far beyond the very first contact, consequently undergoing large deformations and sliding over the object's surface. Therefore, a highly flexible and elastic artificial vibrissa must be used. In [11], a sweeping reconstruction algorithm was presented, which is based on repeatedly inferring from one contact point to the next one by continuous measurement of the moment and rotation angle at the base of the rod. This method dispenses the need of force measurements, but it is limited to tangential contacts along the artificial vibrissa and cannot be used for 3D reconstruction. Another reconstruction approach using the sweeping scanning strategy and evaluating moments and forces at the base of a rod was used in [16] and [17]. In [17], the focus was on the mechanical model, which was limited to a plane and treated analytically. In doing so, the support reactions were generated analytically in a first step, validated using an experimental setup and used for object reconstruction in a second step. With the goal of 3D object scanning, the authors in [16] used a hub load cell to measure the support reactions at the base of a steel wire, which was swept along several edged 3D objects. During scanning, lateral slip of the artificial vibrissa was prevented by adjusting the scanning direction based on the surface normal, which was determined exploiting the support reactions. In

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

contrast to [17], the support reactions were only measured but not generated theoretically. In [18], the authors modeled an artificial vibrissa using a multi-body system. There, the step of generating the support reactions in 3D space was included, but only for an a priori known external load. In addition, the applied force was always perpendicular to the artificial vibrissa, although contact forces at the tip of an artificial vibrissa may have arbitrary orientations in many real world scenarios. The model presented within this paper differs from [18] by considering a changing contact force resulting from the sweep of a rod along a mathematically described object surface. Adapting the model of [17] for 3D scanning and reconstruction, we focus on both, shape reconstruction of 3D objects with a wide-ranging curvature (in contrast to [16]) **and** a theoretically generation of the support reactions during object scanning. The latter provides an important basis for future parameter studies without the necessity of performing a large amount of experiments. During object scanning, tip and tangential contacts are taken into account. Objects are scanned on a straight trail without actively adjusting the scanning direction in order to prevent lateral slip.

The paper at hand is structured in the following way: In Section II the mechanical sensor model is presented and used in two steps: Firstly, we demonstrate, how the support reactions at the base of the sensor during a scanning sweep along a prescribed 3D object surface can be generated theoretically. Secondly, we present a procedure of reconstructing a sequence of contact points solely based on the support reactions, which might either be known from the previous step or by measurements. Both steps are implemented in a scanning and reconstruction algorithm. In Section III this algorithm is used for simulating scanning sweeps along an exemplary 3D surface, analyzing the support reactions at the base of the sensor. Afterwards, these signals are used in order to reconstruct multiple points on the object's surface. Finally, the results of the present paper are summed up and some future research subjects are identified in Section IV.

II. MODELLING

The mechanical model consists of a one-sided clamped rod (highly elastic probe) and an object in a fixed Cartesian (x, y, z)-coordinate system, see Figure 1. The rod is circular-cylindrically shaped and therefore characterized by the length L and a constant circular cross-section, resulting in a constant second moment of area I. It consists of a homogeneous and isotropic material with a constant Young's modulus E. From the outset, we introduce the following units of measure using the mentioned parameters of the rod in order to allow any kind of scaling [17]:

$$\begin{aligned} [length] &:= L, \\ [force] &:= EI/L^2, \\ [moment] &:= EI/L \end{aligned} \tag{1}$$

The object is assumed as a rigid body with a strictly convex, smooth surface $z = C(x, y)$. The scanning sweep of the rod along the object's surface is realized by a kinematic drive, in a way that the clamping position $P_0(x_0, y_0, 0)$ of the rod (system input) is shifted incrementally along a straight trail in the x-y-plane.

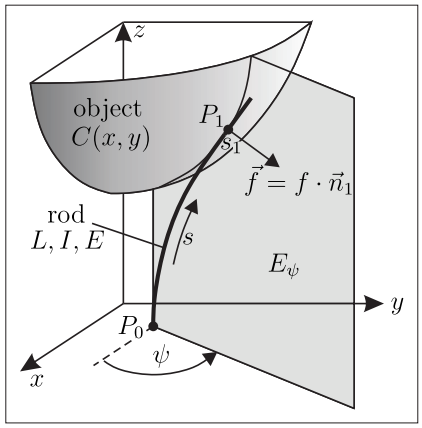


Figure 1. Model for object shape scanning and reconstruction – rod in contact with an object's surface.

After the very first contact between the undeformed rod and the object, the rod gets bent and slides along the surface, see Figure 1. This process is treated as a quasi-static one. As a consequence of the strict convexity of the object multi-point contacts between the object and the rod are excluded. Thus, each preset clamping position P_0 results in a single contact point $P_1(\xi, \eta, \theta)$ on the surface with some contact force $\vec{f}$ acting on the rod. Neglecting frictional effects (approximation for low-friction material pairings), the contact force $\vec{f}$ with magnitude f coincides with the outward pointing unit normal vector $\vec{n}_1$ of the surface, see Figure 1:

$$\vec{n}_1 = \frac{\vec{n}_0}{||\vec{n}_0||_2}, \qquad \text{with} \qquad \vec{n}_0 = \begin{pmatrix} -C_{,x}(\xi, \eta) \\ -C_{,y}(\xi, \eta) \\ 1 \end{pmatrix} \tag{2}$$

$$\vec{f} = f \cdot \vec{n}_1 \tag{3}$$

Due to the homogenity and isotropy of the rod and its support and the single force assumption the elastic line of the rod shrinks to one in a plane E_ψ with some unknown orientation angle ψ (see Figure 1) and the normal vector:

$$\vec{e}_\psi = \begin{pmatrix} -\sin(\psi) \\ \cos(\psi) \\ 0 \end{pmatrix} \tag{4}$$

Thus, parameterizing the elastic line of the rod by means of its slope angle $\varphi(s)$ in dependence on its natural coordinate arc length s yields:

$$\vec{q}(s) = \begin{pmatrix} \frac{dx(s)}{ds} \\ \frac{dy(s)}{ds} \\ \frac{dz(s)}{ds} \end{pmatrix} = \begin{pmatrix} \cos(\varphi(s))\cos(\psi) \\ \cos(\varphi(s))\sin(\psi) \\ \sin(\varphi(s)) \end{pmatrix} \tag{5}$$

$$\frac{d\varphi}{ds} = \kappa(s) \tag{6}$$

Using a dimensionless representation of Euler's constitutive law (mind (1)) the curvature $\kappa(s)$ writes

$$\kappa(s) = m(s) = -f \cdot \det(\vec{r}, \vec{n}_1, \vec{e}_\psi) \tag{7}$$

where $m(s)$ is the bending moment with respect to the deformation plane E_ψ and

$$\vec{r} = \begin{pmatrix} \xi - x(s) \\ \eta - y(s) \\ \theta - z(s) \end{pmatrix}. \tag{8}$$

Using (5), the derivative of (7) writes:

$$\kappa'(s) = m(s) = f \cdot \det\left(\vec{q}(s), \vec{n}_1, \vec{e}_\psi\right) \tag{9}$$

Together with (5), (6) and (9) we find the following system of Ordinary Differential Equations (ODE) of first order, describing the deformation of the rod in space.

$$\boxed{\begin{aligned} x'(s) &= \cos(\varphi(s))\cos(\psi) \\ y'(s) &= \cos(\varphi(s))\sin(\psi) \\ z'(s) &= \sin(\varphi(s)) \\ \varphi'(s) &= \kappa(s) \\ \kappa'(s) &= f \cdot \det\left(\vec{q}(s), \vec{n}_1, \vec{e}_\psi\right) \end{aligned}} \tag{10}$$

In contrast to the ODE system used in [17], the additional parameter ψ allows the deformation plane to rotate in space (with respect to the z-axis) as a consequence of lateral slip during object scanning. Thus, sweeping the rod along an object on a straight scanning trail, the path of the contact point over the object's 3D surface is not a priori known, as it is the case when scanning 2D object contours. Therefore, the procedure suggested in [17], to determine the rod's base position assuming a given contact point on the object is not appropriate for 3D surface scanning, because the base position corresponding to an arbitrarily preset contact point on the 3D surface would not necessarily lie on the specified scanning trail. Therefore, we proceed in inverse direction using the base position P_0 as system input as a basis for determining the corresponding contact position, what better reflects the practical process.

A. Generating the support reactions theoretically

The boundary conditions are formulated distinguishing between tip contacts and tangential contacts. For tip contacts the contact position s_1 along the rod is known: $s_1 = 1$. For tangential contacts s_1 is unknown but the condition $\vec{q}(s_1) \perp \vec{n}_1 \Leftrightarrow \vec{q}(s_1) \cdot \vec{n}_1 = 0$ must be fulfilled. This results in the following Boundary Conditions (BCs) for tip and tangential contacts, respectively:

tip:

$$\boxed{\begin{aligned} x(0) &= x_0 & x(1) &= \xi \\ y(0) &= y_0 & y(1) &= \eta \\ z(0) &= 0 & z(1) &= \theta \\ \varphi(0) &= \frac{\pi}{2} & & \\ & & \kappa(1) &= 0 \end{aligned}} \tag{11}$$

tangential:

$$\boxed{\begin{aligned} x(0) &= x_0 & x(s_1) &= \xi \\ y(0) &= y_0 & y(s_1) &= \eta \\ z(0) &= 0 & z(s_1) &= \theta \\ \varphi(0) &= \frac{\pi}{2} & \vec{q}(s_1) \cdot \vec{n}_1 &= 0 \\ & & \kappa(s_1) &= 0 \end{aligned}} \tag{12}$$

For each clamping position P_0 (system input) on the scanning trail, the Boundary-Value Problems (BVPs) (10)&(11), as well as (10)&(12) are solved (respecting the additional constraint $\vec{n}_1 \cdot \vec{e}_\psi = 0$) using a *Matlab*-algorithm. In doing so, the unknown parameters f, s_1, ψ, ξ and η are determined using shooting methods. The algorithm proceeds by pre-supposing tip contact, checking s_1 for contradictions (e.g. $s_1 > 1$) afterwards in order to decide which of the BCs (11) and (12) correctly describes the actual deformation state. Once the mentioned parameters are known, the six support reactions at the base of the rod are calculated in the following way:

$$\vec{f}_0 = \begin{pmatrix} f_{0x} \\ f_{0y} \\ f_{0z} \end{pmatrix} = -\vec{f}, \tag{13}$$

$$\vec{m}_0 = \begin{pmatrix} m_{0x} \\ m_{0y} \\ m_{0z} \end{pmatrix} = -\begin{pmatrix} \xi - x_0 \\ \eta - y_0 \\ \theta \end{pmatrix} \times \vec{f} \tag{14}$$

The presented model as well as the simulation algorithm are revised versions of those in [20]. The changes and improvements relate in particular to a transformation of the used coordinate system and a more general analytical formulation of the BCs (11)&(12).

B. Reconstructing contact points using the support reactions

For object reconstruction, we assume the support reactions (13)&(14) to be known either from simulations or measurements. In contrast to the previous step of generating the support reactions, this allows to determine the magnitude of the contact force f, its direction $\vec{n}_1$ and the orientation ψ of the bending plane E_ψ in advance:

$$f = ||\vec{f}_0||_2, \quad \vec{n}_1 = -\frac{\vec{f}_0}{||\vec{f}_0||_2}, \quad \psi = \text{atan2}(m_{0x}, -m_{0y}) \tag{15}$$

where *atan2* is the four-quadrant inverse tangent [21]. Having all parameters of the ODE system (10) at hand by use of (15), the initial values follow from the known support position P_0 and $\vec{m}_0$:

$$\boxed{\begin{aligned} x(0) &= x_0 \\ y(0) &= y_0 \\ z(0) &= 0 \\ \varphi(0) &= \frac{\pi}{2} \\ \kappa(0) &= -\vec{m}_0 \cdot \vec{e}_\psi \end{aligned}} \tag{16}$$

Thus, the step of object reconstruction requires the solution of an Initial-Value Problem (IVP) only. The contact point is localized by integrating (10)&(16) with the termination condition, that the curvature at the contact point is zero ($\kappa(s_1) = 0$).

III. RESULTS & DISCUSSION

For all simulations, we used an elliptic paraboloid as an exemplary object surface:

$$(x, y) \mapsto C(x, y) = 0.5x^2 + y^2 + h \tag{17}$$

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

with the object distance $h = 0.4$. The scanning sweeps were realized by shifting the clamping position P_0 on a straight scanning trail in negative x-direction with a constant lateral displacement $y_0 = -0.5{:}0.1{:}0.5$ from the coordinate origin, resulting in a total of eleven trails in the x-y-plane. In Figure 2, the signal strengths of all support reactions generated as described in Subsection II-A are indicated by the color-bars and plotted against the base positions during object scanning (scanning trails).

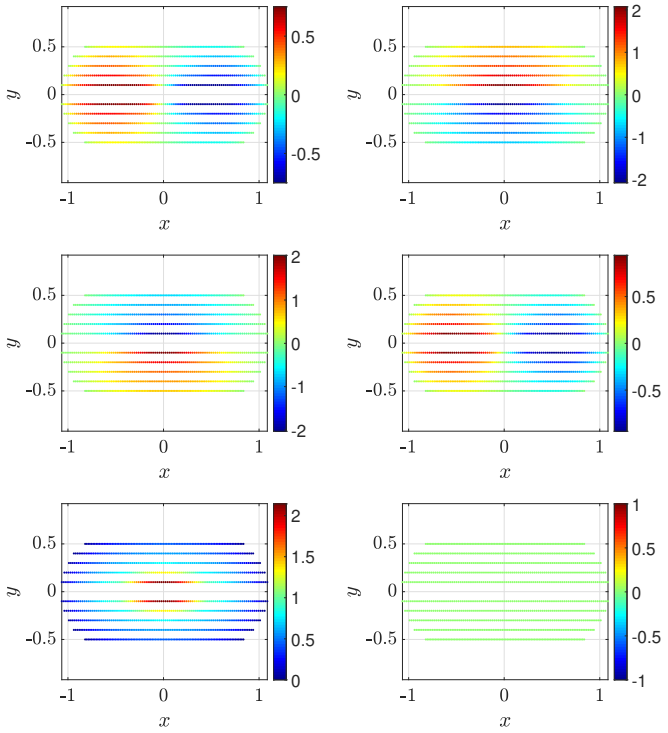


Figure 2. Signal strengths of the observables (support reactions) during object scanning on different scanning trails – reaction forces f_{0x}, f_{0y} and f_{0z} (TB) on the left and reaction moments m_{0x}, m_{0y} and m_{0z} (TB) on the right. The color-bars on the right indicate the signal strengths.

The reaction forces f_{0x}, f_{0y} and f_{0z} from Top to Bottom (TB) are shown on the left and the reaction moments m_{0x}, m_{0y} and m_{0z} (TB) on the right. The color-bars indicate the dimensionless signal strengths for each clamping position (mind (1)). The scanning trail $y_0 = 0$ is omitted in Figure 2 for two reasons: On the one hand, it represents a plane special case with longitudinal slip (contact point movement over the surface within the sensing plane) only, which is not in the focus of the present paper, since it was already analyzed in detail in [17]. On the other hand, the corresponding signals of this sweep are significantly higher compared to all other scanning sweeps and would therefore impair the comparability in Figure 2. It is striking that all components of the support reactions show some symmetric patterns, due to the symmetry of the object. The maximum values of all components increase for smaller values $|y_0|$, i.e., the closer the scanning trail is to the coordinate origin. Mind that all diagrams in Figure 2 have to be read from right to left due to scanning in negative x-direction. It is obvious that the component f_{0x} changes in sign from minus to plus for all chosen scanning trails. The component f_{0y} is negative for positive values y_0 and vise versa. The vertical component f_{0z} is always positive and a glance on the entirety of all trails already makes it possible to get the idea of some longish convex shape. The clamping moment components behave in a similar way as the reaction forces: The component m_{0x} is positive if $y_0 > 0$ and negative if $y_0 < 0$. The component m_{0y} always changes in sign from minus to plus. A great similarity can be seen between the components f_{0x} and m_{0y}, as well as f_{0y} and m_{0x}. The magnitude of m_{0z} must always be zero, since there is no twist of the rod due to the plane bending assumption. Even though especially the vertical component f_{0z} in Figure 2 might give a rough hint on the scanned object shape, the support reactions do not allow a more precise conclusion about the scanned object shape based on purely visual observation. Therefore, the support reactions from Figure 2 are used for the actual object reconstruction as described in Subsection II-B.

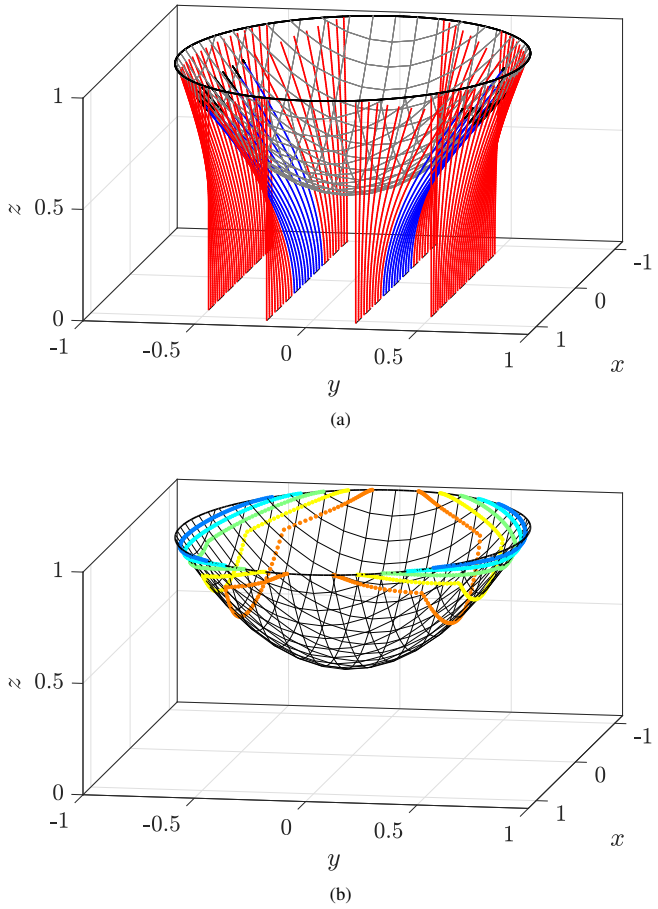


Figure 3. Object surface scanning and reconstruction: (a) Reconstructed elastic lines during object scanning – tip contacts $s \in (0, 1]$ in red, tangential contacts $s \in (0, s_1]$ in blue and $s \in (s_1, 1]$ in black; (b) Reconstructed sequences of contact points based on ten scanning sweeps.

Figure 3a shows the reconstructed elastic lines during sweeps along the elliptic paraboloid (17) on four exemplary scanning trails $y_0 \in \{-0.5; -0.2; +0.2; +0.5\}$. Tip contacts are colored in red. For tangential contacts, the interval $s \in (0, s_1]$ is colored in blue and the straight end $s \in (s_1, 1]$ in black. It can be seen that for each sweep, starting with the

very first contact between the undeformed rod and the surface, the rod bends around the object symmetrically and finally detaches from the object without any snap-off, as observed during plane object scanning [17]. This highlights the fact, that the 2D scanning scenario analyzed in [17] is a special case, which only occurs for special object geometries and arrangements. However, in a real world scanning scenario, geometries and orientations of scanned objects are not known in advance. Therefore, the special case considered in [17] can not be guaranteed and is therefore rather unlikely to occur in reality.

For large lateral displacements $|y_0|$, only tip contacts occur. In contrast, for lower lateral displacements $|y_0|$, tangential contacts increasingly occur. In these cases, the sweep starts with a sequence of tip contacts, continues with tangential contacts and finally ends with another tip contact phase. Of course, for tip contacts ($s_1 = 1$), no axial slip (contact point movement along the rod) but lateral slip (contact point movement along the object's surface) occurs during scanning. For tangential contact, axial and lateral slip occur simultaneously. In Figure 3b the original object surface is superimposed with the reconstructed sequences of contact points. The colors are arbitrary chosen and intended to make it easier to visually distinguish the individual sequences, each resulting from one scanning sweep. The sharp edges in the contact point sequences, in particular evident in the innermost (orange and yellow) sequences result from the transition from tip to tangential contact and vise versa. It is striking that the reconstructed points are unevenly distributed over the scanned surface in a way that especially the lateral area of the surface is characterized by a high density of points. As a consequence, a large reconstruction gap in the area below the center of the object remains. This is due to the fact that even though the trails were distributed equidistantly below the object (see Figure 2), there seems to be an area enclosed by the innermost orange sequences in Figure 3b, which cannot be reached by the rod. However, it can be assumed that the size of this gap might be reduced by scanning in different directions (e.g. perpendicular to the trails in Figure 2) or/and at different object distances h. Returning to the biological paragon, both ideas can be found in the whisking behavior of rats during object exploration [19]. Besides the reconstructed contact points in Figure 3b, the presented measuring principle provides the surface normals at each point, evaluating (15). Thus, the sensed data consisting of a 3D point cloud accompanied by normal vectors is very similar to the one provided by some optical systems, e.g., a laser range finder. This highlights that the presented measuring principle is highly suitable to complement optical sensors in robot exploration and path planning tasks.

IV. Conclusion

Within the paper at hand we presented a vibrissa-inspired tactile measuring principle for 3D object surface scanning and reconstruction. In doing so, we considered two consecutive processes separately: firstly, we analyzed the process of theoretically generating the support reactions at the base of a rod, which is swept along a 3D object surface. Secondly, we demonstrated how to use these quantities in order to reconstruct sequences of contact points on the object's surface. Both steps were implemented in an algorithm and simulated to demonstrate the general applicability. Instead of representing a self-contained investigation, the present paper should be seen as a preliminary concept study. However, it demonstrates that the presented measuring principle is well suited to complement optical sensors in the environmental exploration of robots. By clarifying which mechanical signal strengths are detected during object scanning and how these signals can be used for object reconstruction, we provided a basis for further investigations connecting the mechanical signals (observables) with the actual measurand (3D surface). In doing so, we aim to implement the presented concept into an intelligent tactile sensor in the future. The findings of the present paper are to be validated using an experimental setup in future works. In this context, the effect of some aspects, which are neglected within the presented paper (in particular friction and dynamical effects) on the reconstruction quality should be examined. In addition, the influence of some external disturbance factors, e.g., temperature variations and wind flows, have to be investigated. First experimental investigations identifying fluid flows were already presented in [22]. As the presented measuring principle can be realized with different materials and sensors, it can be assumed, that the temperature dependence of the overall system determines by the thermal properties of the used materials and force-torque-sensors.

References

[1] M. Brecht, B. Preilowski, and M. M. Merzenich, "Functional architecture of the mystacial vibrissae," Behav. Brain Res., vol. 84, pp. 8197, 1997.

[2] E. Guić-Robles, C. Valdivieso, and G. Guajardo, "Rats can learn a roughness discrimination using only their vibrissal system," Behav. Brain Res., vol. 31, pp. 285–289, 1989.

[3] G. E. Carvell and D. J. Simons, "Biometric Analyses of Vibrissal Tactile Discrimination in the Rat," J. Neurosci., vol. 10, pp. 26382648, 1990.

[4] T. J. Prescott, B. Mitchinson, and R. A. Grant, "Vibrissal behavior and function" Scholarpedia 6(10):6642, 2011.

[5] H. M. Belli, A. E. T. Yang, C. S. Bresee, and M. J. Z. Hartmann, "Variations in vibrissal geometry across the rat mystacial pad: Base diameter, medulla, and taper," J. Neurophysiol., vol. 117, pp. 1807–1820, 2016.

[6] D. Voges et al., "Structural Characterization of the Whisker System of the Rat," J. Sens., vol. 12, pp. 332–229, 2012.

[7] D. Campagner, M. H. Evans, M. S. E. Loft, and R. S. Petersen, "What the whiskers tell the brain," Neurosci. vol. 368, pp. 95–108, 2018.

[8] S. Ebara, T. Furuta, and K. Kumamoto, "Vibrissal mechanoreceptors". Scholarpedia. 12, pp. 32372. DOI: 10.4249/scholarpedia.32372, 2017.

[9] T. Furuta et al., "The cellular and mechanical basis for response characteristics of identified primary afferents in the rat vibrissal system," Curr. Biol., vol. 30, pp. 814–826, 2020.

[10] J. H. Solomon and M. J. Z. Hartmann, "Radial distance determination in the rat vibrissal system and the effects of Weber's law," Phil. Trans. R. Soc. B., vol. 366, pp. 3049–3057., 2011.

[11] J. H. Solomon and M. J. Z. Hartmann, "Extracting object contours with the sweep of a robotic whisker using torque information," Int. J. Robot Res. vol. 29, pp. 1233-1245, 2010.

[12] J. H. Solomon and M. J. Z. Hartmann, "Artificial whiskers suitable for array implementation: accounting for lateral slip and surface friction," IEEE Trans. Robot., vol. 24, pp. 1157–1167, 2008.

[13] D. Kim and R. Mller, "Biomimetic whiskers for shape recognition," In Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21-27 May 1995, pp. 1113-1119.

[14] M. Kaneko, N. Kanayama, and T. Tsuji, "Active antenna for contact sensing," IEEE Trans. Robot. Autom., vol. 14, pp. 278-291, 1998.

[15] A. E. Schultz, J. H. Solomon, M. A. Peshkin, and M. J. Z. Hartmann, "Multifunctional Whisker Arrays for Distance Detection, Terrain Mapping, and Object Feature Extraction," IEEE International Conference

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

on Robotics and Automation, Barcelona, Spain, 18-22 April 2005, pp. 2588-2593.

[16] T. N. Clements and C. D. Rahn, “Three-Dimensional Contact Imaging With an Actuated Whisker,” IEEE Trans. Rob., vol. 22, pp. 844–848, 2006.

[17] C. Will, C. Behn, and J. Steigenberger, “Object contour scanning using elastically supported technical vibrissae,” ZAMM J. Appl. Math. Mec., vol. 98, pp. 289-305, 2018.

[18] L. A. Huet, J. W. Rudnicki, and M. J. Z. Hartmann, “Tactile sensing with whiskers of various shapes: determining the three-dimensional location of object contact Based on mechanical signals at the whisker base.” Soft Robot., vol. 4, pp. 88–103, 2017.

[19] R. A. Grant, B. Mitchinson, C. W. Fox, and T. J. Prescott, “Active Touch Sensing in the Rat: Anticipatory and Regulatory Control of Whisker Movements During Surface Exploration,” J. Neurophysiol., vol. 101, pp. 862-874, 2008.

[20] L. Merker, J. Steigenberger, and C. Behn, “3D Recognition Of Obstacles Using A Vibrissa-like Tactile Sensor,” IEEE International Conference on Flexible and Printable Sensors and Systems (FLEPS), Glasgow, United Kingdom, 08-10 July 2019.

[21] MathWorks®, Support (R2020b), https://de.mathworks.com/help/matlab/ref/atan2.html, retrieved: October, 2020.

[22] M. Scharff et al., “An artificial vibrissa-like sensor for detection of flows,” Sensors, vol. 19, 2019.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

Generating Simulation Models From CAD-Based Facility Layouts

Rui Pinto, Susana Aguiar, Gil Gonçalves
Research Center for Systems and Technologies
Faculty of Engineering, University of Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
Email: {rpinto, saguiar, gil}@fe.up.pt

***Abstract*—The latest advances in industry have been boosted by the application of the Industrial Internet of Things (IIoT), which is being supported by the implementation of Cyber-Physical Production Systems (CPPS). In this context, simulation and optimization models have, for some time now, been used to reduce CPPS design complexity, implementation time, and operating costs throughout its life cycle. Considering the complexity and heterogeneity of CPPS components and their different application areas within the manufacturing context, simulation models may contain unreliable representation of the real system, since the behavior of the CPPS in a physical environment can be quite different from the same CPPS, specified in a simulation model, considering the same events. Thus, facing this inherent difficulty in reliably modeling a CPPS in a virtual environment, the authors propose a tool - *Layout CAD Interface*, which enables simulation modeling software with automatic generation capabilities of discrete-event simulation models. The tackled simulation software was *Simio*, and the generation of simulation models was based on Computer-Aided Design (CAD) files referring to shop-floor layouts.**

Keywords–Cyber-Physical Production System; Simulation; Simio; CAD.

I. INTRODUCTION

Cyber-Physical Production Systems (CPPS) are used whenever complex manufacturing systems need to communicate with the digital world, allowing their performance to be optimized and their efficiency improved. This type of system plays an increasingly important role in industrial processes and production control, leveraging the concept of intelligent factory in the context of the Industrial Internet of Things (IIoT). This are being used to build Internet-based architectures that facilitate remote control of currently isolated production systems, leading to the fusion between the physical and virtual worlds, which is the basis of IIoT applications

The complexity of production systems in the various production sectors is increasing and results from a greater need for flexibility and interoperability to produce low volumes and high variety of customized products, according to customer specifications. This is known as the *Mass Customization & Personalization* [1] production paradigm. To meet these requirements, virtual environments based on optimization and simulation models allow the real-time evaluation of the operational and safety characteristics of the real system. In this way, virtualisation acts as an interface between the real productive system and the decision makers, allowing the simulation and prediction of the future state of the system against different operating scenarios (e.g., different levels of demand) or stochastic events (e.g., a significant increase in demand).

The decision support systems, based on simulation and optimization, are particularly interesting because they allow simultaneously to predict and evaluate the functioning of the productive systems in a virtual environment. The main objective of the simulation and optimization methods is to model the individual aspects of the manufacturing equipment and the production lines with a high degree of detail, in order to minimize the costs involved. These methods have been used to anticipate the operational performance of work centers, production lines and, when considering more complex systems, the simulation and optimization models allow to create virtual environments of complete factories by detailing the dynamics of the flow of materials, the effective capacity and efficiency of the equipment, the time evolution of the Work-in-Progress (WIP) and the intermediate stocks, and the existing wastes associated with the tasks of rework, setup, or movement.

Manufacturing systems modeling and simulation is a research topic that has been addressed by the academic community and industry. Oztemel and Gursev [2] defend the importance for companies to primarily understand the features and content of the Industry 4.0 for potential transformation from machine dominant manufacturing to digital manufacturing. These digital environments are highly oriented to the data that different components/machines produce or collect. Jain [3], Jain *et al.* [4] and Sacco *et al.* [5] argue that there must be virtual models of the factory - Virtual Factory, in order to build real system behavior analysis applications. Also, these models also serve as generators of information, which can be used in evaluating the configuration and adaptation of the current real systems.

Regarding the use of simulation models, as representations of the real physical systems, developing these models imply studying the real system based on physical and/or mathematical models, in order to replicate their behavior. The execution of these models result in a deeper knowledge about the behavior of the system in different contexts, allowing to evaluate the impact of alternative strategies of operation. However, the development of reliable simulation models is not trivial and requires a high level of knowledge. If the model is not a valid representation of the system under study, the result of its execution will yield little useful information about the actual system. Building a detailed and reliable model of the production system is a task that can last from a few days to a few weeks or months depending on the complexity of the system that is being represented. Furthermore, several of the current simulation tools require knowing the simulation language being used. With this in mind, new approaches for CPPS simulation should address the challenges associated with the simulation model specification and development.

The R&D Portuguese National project PRODUTECH-SIF [6] focuses on the development of new methodologies to promote the integration of the design, engineering, commis-

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

sioning, operation and maintenance phases regarding CPPS. As such, these methodologies will significantly reduce the operating costs and the implementation cycles of improvement actions, since they will allow the validation, in a virtual environment, of the key characteristics of the systems and reduce the risks of their implementation in the real scenario.

The main outcome consists in a tool, designated *CPPS Design Platform*, which contemplates a reference design model for the standardization of the design, implementation and virtual installation of CPPS. This tool, available as a service within an *IIoT Platform* for production information management and service delivery, is supported by several modules, such as: *Simulation Module*, for modeling stochastic behavior and events on the factory floor; *Optimization Module*, for the generation of optimized production plans; *Layout CAD Interface*, for semi-automatic creation of simulation models based on Computer-Aided Design (CAD) shop-floor layout files; and *Communication Interfaces*, which provide interconnection between shop-floor equipment and external information systems, such as Enterprise Resource Planning (ERP). *IIoT Platform* also supports system diagnosis and the calculation of performance indicators for further improvement, available in the *CPPS Performance & Diagnostic Platform*, and the generation and scheduling of production plans, available in the *CPPS Planning & Operations Platform*.

The work described in this paper focuses on the *Layout CAD Interface* development, which is part of the *CPPS Design Platform*, and tackles the automation of the simulation model generation, by enabling simulation software with automatic generation capabilities of simulation models, or part of them, depending on their complexity and customization. Thus, the simulation model creation and execution is based on shop-floor layout information, which is contained in CAD files [7], and process data, which is contained in ERP systems. In this work, the simulation software used was *Simio* [8] [9].

The structure of the paper is as follows: in Section II, literature review is provided. In Section III, the overall architecture of the system being developed is described and, in Section IV, the *Layout CAD Interface* tool functionalities are detailed. In Section V, a CPPS use case scenario is introduced, where the *Layout CAD Interface* is validated and, finally, in Section VI the conclusion and future work are presented.

II. Related Work

Modern practice is to use CAD in the planning stage of facility layout design, which consists on organizing facilities, such as industrial equipment, in the shop-floor. The problem of organizing facilities in the most efficient way possible, currently motivates active research work, in a field known as Facility Layout Planning [10]. CAD refers to the use of computers, or more specifically, CAD software, to aid in the creation, modification, analysis, or optimization of a design. These designs often represent technical, engineering, electronic/automation, mechanical and manufacturing drawings. In the context of manufacturing, CAD specific tools are used to develop and optimize 2D and 3D factory layout solutions, by enabling the design layouts of machines, production lines or entire manufacturing facilities. The main goal is to enable factory managers to try different layout scenarios, arranging industrial equipment according to product demands or equipment functionality, in order to identify the most efficient layout scenario, regarding waste elimination in material flows, inventory handling and management.

The term CAD gained its notoriety with the work developed by Douglas T. Ross in the computer-aided design project [11] at MIT. The long-term goal of the project was to use a computer as an active partner to the designer, thus reducing the elapsed time and resources necessary in completing the design process. Regarding commercial software, *AutoCAD*, which is *Autodesk*'s flagship CAD software, has grown to become one of the most widely used CAD program for 2D non-specialized applications. Nowadays there are CAD systems for all of the major computer platforms. Some of the commercial tools most commonly used were developed by *Autodesk*, *Dassault Systems*, *Siemens PLM Software*, *PTC Creo* and *SketchUp*. Depending on the type of software tool used, the CAD drawings can be 2D or 3D.

The most popular CAD file format is the *DraWinG* (DWG), a native proprietary file format for *AutoCAD*, used for storing 2D and 3D design data and metadata. DWG is a binary file, so all types of information are stored efficiently, including 3D elements and photos. Since several CAD practitioners do not have access to licensed software, such as *AutoCAD*, *Autodesk* proposed the *Drawing Exchange Format* (DXF), which is a neutral file exchange, i.e., uses an intermediary neutral format to translate data between CAD systems, enabling data interoperability between *AutoCAD* and other CAD systems. DXF is a vector graphic file format that stores only 2D drawings. Regarding file size, DXF is a plain-text format and complicated drawings are usually slightly larger than DWG. Considering drawing content, DXF only supports 2D shapes, such as lines, polygons, circles and text. Other neutral file exchange are the *ISO 10303 – STandard for the Exchange of Product model data* (STEP) and *Initial Graphics Exchange Specification* (IGES).

Since CAD files typically define the static arrangement of facilities in a shop-floor, only structural analysis can be achieved with these tools, typically performed in the planning phase. For a proper analysis of the manufacturing system behavior and production dynamics, along with the planning, evaluation and monitoring of the relevant processes in the operational phase, simulation tools have been used for decades. Also, for further system analysis in the context of virtual commissioning as "emulation" for the real system, simulation models must use real-time process data collected from the shop-floor equipment. This can be accomplished by the integration of simulation with tools such as ERP.

Dias *et al.* [12] underline the frequently usage of CAD, process simulation and information systems software tools for production system design, and claims that they have been used with low levels of integration, resulting in duplicated work, incoherence and errors during the planning phase. Facing this, the authors propose an integrated approach for systematic system design, based on these three software classes (CAD, process simulation and information systems software tools).

Regarding the manufacturing virtual commissioning based in simulation, according to AbouRizk and Mather [13], a drawback of computer simulation is the investment required to build simulation models, such as time to learn the languages and human effort to put together the model. It is very important to model accurately the system in a simulation model, in a

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

way that allows sufficiently exact predictions about the real system behavior. To tackle this challenge, the authors refer to approaches that automatically generate simulation models, specifically structural approaches, where model generation is based on data describing the structure of a system, typically in the form of factory layout data from relevant CAD-systems. These approaches are also known as "data-driven model generation", and can be distinguished into either supporting the planning phase or the operational phase of a factory.

According to Bergmann and Strassburger [14], challenges for automatic model generation may be: 1) Incomplete data or low level of detail within external systems; 2) Generation of dynamic/complex behaviour, i.e, dealing with missing information regarding the dynamic behaviour of the system; 3) Support of cyclic approaches involving multiple generation cycles, i.e., how to incrementally generate models, allowing manually added details to survive in such cyclic model generation scenarios; and 4) Support of multiple life cycle phases of the production system, i.e., how models of the planning phase can be adapted to work in the operational phase.

Considering the literature for automatic model generation, proposed solutions may be supported by CAD-based data, referring to the structure of a system, or non CAD-based. Regarding CAD-based automatic model generation, Zhai *et al.* [15] proposed an integrated simulated method, which combines CAD, virtual reality and discrete event simulation techniques, for virtual factory engineering. Regarding static simulation for factory layout assessment, the authors use input information of the factory plant that includes the shapes and sizes of the facilities, operation rooms, among other. This information may be collected from CAD files, however it is not clear how this is accomplished.

Also, Dias *et al.* [16] proposed an approach for integrating simulation and CAD. *AutoCAD* is used for the layout design and *WITNESS* is the simulation platform. *Microsoft Access* is used as the database for data exchange between *AutoCAD* and *WITNESS*, enabling the integration of both software tools.

Hoffmann *et al.* [17] described concepts for the systematic design of manufacturing system models, based on model libraries and standardized recipes for the design of component models from CAD data. The authors used *CIROS* for the plant simulation tool, which provides import filters for STEP/IGES file formats to collect geometric data from the plant. The authors claim that manual simplification of overly complex geometry data may become necessary. Also, if manual hierarchical structuring of the CAD data into objects, sections and components is not performed, the generated simulation model is not usable for the functional modelling, nor would be a simulation based on such model.

Lorenz and Schulze [18] focus only on the layout based model problem generation, considering every type of layout, such as barber shops, job shop, road traffic or copper smelter. Their approach can be used for any simulator with a language oriented model description interface, and uses the DXF file format as the layout basis for the model generation. The DXF file is inputted into a modeling filter, which eliminates unneeded layers of the file and transforms the DXF into a Proof Layout file, for a native animation layout. Later, domain specific information regarding simulation and animation specifics is appended to the native animation layout, resulting in an animation layout and, after analysis, a structure file. This structure file contains all classes, objects with names and positions, messages, bars, plots and paths in a text format, which is the input for the simulation model generators.

AbouRizk and Mather [13] propose an approach for simplifying the process of building and experimenting with computer simulation models of heavy earth-moving construction operations, where simulation models are automatically generated from high-level descriptions in CAD. In this approach, a commercial CAD tool (*MicroStation*) is extended for specify construction information, such as facility/components to be constructed, construction methods and resources, and sharing these information with the simulation tool (*Visual Slam*). The authors propose an add-on tool referred to as the "Intelligent Data Manager", to exchange data between the two systems. Other *MicroStation* add-ons, specifically developed for this type of application, are used to relate CAD objects to the construction objects and extract all construction objects from the CAD model and translate them into simulation information.

Moorthy [19] proposed an automated method for generating simulation models and 3D model animations directly from CAD drawings. First, facility layout drawings are developed using the *FactoryCAD* (run from within *AutoCAD* application). Then, from the *FactoryCAD*, a Simulation Data Exchange (SDX) is generated, which is an ASCII text file, containing a compilation of physical data, manufacturing and production data, and simulation data. Finally, the SDX file is used as an input to generate discrete event simulation models in several tools. The simulation model generates animation data that is subsequently available for dynamic viewing and analysis in the *FactoryTalk VIEW* environment.

In this work, the authors want to achieve CAD-based automatic simulation model generation in *Simio*, since this is one of the most popular simulation tools among CPPS scenarios. Also, the capability of representation of dynamic information and components in the simulation model is highly desirable, since layout planning is a task performed several times during the entire manufacturing system lifecycle, according to ever changing product demand (both variety and quantity). Considering the literature review, the authors believe that there are still gaps in current solutions, considering the product described.

III. Overall System Architecture

Part of the work developed within the PRODUTECH-SIF [6] project addresses, in an integrated way, the design, evaluation, operationalization and optimization of CPPSs. These bring several benefits, from a dynamic response in terms of production to the variable demand for new products, as well as real-time optimization of production and operation processes (particularly value chains). Thus, the scenario where the product diversity is high and its life cycle is tendentiously shorter, the tools and methodology being developed allow for the optimized design, evaluation and operation of CPPSs for each reality. More specifically, the work focus is on the development of an integrated tool, enabling factory simulation and monitoring, planning and scheduling of operations, and production diagnosis for performance improvement.

In the case of the PRODUTECH-SIF project, it was developed a solution, designated *CPPS Design Platform*, for flexible high performance CPPS design, used for the integration of

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

the design, implementation and installation phases of CPPS, with the objective of reducing implementation time and risks, and operating costs throughout their life cycle. This solution contemplates the simulation of the manufacturing environment and processes of the factory floor, as well as the optimization of production plans, taking into consideration real time information of the factory shop-floor, such as operating status of the equipment, production orders, and WIP. Also, the *CPPS Design Platform* integrates a CAD-based automatic simulation model generation solution, designated *Layout CAD Interface*.

The *CPPS Design Platform* provides a service available within the *IIoT Platform*, used for micro-services management, which consist in modules involved in the design and management of the CPPS. The *IIoT Platform* also provides other services for the planning and scheduling of lot sizing, capacity assessment, collaborative process management, and real-time scheduling, which ensure efficient, integrated and adaptive management of the CPPS and the diagnosis of the productive system in an agile and automatic manner, classification of the improvement needs and consequent identification of the resolution method to be implemented. Services regarding planning and scheduling are provided by the *CPPS Planning & Operations Platform* and services regarding methodologies for diagnosis are provided by the *CPPS Performance & Diagnostic Platform*. Figure 1 represents the overall architecture of the system being developed within the PRODUTECH-SIF project, regarding to solutions dedicated to the life cycle management of CPPSs, from their design and development, to their operational management and improvement.

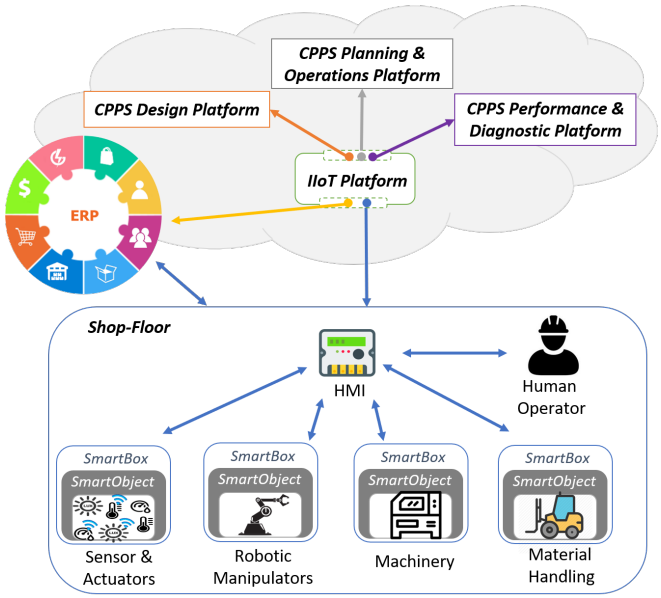


Figure 1. System Architecture Within the PRODUTECH-SIF [6] Project.

In this context, the *IIoT Platform* serves to interconnect productive equipment, information systems, sensors, devices, people and products that are part of a production system, allowing the collection and sharing of information among all stakeholders providing services. The integration and virtualization of shop-floor entities into the *IIoT Platform*, such as sensor & actuators, robotic manipulators, machines and material handling equipment, is achieved by the encapsulation of such entities, using the *SmartObject* and *SmartBox* concepts [20], also developed within the scope of this project.

The *CPPS Design Platform* is expected to run simulation and optimization models, which can use relevant information of the factory shop-floor in real time during their operation time. These models will include information from ERP systems, such as product process orders, available materials and routes between the different shop-floor entities, as well as real time information from the *SmartObjects*/*SmartBoxes*, such as equipment state, failure events and WIP. In addition, these simulation and optimization models may be integrated with information from the *CPPS Planning & Operations Platform*, aiming at solving the problem of integration of different decision levels, from design, planning and scheduling. Finally, the results of the simulation execution will be used as input to the *CPPS Performance & Diagnostic Platform*, in order to run diagnostic and performance improvement models based on the calculation of several Key Performance Indicators (KPIs).

A. CPPS Design Platform

The *CPPS Design Platform* is supported by a simulation modeling software, integrated with various modules, namely the *Simulation Module*, *Optimization Module*, *Layout CAD Interface*, and *Communication Interfaces*. The most relevant communication interface of the *CPPS Design Platform* provides interconnection with the *IIoT Platform*, which manages the overall interactions with ERPs, *SmartObjects* and *SmartBox* within the shop-floor, the *CPPS Planning & Operations Platform* and the *CPPS Performance & Diagnostic Platform*. Figure 2 represents the architecture of the *CPPS Design Platform*.

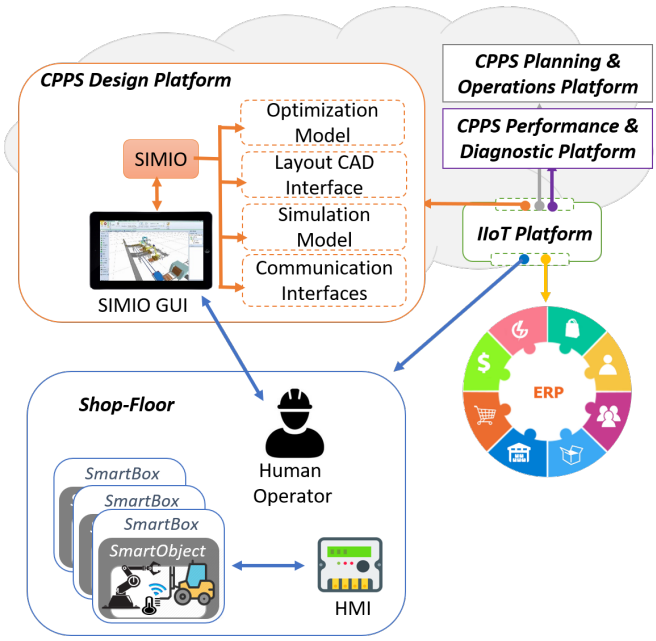


Figure 2. *CPPS Design Platform* Architecture.

The simulation modeling software is the main component of this platform, which, through the use of simulation models representing the dynamics between all shop floor equipment, operators, conveyors, as well as the various operations and processes between them, allows the study of the behavior of

the system. For this, *Simio* is used. *Simio* is a simulation, production planning and scheduling software to support the object modeling paradigm. This approach is suitable to model a CPPS, where complex stochastic simulation models are constructed by combining different objects representing different components of the system under consideration. By using the *Simio* GUI, the human operator can build and run 3D animated models from a variety of systems.

In *Simio*, the concept of *Process* exists, which represents a custom logic that can be included in the model objects, so that they behave and react to events in a certain way. This custom logic can be used to capture / free resources, assign values to different variables, change the network connections used, evaluate alternatives, etc. Every *Simio* process consist of several interconnected *Steps*, which can already exist, predefined, in *Simio* or be added to the *Simio* by the user, in order to extend the base framework functionality. These *User Defined Steps* can be developed by the user using the *Simio* API.

The *Optimization Module* will implement a new methodology related to operational planning and scheduling, to support the simulation module. The *Optimization Module* will permit to introduce variation in the generation of production plans, contemplating the associated risks. Thus, a production plan can be evaluated not only for its viability at the time the plan was generated, but also for its robustness over time, considering the associated risk in the order of execution of the various operations (critical operations, which depend of factors such as demand, existence of stocks and maintenance of equipment).

Before executing the simulation, the *Optimization Module* obtains an estimate of the quantities of the manufacturing orders, i.e., the optimization model will determine the products, the quantity and the planning period in which the production should take place. Taking into account this new dimension, the *Optimization Module* aims to optimize production plans, which will later make available to *Simio*, as a starting point to execute a new simulation. In this case, the *Optimization Module* will be based in the *IBM-CPLEX* tool [21].

Finally, the *Layout CAD Interface*, detailed next, provides to *Simio* automatic generation capabilities of simulation models, or part of them, depending on their complexity and customization, based in facility layout CAD files.

IV. Layout CAD Interface

The *Layout CAD Interface* is a software application developed within the context of the PRODUTECH-SIF project, and it is one of the modules of the *CPPS Design Platform*, which enables simulation modeling software to automatically generate simulation models based on shop-floor layout CAD files. The main goal of the *Layout CAD Interface* is to provide to a human operator means for simplifying CPPS simulation modeling. When using the *Layout CAD Interface*, the human operator is capable of extracting facility layout and other information from a CAD file. Also, this tool is an Java software application, independent from CAD and simulation modeling specific software. Currently being developed, this tool is, so far, capable of interpret DXF format CAD files. For this, it uses the *Kabeja* open Java library [22]. *Kabeja* is a library for parsing, processing, and converting *Autodesk*'s DXF file format to other output formats, such as SVG, JPEG, PNG, TIFF, PDF and XML.

Due to the nature of the DXF file type, extracting information of industrial equipment and facility layout is not trivial. As seen before, the DXF file is a vector graphic file format that only supports simple 2D shapes, such as lines, polygons, circles, and text [23] and [24], other more complex designs are supported in, e.g., the DWG file format. In this first version, the *Layout CAD Interface* only supports the DXF file type, since it is the most popular open-source format among every licensed or unlicensed CAD-based software.

As for the integration with simulation modeling tools, the *Layout CAD Interface* only outputs simulation data in a specific format interpretable by *Simio*, since it is the simulation modeling tool used among the partners involved in the project. Figure 3 represents an UML Use Case Diagram of the *Layout CAD Interface* tool.

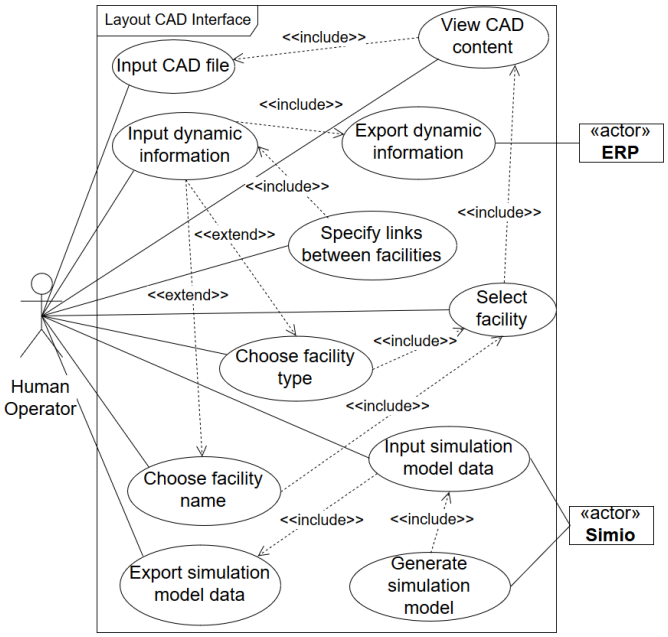


Figure 3. UML Use Case Diagram of the *Layout CAD Interface*.

Considering the UML Use Case diagram of the *Layout CAD Interface*, the system actors represented are every entity that plays a role in the system. In this case, the represented actors are the *Human Operator*, the *ERP* and the *Simio*. The *Human Operator* represents the person responsible for the CPPS design and simulation modeling, and which will in fact use the *Layout CAD Interface* for simplifying the model generation. The *ERP* and the *Simio* actors represent external systems that will interact with the *Layout CAD Interface*. Regarding the use cases, which represent the features interactions or the actions that an actor can do in the system, and the relations among use cases and actors:

- *Input CAD file* - The *Human Operator* can import to the *Layout CAD Interface* a CAD file, which contains a static information of the shop-floor layout. Currently, the *Layout CAD Interface* is limited to DXF file formats.
- *View CAD content* - Since, currently, the *Layout CAD Interface* is limited to DXF formats, the shop-floor layout representations considered are in 2D. This use

case represents the *Human Operator* capability of viewing, within a canvas, a 2D graphical representation the information contained in the CAD file.

- *Input dynamic information* - While static information of the shop-floor layout can be collected from the CAD file, dynamic information, such as equipment characteristics and links between facilities in a given process, are collected from ERP systems. If the *Layout CAD Interface* is integrated with an ERP, the *Human Operator* can import dynamic information to include in the simulation model. Currently, the *Layout CAD Interface* is prepared to import a CSV file, containing exported dynamic information from a ERP.
- *Export dynamic information* - The *ERP*, representing ERP systems integrated with the *Layout CAD Interface*, are capable of exporting CSV files, containing dynamic information regarding facilities, such as equipment characteristics and links between facilities in a given process.
- *Select facility* - Since the 2D shop-floor layout is represented within a canvas, the *Human Operator* can select facilities, by drawing a rectangle around the desired facility. This selection allows the *Layout CAD Interface* to collect from the CAD file information about the size and position of the facility selected.
- *Choose facility name* - The *Human Operator*, after selecting a facility, must choose a proper name. For this, the *Human Operator* has the possibility of manually writing the facility name, or, if dynamic information is available from an ERP system, the *Human Operator* can choose a name from a predefined list of names regarding every existing facility.
- *Choose facility type* - The *Human Operator*, after selecting a facility, can choose a facility type to be included in the simulation model. Considering *Simio*'s models, typical facility types are: server, workstation, source, sink, and vehicle.
- *Specify links between facilities* - If dynamic information is available from an ERP system, the *Human Operator* is capable of define links between facilities, i.e., connections between different facilities, regarding material and product flow in a given process.
- *Export simulation model data* - After defining all static and dynamic information to be included in the simulation model, the *Human Operator* can export the simulation model data, which will be used as input to generate the respective simulation model. The information considered contains the definition of all facilities names, types, location on the shop-floor (x and y coordinates), dimensions (length and width) and the links between facilities. Since the simulation modeling tool used is *Simio*, the simulation data output is in a CSV file, with the format of the content interpretable by *Simio*.
- *Input simulation model data* - The CSV file generated by the *Layout CAD Interface* tool is formatted to be imported into *Simio*. This is achieved by the use of a *Simio User Add-in*. In this case, the *Simio User Add-in*, built in C#, using the *Simio* API, is capable of importing the CSV file exported by the *Layout CAD Interface* and generate the simulation model.
- *Generate simulation model* - By executing the *Simio User Add-in*, *Simio* imports the CSV file containing simulation data outputted by the *Layout CAD Interface*, to generate the simulation model.

V. LAYOUT CAD INTERFACE VALIDATION

With the purpose of validating the developed *Layout CAD Interface*, two test case scenarios were defined, which will be described next.

A. Test Case 1

Test Case 1 represents a process layout situation, where the design of the shop-floor facilities aims to arrange facilities according to their function. In this case, several equipments within a given department are represented. Also, it is assumed that only access to static data is available. Figure 4 represents the *Layout CAD Interface* and the respective simulation model layout in *Simio*, regarding Test Case 1.

In order to validate the *Layout CAD Interface* in Test Case 1, the *Human Operator* inputs the CAD file, already containing the respective facility layout, which is then available in the canvas. For this, button number 1 can be used - *Open CAD File* in Figure 4 and view the CAD content in the canvas (number 6 in Figure 4). Then, to select a given facility in the canvas, the *Human Operator* can click and hold in one of the facility corners and drag until the opposite corner, so that the equipment is inside a red square. To save the facility, pop-ups will appear, in order to manually input the name and choose a type. In this test case, there are five workstations (*mvc1*, *mvc2*, *mvc3*, *mvc4*, and *mvc5*) and two servers (*mbj1* and *mbj2*). For this, it can be used the button number 3 - *Save Facility* in Figure 4.

Finally, after all the facilities have been saved, the output file is produced in a standardized format, to be used in *Simio*, interpreted by the *Simio User Add-in* used. For this, it can be used the button number 5 - *Save File* in Figure 4. One limitation, regarding the placement of facilities within a simulation model, is the orientation of the facility. In this case, all facilities are placed with a 90^{o} orientation, since *Simio* API doesn't currently allow adjusting this parameter.

B. Test Case 2

Test Case 2 represents a product layout situation, where there is a production line and the facilities are arranged according to a particular production sequence. In this case, it is represented a simple wine bottling process, consisting of four equipments with different functions: fill the bottles with the wine; seal the bottle with the cork; labeling the bottle; bottle packaging. At the end of the process, a vehicle transports the wine bottles to a warehouse. Also, it is assumed that access to both static and dynamic data is available. Figure 5 represents the *Layout CAD Interface* and the respective simulation model layout in *Simio*, regarding Test Case 2.

Regarding the *Layout CAD Interface* in this test case, besides inputting the CAD file, the *Human Operator* also loads dynamic information exported from an ERP system, by the input of the CSV file, containing the list of the available facilities. For this, it can be used the button number 2 - *Open*

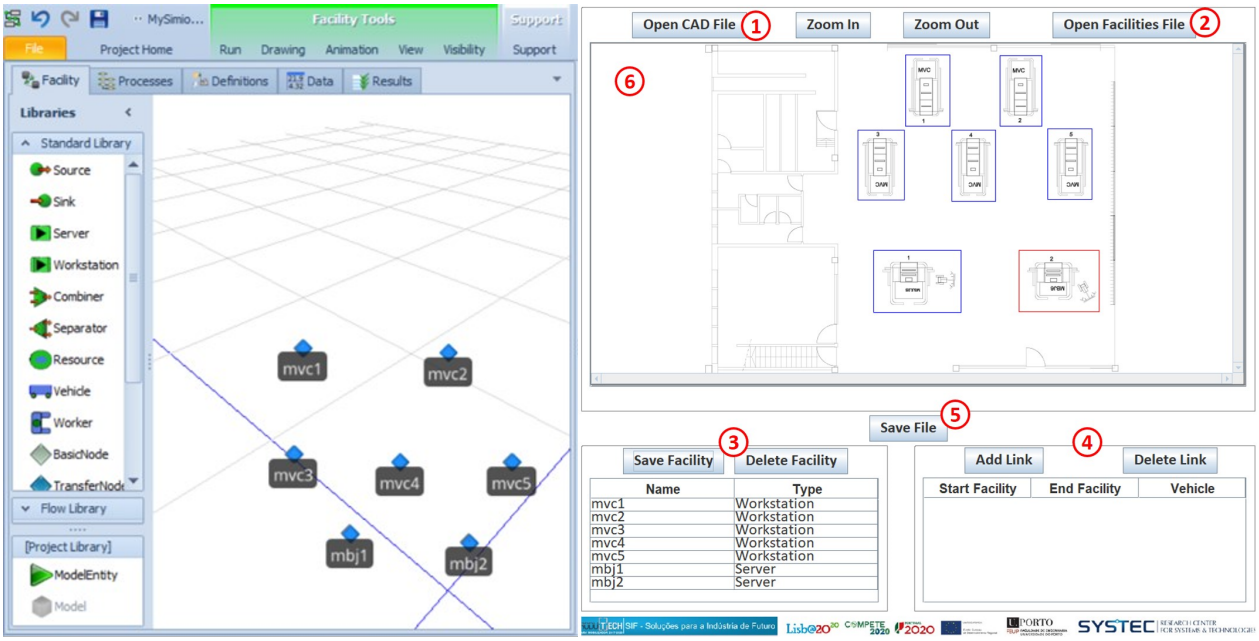


Figure 4. *Layout CAD Interface* and simulation model layout in *Simio*, regarding Test Case 1.

Facilities File in Figure 5. In this case, there is one source (*source1*), four workstations (*Filler*, *Cork_crew*, *Labelling*, and *Packing*), one sink (*warehouse*), and one vehicle (*Transport1*). When selecting a facility in the canvas, a pop-up appear, which allows to choose a name and type from a drop-down list.

Finally, the *Human Operator* can specify links between facilities in order to provide information regarding the production process flow. He can choose, from the predefined list of facilities, the facility that marks the process start and end points, and the transport type facilities used. The option ”none”, will always be available and should be the choice if no vehicles are used to transport the pieces between the start and end point of the link. For this, it can be used the button number 4 - *Add Link* in Figure 5. In this case, five links are defined, where only the link between the *Packing* and *Warehouse* facilities have material transport. The other links use conveyor belts for transportation. As in Test Case 1, the output file can be produced and used in *Simio* (Figure 5). Regarding the links between facilities, the tool limits the type of facilities linked to each other, according to the *Simio* capabilities. In this case, it is only possible to link source or server with Workstation/Vehicles and again with a sink. Also, moving facilities, such as vehicles, may be difficult to generate automatically, since these components usually are not represented in static CAD files.

VI. Conclusion and Future Work

The current document refers to the development of a tool, designated as *Layout CAD Interface*, focused in enabling the simulation software *Simio*, with the capability of semi-automatic generation of discrete-event simulation models regarding CPPS. This tool was part of the development of the *CPPS Design Platform*, within the PRODUTECH-SIF project, which supported the design and maintenance of CPPS. Regarding the simulation model generation, while the *Layout CAD Interface* contributed to the static information of the model (shop-floor layout), the dynamic information (system behavior and production) was collected from the ERP systems and directly from the shop-floor equipment.

The *Layout CAD Interface* validation results showed that the tool proposed is suitable for automatic generation of simulation models, considering both process (Test Case 1) and product (Test Case 2) layouts. While production layouts are suitable for mass production with less job variety, process layout are suitable for moderate production with more job variety. According to the different levels of product demand during the whole lifecycle of a manufacturing system, both scenarios will greatly benefit from the usage of such a tool, since flexible layout reconfiguration and earlier planning for cost reduction are frequent needs.

Regarding future work, the *Layout CAD Interface* may be extended to support 3D designs, which will permit the replacement of the *Simio* default simulation objects for the real design models of the facility to be included in the simulation. This may be achieved by adding the support of CAD file formats that support 3D designs. Also, further definition with the input of the dynamic components of the simulation model are needed, in order to enable the fully automatic generation of modules, such as: WIP, production plan within a given time, facility scheduling, production capacity and down-times, alternative process flow and bill of materials.

Acknowledgment

This paper and the research behind it would not have been possible without the exceptional support of all partners within the project “programa mobilizador PRODUTECH SIF – Soluções para a Indústria do Futuro” (Solutions for the Industry of the Future), which embodies a comprehensive response towards the development and implementation of new production systems, to meet the challenges and opportunities of the Industry 4.0.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6

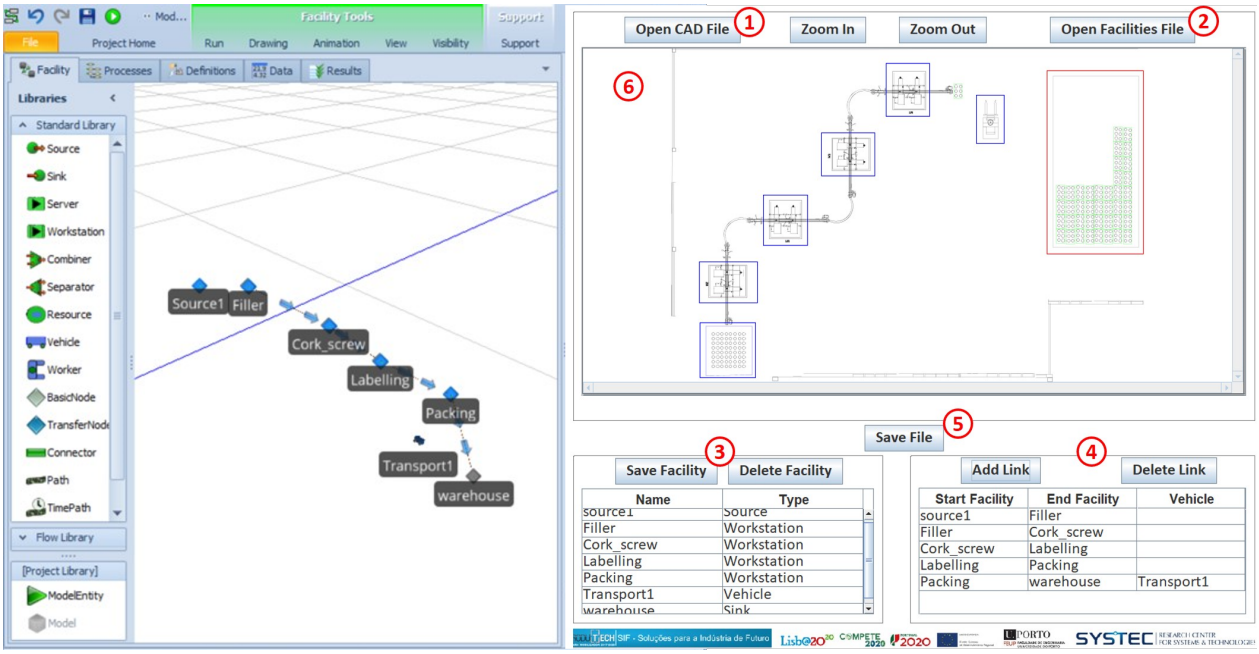


Figure 5. *Layout CAD Interface* and simulation model layout in *Simio*, regarding Test Case 2.

REFERENCES

[1] S. J. Hu, "Evolving paradigms of manufacturing: From mass production to mass customization and personalization," Procedia CIRP, vol. 7, 2013, pp. 3 – 8, forty Sixth CIRP Conference on Manufacturing Systems 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212827113002096

[2] E. Oztemel and S. Gursev, "Literature review of industry 4.0 and related technologies," Journal of Intelligent Manufacturing, vol. 31, no. 1, 2020, pp. 127–182.

[3] S. Jain and G. Shao, "Virtual factory revisited for manufacturing data analytics," in Proceedings of the 2014 Winter Simulation Conference. IEEE Press, 2014, pp. 887–898.

[4] S. Jain, N. Fong Choong, K. Maung Aye, and M. Luo, "Virtual factory: an integrated approach to manufacturing systems modeling," International Journal of Operations & Production Management, vol. 21, no. 5/6, 2001, pp. 594–608.

[5] M. Sacco, G. Dal Maso, F. Milella, P. Pedrazzoli, D. Rovere, and W. Terkaj, "Virtual factory manager," in International Conference on Virtual and Mixed Reality. Springer, 2011, pp. 397–406.

[6] "PRODUTECH-SIF," URL: http://mobilizadores.produtech.org/en/produtech-sif?set_language=en [accessed: 2020-09-14].

[7] T.-C. Chang and R. A. Wysk, Computer-aided manufacturing. Prentice Hall PTR, 1997.

[8] C. D. Pegden, "Simio: a new simulation system based on intelligent objects," in Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come. IEEE Press, 2007, pp. 2293–2300.

[9] D. T. Sturrock and C. D. Pegden, "Recent innovations in simio," in Proceedings of the 2011 Winter Simulation Conference (WSC). IEEE, 2011, pp. 52–62.

[10] R. Pinto, J. Gonçalves, H. L. Cardoso, E. Oliveira, G. Gonçalves, and B. Carvalho, "A facility layout planner tool based on genetic algorithms," in 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Dec 2016, pp. 1–8.

[11] S. A. Coons and R. W. Mann, Computer-aided design related to the engineering design process. MIT Electronic Systems Laboratory, 1960.

[12] L. Dias, G. Pereira, P. Vik, and J. A. Oliveira, "Layout and process optimisation: using computer-aided design (cad) and simulation through an integrated systems design tool," International Journal of Simulation and Process Modelling, vol. 9, no. 1/2, 2014, pp. 46–62.

[13] S. AbouRizk and K. Mather, "Simplifying simulation modeling through integration with 3d cad," Journal of construction engineering and management, vol. 126, no. 6, 2000, pp. 475–483.

[14] S. Bergmann and S. Strassburger, "Challenges for the automatic generation of simulation models for production systems," in Proceedings of the 2010 Summer Computer Simulation Conference. Society for Computer Simulation International, 2010, pp. 545–549.

[15] W. Zhai, X. Fan, J. Yan, and P. Zhu, "An integrated simulation method to support virtual factory engineering," International Journal of CAD/CAM, vol. 2, no. 1, 2002, pp. 39–44.

[16] L. Dias, G. Pereira, P. Vik, and J. A. Oliveira, "Integrated systems design in an automotive industry-using cad and simulation in layout and process optimization," in 11th International Conference on Modeling and Applied Simulation, MAS 2012. Caltek Srl, 2012, pp. 326–334.

[17] P. Hoffmann, R. Schumann, T. M. Maksoud, and G. C. Premier, "Virtual commissioning of manufacturing systems a review and new approaches for simplification." in ECMS. Kuala Lumpur, Malaysia, 2010, pp. 175–181.

[18] P. Lorenz and T. Schulze, "Layout based model generation," in Proceedings of the 27th conference on Winter simulation. IEEE Computer Society, 1995, pp. 728–735.

[19] S. Moorthy, "Integrating the cad model with dynamic simulation: simulation data exchange," in WSC'99. 1999 Winter Simulation Conference Proceedings.'Simulation-A Bridge to the Future'(Cat. No. 99CH37038), vol. 1. IEEE, 1999, pp. 276–280.

[20] L. Neto, G. Gonçalves, P. Torres, R. Dionísio, and S. Malhão, "An industry 4.0 self description information model for software components contained in the administration shell," in The Eighth International Conference on Intelligent Systems and Applications. International Academy, Research, and Industry Association, 2019.

[21] C. Bliek1ú, P. Bonami, and A. Lodi, "Solving mixed-integer quadratic programming problems with ibm-cplex: a progress report," in Proceedings of the twenty-sixth RAMP symposium, 2014, pp. 16–17.

[22] "Kabeja," URL: http://kabeja.sourceforge.net/ [accessed: 2019-06-12].

[23] "Dxf specification - autodesk," URL: https://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_enu.pdf [accessed: 2020-09-14].

[24] I. Kumari and A. M. Magar, "Dxf file extraction and feature recognition," International Journal of Engineering and Technology, vol. 4, no. 2, 2012, pp. 93–96.

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-798-6