Proceedings of the 25th International Conference on System Theory, Control and Computing, Iasi, Romania, October 20-23, 2021

Real-Time Motion Control of an Electric Driven OMR using a ROS to Matlab Bridged Approach

Constantin-Catalin Dosoftei
Department of Automatic Control and Applied Informatics
Gheorghe Asachi Technical University of Iasi
Iasi, Romania
constantin-catalin.dosoftei@academic.tuiasi.ro

Alexandru-Tudor Popovici
Department of Automatic Control and Applied Informatics
Gheorghe Asachi Technical University of Iasi
Iasi, Romania
alexandru-tudor.popovici@academic.tuiasi.ro

Petru-Razvan Sacaleanu
Department of Automatic Control and Applied Informatics
Gheorghe Asachi Technical University of Iasi
Iasi, Romania
petru-razvan.sacaleanu@student.tuiasi.ro

Cristina Budaciu
Department of Automatic Control and Applied Informatics
Gheorghe Asachi Technical University of Iasi
Iasi, Romania
cristina.budaciu@academic.tuiasi.ro

2021 25th International Conference on System Theory, Control and Computing (ICSTCC) | 978-1-6654-1496-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSTCC52150.2021.9607163

***Abstract*—The ability to remotely monitor and track the location of autonomous mobile robots assures more control to the logistics companies, which guarantees the supply chain management is more efficient and secure. Therefore, many efforts and studies are focused on providing information about the current and past position of systems involved in the logistics chain, through the use of sensors and GPS-like systems. In recent years, Omnidirectional Mobile Robots are increasingly involved in complex industrial environments due to their navigation capabilities for complex trajectory planning together with task assignments.**

This paper presents an evaluation study of integrated development for an Omnidirectional Mobile Robot that is designed to serve in a dynamic logistic environment using Marvelmind Indoor Positioning System. The overall supervised system can be efficiently developed using Robot Operating System nodes, the path planning nodes benefiting from using Matlab-Simulink. This paper relates the work-in-progress of a real omnidirectional mobile robots (OMRs) platform subject to indoor localization with Marvelmind IPS network. This study emphasizes the need for consistent data taken from the sensors placed on the robot, the validation being a very important step for accurate robot localization.

***Keywords*—omnidirectional mobile robot, ROS environment, logistic navigation system**

I. INTRODUCTION

The field that is with the most significant increase in the development of mobile robots is the logistics area due to the impact of increasing the popularity of e-commerce. This is illustrated by the fact that over 4 million commercial robots will be installed in over 50K warehouses by 2025 [1]. In logistic applications there are operational two types of autonomous mobile robots (AMRs), classified by the environment in which they navigate: unmanned ground vehicles (UGVs) that operate while in contact with the ground and unmanned aerial vehicles (UAVs) that can fly - commonly known as a drone, even used together for complex applications in warehouse space where tracking requirements at all stages are stringent [2]. In the current logistic facilities, there is typically a high-traffic environment, with narrow aisles among diverse obstacles where UGVs must be capable of moving with reaching parameters of efficiency. In order to succeed in such an environment where the UGVs manoeuvrability is very important, OMRs are the proper solution already started to be implemented in modern logistic areas.

In a logistical environment, the collaborative autonomous mobile robots are integrated with other complementary ones in order to improve processes and workflows. This is usually achieved by performing repetitive tasks – transportation of goods between locations, collaboratively working with operators for picking and sorting job. The inertial navigation system is appropriate for most mobile platforms, while the high performance is restricted by its hardware cost. With the development of the GPS and its successful application in the field of entertainment, the integration of the GPS and other on-board sensors can also be used on mobile robots [3].

The fundamental basis of the embedded real-time robotics system was developed by the authors in previous study [4], being the architecture concept that has to be validated by using a highly accurate ($\pm 2\,\mathrm{cm}$) method for indoor localisation. Typical experiments are performed, so the longitudinal translation motion was studied for the tracking performance in the square trajectory experiment and the results are comparatively analysed both from IPS and estimated odometry position.

The paper is structured as follows: In section II, an overview on related work is given. In section III, the OMRs indoor localization system is described, followed by SW architecture developed as a bridge between MATLAB-Simulink interface and Robot Operating System (ROS) services. Section IV is dedicated to experimental setup. Some limitations and difficulties are reported as a conclusion of our work.

II. RELATED WORK

Many studies have investigated the issues specific to OMR about the phenomena taking place between the wheels and

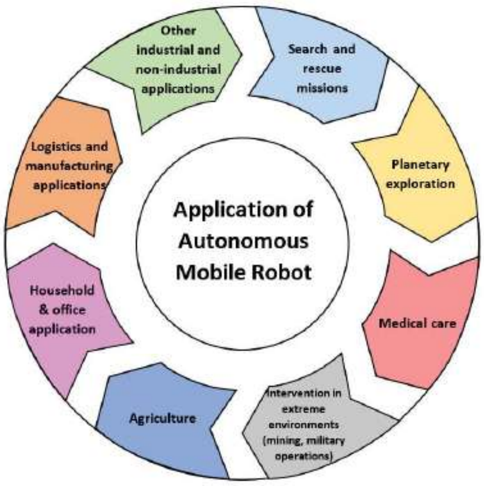


Fig. 1. AMRs applications.

different types of floor. The slippage determines losing velocity, having an important impact on lateral displacement, which affects the positioning estimation accuracy. Therefore are necessary different corrections for compensation for this inconvenience.

There are two main directions to reduce the impact of slippage: the usage of coefficients for corrections (linear or non-linear motion compensation) obtained from simulations [5]–[7] where are tested different scenarios regarding the contact forces between wheels and floor because these play a key role in the analysis of OMRs dynamics, and the second, by using external sensors for precise positioning to measure in the real-time pose of robot and using compensation algorithm to minimize the error between the current and the desired position [2], [8], [9].

The actual paper investigates the second direction through using localization system in indoor environment based on ultrasonic beacons. This work is part of an ongoing research project, ROSY-LOGISTIC, which aims to develop a handling and transport solution in complex logistic environments using a fleet of autonomous omnidirectional mobile robots coordinated by a warehouse management system. At the current stage, the testing and validation of the localization of the experimental model in laboratory conditions is considered by executing typical motion scenarios with the verification of the technical performance parameters.

III. OMR Indoor navigation and position

The planning and tracking trajectory for the mobile robots require as accurate knowledge as possible about the current position and orientation of the mobile robot. In outdoor applications, global positioning system (GPS) technology offers good positioning accuracy and is used in many automated driving applications. Unfortunately, for indoor localization, GPS technology cannot offer similar reliability because the signals of the satellites lose much strength when penetrating a building. There are several approaches for indoor localisation, called indoor positioning system (IPS), with their own advantages and limitations [9], [10].

In this paper, it is used for indoor localization a solution developed by Marvelmind Robotics (Starter Set HW v4.9-IMU-NIA), with a promised accuracy of $20\,\mathrm{mm}$ [11]. The working principle is based on a network composed of fixed ultrasonic beacons linked by radio interface in license free Industrial, Scientific, and Medical (ISM) band and a mobile beacon, called hedgehog, installed on the target which must be localized, in our case on OMR. The mobile beacon contains and an OMR (accelerometer + gyroscope + compass module). Location of OMR is calculated based on the propagation delay, also named time of flight (ToF), of the acoustic signal between stationary beacons grouped in the network and the hedgehog using trilateration. The distances between beacons is recommended to be $30\,\mathrm{m}$ and coverage area for this system with four stationary beacons is up to $1000\,\mathrm{m}^2$. The system is completed by a modem with role by central controller of the system and communications with all beacons is done through radio communication for a $433\,\mathrm{MHz}$ band.

A. The OMR hardware architecture

The architecture of the OMR platform used in the current study is structured on two hierarchical levels.

The low level part implements a hardware abstraction layer (HAL), which is centered around the vehicle controller (STM32F103R) that manages the real-time control tasks (inverse kinematic calculations for motion actuation through closed loop speed control using encoders for each of the four DC motors, direct kinematics for odometry information, and other sensors or peripheral integration).

The high level part implements a navigation abstraction layer (NAL), with functionality dedicated to communication, navigation and processing sensors data (i.e. inertial measurement unit (IMU), light based detection and ranging system (LiDAR), IPS and depth camera) is managed by an NVIDIA Jetson Nano embedded computer running a GNU/Linux distribution that hosts the ROS nodes.

B. The vehicle controller software

The vehicle controller is designed to receive relative motion commands from multiple sources, prioritize between them, apply acceleration and speed saturation to reflect the mechanical abilities of the platform and then ensures that the motors are actuated accordingly. It uses the inverse kinematic model to control the OMR's motion by calculating the angular speed of each wheel ($\omega_i, i = \overline{1,4}$) based on the relative linear robot velocity components (v_x, v_y) and the angular speed (Ω). Inverse kinematic model of the OMR, in the matrix representation, is calculated in [4] and is presented in (1), while the platform mechanical parameters are presented in Fig. 2 and their specific values are listed in Table I.

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \frac{1}{R}\begin{bmatrix}1 & 1 & -(l_x+l_y)\\ 1 & -1 & -(l_x+l_y)\\ 1 & 1 & (l_x+l_y)\\ 1 & -1 & (l_x+l_y)\end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ \Omega\end{bmatrix} \tag{1}$$

The vehicle controller uses the direct kinematic model (2) [4] to determine the actual OMR's relative speed to ground based on the angular speed of each wheel measured with the feedback quadrature encoders.

$$\begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} = \frac{R}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \frac{-1}{l_x+l_y} & \frac{-1}{l_x+l_y} & \frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{2}$$

The obtained estimated velocity information is passed to the higher level software to performs odometry calculations.

C. The high performance embedded computer

For the high level functionalities of the OMR an embedded computer with advanced processing capabilities, NVIDIA Jetson Nano, is used for running the navigation tasks and data acquisition from the LiDAR and the video/depth camera which require larger bandwidth and computation resources. Also at this level the Marvel IPS hedgehog is connected for receiving localization information usable for navigation and validation.

The embedded computer commands the vehicle controller using a virtual serial COM port (VCP) through which it demands relative motion speeds and receives the actual relative motion speeds as feedback.

TABLE I
DIMENSIONS OF OMR

Parameter (symbol)	*Value*	*Unit*
Distance l_x	0.294	[m]
Distance l_y	0.200	[m]
Wheel radius (R)	0.076	[m]

IV. ROS INTEGRATION WITH MATLAB

ROS is an environment for developing robotics software. It is a collection of libraries and tools that aim to facilitate the work of creating behaviour across a wide diversity of robotic platforms. The software is structured as a large number of generic modules, developed in a variety of languages (C++, Python, Ruby, Java, LISP, Perl and others) [12], which exchange messages to one another.

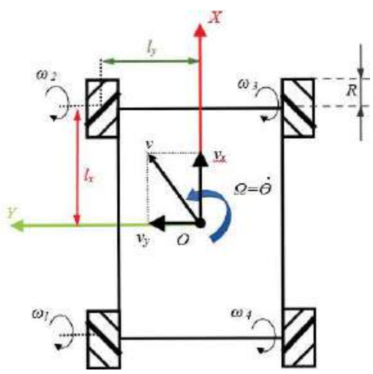


Fig. 2. The velocity vector in the coordinate system of OMR.

The philosophy of this open-source software encourages global robotics developers to cooperate and force the limit. The independent software modules which solve one or more tasks (running simultaneously) can be illustrated using a graph model in which the vertices ar the ROS nodes and the edges are the messages(topics) exchanged between them. In Fig. 3 are depicted the nodes running for implementing the OMR application.

A. ROS components

A ROS node is an instance of an executable, a process. In ROS there is core or master node that provides a declaration and registration service which makes it possible for nodes to find each other and exchange data.

A node can encapsulate a sensor, an actuator, a processing or monitoring algorithm, but every node that starts running needs to declare itself to the master. A node can be declared as a Subscriber (the purpose would be to 'listen' to a message type on a certain topic) or as a Publisher (its purpose would be to 'talk' or to 'send' a certain message through a certain topic). The most important requirement for a successful connection between two nodes (publisher and subscriber) consists in the message type that is being sent and received (published and subscribed). ROS offers a standard set of message types, but also allows creating user defined types.

A topic is a data transport system based on a Publisher/Subscriber system. One or more nodes are able to publish data to a topic. In the same way, one or more nodes are able to subscribe (or to request) data from a certain topic.

One of the strongest features of ROS is the powerful development toolset. The tools provided by ROS support introspecting, debugging, plotting and visualizing the state of the system being developed. The publish/subscribe mechanism, which is the one used in the communication between ROS nodes, allows to spontaneously introspect the data flowing through the system, making it easy to comprehend and debug issues as they occur. The ROS tools also include a collection of graphical and command line utilities that simplify development and debugging, like *rviz* and *rqt*.

B. The OMR's ROS structure

From the ROS point of view, the OMR used for the applied research of this paper runs the nodes presented in Fig. 3 of which most are implemented in C++ and Python, except the navigation node that is generated using a Matlab-Simulink model and contains the custom trajectory controller used for the following experiments.

The Matlab generated node registers as a subscriber to the odometry and IPS topics for obtaining real-time positioning information and in turn controls the OMR's motion speed relative to ground by publishing messages on the velocity command (*/cmd_vel*) topic of the robot.

Although the odometry and velocity command topics use standard ROS message types, the Marvel IPS node provided by the manufacturer uses a custom message type that requires additional steps for the Matlab node to be able to subscribe to the topic.

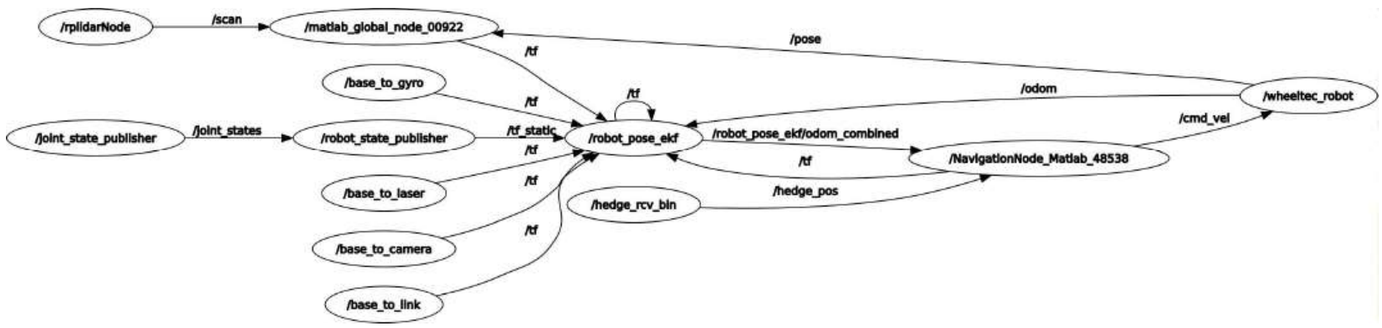


Fig. 3. ROS Software graph implemented in application controller.

C. IPS ROS node for Matlab-Simulink integration

Marvelmind has made available a ROS package (*marvelmind_nav*) that makes possible the acquisition of data from the IPS, that includes the current position of the hedgehog. In order to be used, the package needs to be installed in the ROS workspace of the OMR, operation that can be using the following steps:

- open a terminal on the embedded computer (locally or remote)
- switch to the sources directory (*src*) of the ROS workspace
- clone using git [13] or download the contents of the ROS Marvelmind package repository [14] into a new folder named $marvelmind\_nav$ in the current path
- build the package: *catkin_make -only-pkg-with-depts marvelmind_nav*
- start the hedge position receive node with the following comand, where the x in $ttyACMx$ should be replaced with the proper device number (usually 1) assigned to the VCP device that appears when connecting the hedge to the embedded computer on the USB port: *rosrun marvelmind_nav hedge_rcv_bin /dev/ttyACMx*

The previously started node ($hedge\_rcv\_bin$) works as a publisher and sends the position data through three possible topics: $hedge\_pos$, $hedge\_pos\_a$ and $hedge\_pos\_ang$. Unfortunately the message types of this topics are custom and it most likely requires additional steps for the topics to be available in the Matlab-Simulink model of the navigation node [15].

V. Laboratory Environment and Experiments

A. The experimental setup

The experiments were carried out in the lab environment depicted in Fig. 4 where there were installed four stationary beacons and the modem module used for configuring and monitoring the IPS network.

From the software point of view, Matlab was chosen for modelling the control system because it is a powerful tool for testing control systems satisfying the need for rapid control prototyping. Facilities offered by the Robotics Systems Toolbox has contributed to increase the interoperability with OMRs for whose development the ROS environment is more and more preferred.

The pose controller implemented as a Matlab model (represented in Fig. 5 generates the longitudinal and lateral speeds which are transmitted through the ROS network to the vehicle controller on the OMR. The controller tries to approach the target point received from the waypoint planner and when it reaches within a predetermined radius of the target point reports to the way-point planner that command is completed.

Besides the current target coordinates, the way-point planner signals to the trajectory controller if the current destination is the last point in the buffer or not. In case there is at least another way-point to follow, the trajectory controller simply sets the OMR fixed velocity vector towards the target, otherwise it controls the magnitude of the velocity vector using a saturated PI controller.

The initial posture of the OMR for all experiments was set as close to $(X, Y, \theta) = (0, 0, 0)$ as possible regarding the configured IPS map in order to ease the data analysis. The trajectory was designed to return to the origin in order to reduce the dead-time between experiments.

The value for angular speed is kept at 0 to investigate the omnidirectional translation abilities of the platform and also to observe effects of slippage regarding its orientation.

B. Position Control performances for the test reference points

The experiments were conducted using a square-shaped (one-meter side) trajectory. The red lines in Fig. 6 is the reference trajectory which is generated by the trajectory path block from Matlab-Simulink. The green curve is the estimated trajectory calculated based on the relative speeds reported by the vehicle controller and integrated by the ROS odometry node from OMR that sends it to the Matlab node. During the longitudinal displacement the estimation from odometry is relatively consistent with reference trajectory, while during the lateral displacement the OMR clearly deviates from the reference trajectory. The cumulative odometry errors prevent the open-loop odometry method from reaching zero state error. To minimize the impact usually are deducted experimental different coefficients for correction. This sensitive issue with reaching the target is solved by using the controller based on the IPS feedback. The blue points represent the real time trajectory of OMR and the performance of final positioning is improved, in the limit of tolerance given by IPS specified in section III.

Fig. 4. Panoramic (distorted) view of the laboratory environment for robot navigation experiment. The ultrasonic beacons are pointed by red arrows.

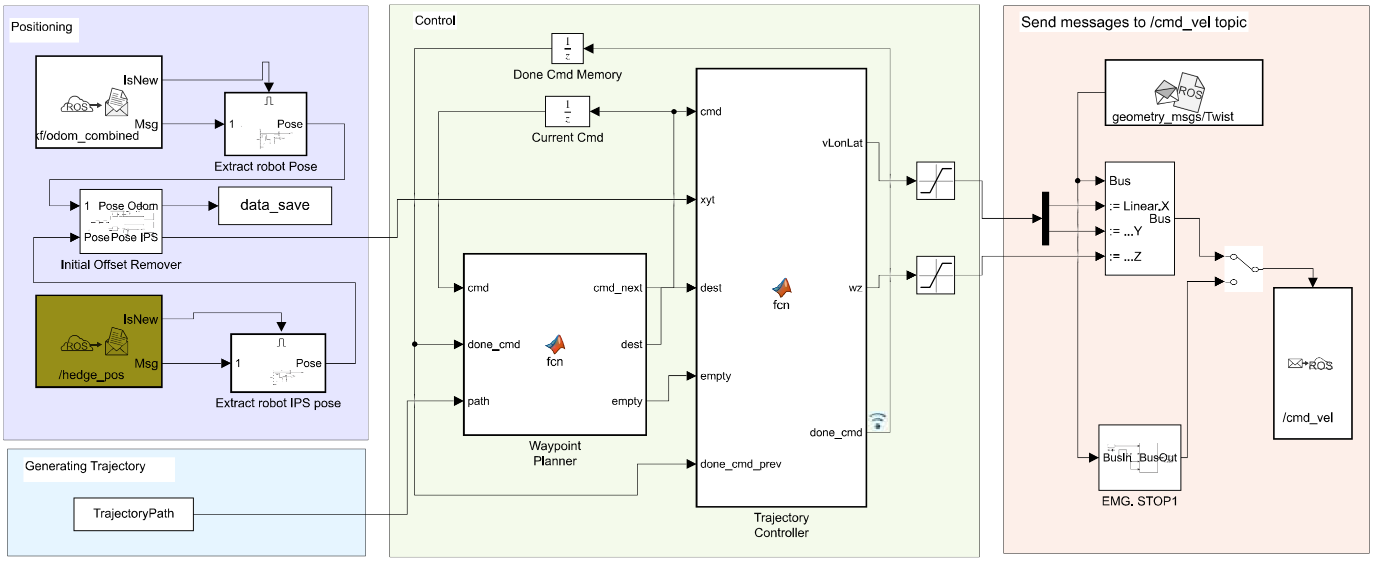


Fig. 5. ROS application node implemented in Matlab.

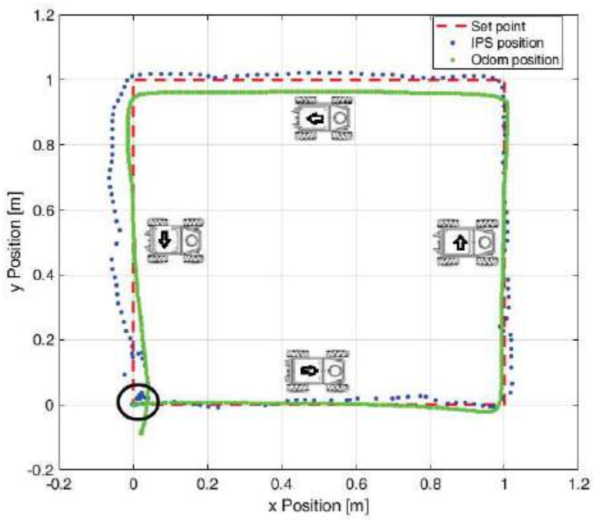


Fig. 6. The comparison between estimated and real position on the square trajectory.

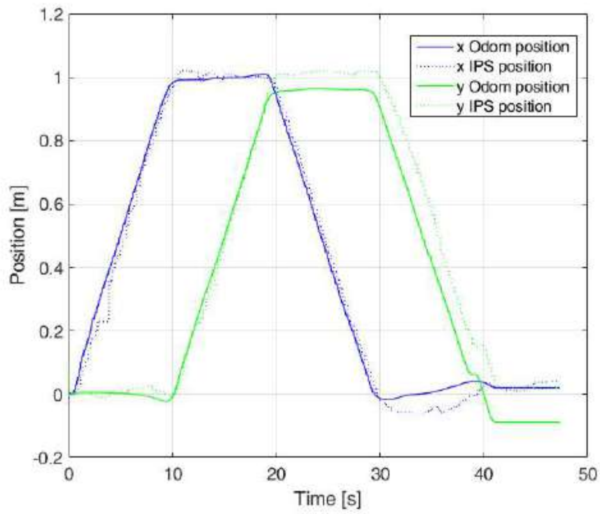


Fig. 7. Displacements of the OMR in translation motion along the square trajectory.

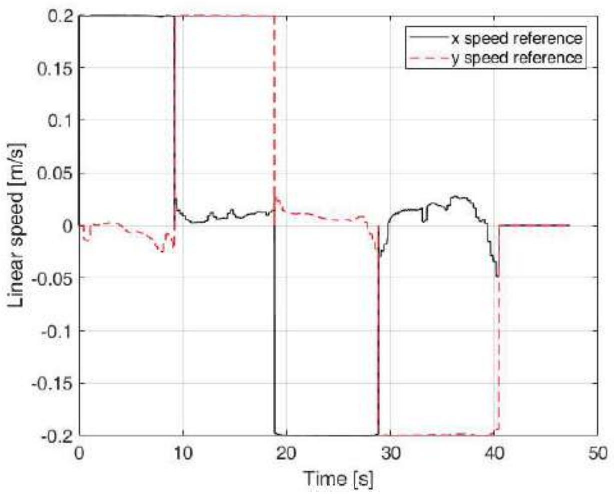


Fig. 8. Linear speeds reference to OMR in translation motion along a square.

Fig. 7 illustrates the displacement comparison curves on the X-axis and Y-axis showing the difference between the estimated odometry position (continuous line) and the IPS reported position (doted line). While for the blue curves the reference tracking error is minor, determined mainly by the latency of the IPS system and the saturation of the demanded OMR acceleration by the vehicle controller, in the green curves the difference is significant, in detrimental to odometry estimation.

Fig. 8 illustrates the longitudinal and lateral speeds demanded by the trajectory controller. Certain oscillations are visible around the zero value of the direction that is expected to not register movement. The behaviour can be explained by the response of the trajectory controller to different disturbances during the motion, e.g: initial miss-alignment of the OMR local axis and map axis, slipping of the wheels or the IPS' localization jitter.

VI. Conclusion

The conducted experiment proved that the ultrasound based IPS is a feasible solution especially for validating control strategies, but also for closed loop position control of an OMR even when simple control strategies are used.

Although the ability to translate in any direction of the OMRs is an advantage, in real world implementations slippage occurs, especially during lateral motion [5], fact that makes the estimated odometry based on the wheels' angular speed monitoring quite unreliable for ample movements, consequence that may require external position measurement systems in order to achieve the needed positioning accuracy for specific applications.

Considering the demand for highly accurate positioning specific for logistic applications of OMRs and given that even IPS solutions may suffer certain limitations (like limited sample rate, latency, blind-spots, shadowing or jittering) it is of interest to combine IPS with other methods, i.e. the odometry based on wheel's encoder signals in order to obtain better results during critical operations.

Acknowledgment

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI – UEFISCDI, project no. PN-III-P2-2.1-PTE-2019-0731, contract no. 19/2020, within PNCDI III, second stages.

References

[1] A. Research. (2021) Abi research homepage. [Online]. Available: https://https://www.abiresearch.com/

[2] I. Kalinov, "Development of a heterogeneous robotic system for automated inventory stocktaking of industrial warehouse," Ph.D. dissertation, Skolkovo Institue of Science and Technology, Moscow, 12 2020. [Online]. Available: https://www.skoltech.ru/app/data/uploads/2020/12/thesis4.pdf

[3] H.-J. Lee and H. Yi, "Development of an onboard robotic platform for embedded programming education," *Sensors*, vol. 21, no. 11, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3916

[4] C.-C. Dosoftei, A.-T. Popovici, P.-R. Sacaleanu, P.-M. Gherghel, and C. Budaciu, "Hardware in the loop topology for an omnidirectional mobile robot using matlab in a robot operating system environment," *Symmetry*, vol. 13, no. 6, 2021. [Online]. Available: https://www.mdpi.com/2073-8994/13/6/969

[5] Y. Li, S. Ge, S. Dai, L. Zhao, X. Yan, Y. Zheng, and Y. Shi, "Kinematic modeling of a combined system of multiple mecanum-wheeled robots with velocity compensation," *Sensors*, vol. 20, no. 1, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/75

[6] B. Chu, "Position compensation algorithm for omnidirectional mobile robots and its experimental evaluation," *International Journal of Precision Engineering and Manufacturing*, vol. 18, no. 12, pp. 1755–1762, 2017.

[7] P. Tian, Y. N. Zhang, J. Zhang, N. M. Yan, and W. Zeng, "Research on simulation of motion compensation for 8× 8 omnidirectional platform based on back propagation network," in *Applied Mechanics and Materials*, vol. 299. Trans Tech Publ, 2013, pp. 44–47.

[8] J. Zuo, S. Liu, H. Xia, and Y. Qiao, "Multi-phase fingerprint map based on interpolation for indoor localization using ibeacons," *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3351–3359, 2018.

[9] V. J. Expósito Jiménez, C. Schwarzl, and H. Martin, "Evaluation of an indoor localization system for a mobile robot," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, pp. 1–5.

[10] C. Budaciu, N. Botezatu, M. Kloetzer, and A. Burlacu, "On the evaluation of the crazyflie modular quadcopter system," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 1189–1195.

[11] Marvelmind. (2007) Marvelmind robotics precise (±2cm) indoor positioning and navigation. [Online]. Available: https://https://https://marvelmind.com//

[12] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS*. O'Reilly Media, 12 2015.

[13] S. C. . S. F. Conservancy. (2021) Git - free and open source distributed version control system - download page. [Online]. Available: https://git-scm.com/downloads

[14] M. Robotics. (2021) Ros marvelmind package git repository page. [Online]. Available: https://bitbucket.org/marvelmind_robotics/ros_marvelmind_package

[15] Mathworks. (2021) Matlab - ros system requirements. [Online]. Available: https://www.mathworks.com/help/ros/gs/ros-system-requirements.html

Real-time Communication between Automation Studio and PLC based on OPC Technology for control 3-DoF robot

Constantin-Cătălin Dosoftei, Alexandru Lupu, Lucian Mastacan
Dept. of Automatic Control and Applied Informatics,
"Gheorghe Asachi" Technical University of Iasi
Iasi, Romania
{cdosoftei, lupu.alexandru, lmastacan} @ac.tuiasi.ro

***Abstract*—The approach from this work is multidisciplinary, similar with challenges from industrial word which started to implement the new industrial revolution in more and more areas. This paper presents a method of implementing a remote real-time communication, based on Open Platform Communication (OPC) on the Ethernet, for the communication between Automation Studio software solution and a Programmable Logic Computer (PLC) which control a 3-DoF pneumatic robot with a handle gripper. The model of robot is implemented in Automation Studio and this paper uses the benefits of OPC Server in order to connect the simulated system with a PLC. A real PLC is used to implement the supervisory control of the simulated plant. Relation between the physical and the virtual world is an important component of Industry 4.0, represent cyber-physical system.**

Keywords—Industrial ecquipments, Automation Studio, OPC server, robot, virtual model

I. Introduction

Simulation can be used before starting the physical implementation of a control system to evaluate the effect of real-world actions without making changes to the real system. In industry field, the information obtained after performing simulations, minimizes the material risks, reduces costs, time and energy for the implementation of a control system while in the academic environment, simulation can be used for the easier understanding of processes.

Many simulation software exist and almost all of them are based on high-level languages that can range from functional to graphic models. Automation Studio (AS) is such a simulation and project documentation software package, which offers intuitive design, animation, simulation and system analysis functionalities in a versatile and user-friendly environment. This solution covers a wide array of technologies: control systems, pneumatics, hydraulics, mechanical systems, electronic and electrical, fluid power design being intended to be used by engineers in a wide variety of related fields [1], [2].

The challenge for the presented work was to make a connection between AS software and an industrial equipment represented by a robot (cylindrical configuration arm), in order to analyze the behavior and the difference between the movements of the virtual 3D realistic model and the real process. The virtual model will represent the digital twin of robot and in actual paper the model will be validated through an approach of the type hardware in the loop. Using digital twin along with the ability to collect and store data from sensors of robot can be realised various analyzes are being used to optimize the operation and maintenance of physical robot components.

This work was supported by a research grant of the TUIASI, project number GnaC2018_190

The virtual component model of the robot was drawn using the design leading software CATIA saved with extension .igs, imported in AS with the purpose of creating a cinematic animation. Creating of the model follows the way of low level modelling of a cyber physical system it is an interdisciplinary challenge which is suitable for a higher technical level.

A robotic system is a complex system consisting of multiple parts: manipulator, end effector, actuators, sensors and controller, all of them being industrial equipments. The robot used in the paper is 3DoF (Degrees of Freedom) type cylindrical with two prismatic movements, realized with pneumatic cylinders and one revolute movement obtained with a limited angle actuator - (R2P) [3]. This type of robot is commonly used in industry: assembly operations, handling machine tools, handling at die-casting machines, spot welding operations or painting operations.

The AS has predefined communications with two types of PLC [2], but in this work the controller of robot is a PLC which is not included in standard library of AS. This issue represented a new challenge for this paper through using an external OPC server which makes possible the data exchange between simulated and animated process developed in AS and control device of robot realized with a PLC XBC-DR30SU series. This bridge, realized with OPC server, gives the compatibility and interoperability of the system.

The 3DoF robot was equipped with a different type of proximity sensor to provide position feedback, because in robotic application feedback is crucial. Usually for pneumatic cylinder with magnetic piston magnetic proximity sensors are used to detect magnet field integrated in piston [4]. Where space does not allow due the robot chassis, or the piston is not magnetic, inductive sensors are used. Additional proportional pressure transducers were used to measure the pressure in both chambers of cylinder (positive chamber = piston side and negative chamber = rod side). From variation of pressure we proof in the paper that for a double acting pneumatic cylinder is possible to have the position of piston. All sensors measurements are inputs for PLC and through OPC server are used for monitoring and analysis of real-time data in AS. Acting the robot is done through control of solenoid directional valves by outputs of PLC. These valves open, close or divert the flow of compressed air, so that the actuators for linear, angle and gripper movements to assume different positions with respect to past positions [4].

II. HARDWARE SETUP

The robot considered in this application has 3DoF represented by: up and down, forward and backward, left and right rotation at an angle 90º. The opening and closing of the robot's gripper is not considered as the independent robot axis, because doesn't contribute to either the position or the orientation of the robots working space.

The model is made using a structure with pneumatic cylinders. This assemble allows the robot arm to make a rotation and two linear translations. End effector it is a mechanical gripper, used to catch and hold an object and place it at a desired location [3].

A. *Pneumatic actuators*

The term used for a mechanism that drives a robot arm is actuator. These actuators may be electric motors, pneumatic or hydraulic cylinders. The pneumatic actuators are generally suited to driving prismatic joints since they produce linear motion directly. They are often referred to as linear actuators which have a head-tail movement with a fast response and low-cost. The rotary moving (from first joint) is obtained with a pneumatic rotary actuator with double acting. Pneumatic actuation is commonly used in industrial and commercial applications for its low cost, compact size, high power-to-weight ratio, reliability and low maintenance [5].

Translations are realized with two cylinders: the up/down (U/D) movement is made by a magnetic double action cylinder 41M2P080A0200 with diameter Ø80mm and the stroke 200 mm. At a pressure equal with 4 bar this cylinder generate a force for thrust side equal with 1,77kN and for traction side the force is 1,6 kN. The forward/backward (F/B) is a magnetic double ended piston rod (this type building is in link with equilibrum of F/B moving) with double action cylinder 60M6L063A0400 with diameter Ø 63 mm and stroke 400 mm. Both cylinders have adjustable cushioning system which is shock absorbers. The piston has a magnetic ring in order to determine the position using magnetic sensors. The magnetic and inductiv sensors are used for the position feedback at end of strokes. For U/D cylinder in this research were replaced proximity sensors with two pressure transducers PT010RG02 0- 10 bar/4-20 mA.

The rotation of the robot is realized with a 90 degrees rotary actuator (R/L) model ARP 055. Basically, the linear move of cylinder is transform in a circular move using a set of gears. In order does not have big friction in the revolute joint, the robot chassis is build on thrust ball bearing.

B. *Pneumatic-mechanic gripper*

The end-of-arm-tooling for application which does not require force control in order to be safely picked, the usual solution is to use pneumatic - mechanic gripper [4]. The 180 degrees angular gripper (G) CGSN-25 can provide high precision and high grip force. Also, proximity sensor is added, to determine if the gripper is close or open.

C. *Solenoid valves*

The pneumatic circuit requests directional valves to control the movement of each cylinder. All valves are mounted on manifold with a comon port to inlet pressure and two ports for exhaust pressure. In implementation are used solenoid valves with electric command, mechanical spring return and external servo pilot. For rotary actuator is used a bistable valve 5-Ports, 3-Positions (5/3) with central position closed model 338D-015-02 with coils voltage 24 V DC – receives the command directly from PLC output. For other cylinders and gripper valves are used model 358-015, 5/2 monostable with one coil also commanded directly from PLC output [2].

III. MODELING AND 3D IMPLEMENTATION OF ROBOT IN AS

A. *Modelling robot*

The robot is implemented in software environment splited on components which give the movement. Individual components which compound the project use standard elements predifined in Pneumatic Library or non standard elements designed and developed by user. For both type of components technical characteristics and operating conditions can be configured with the purpose of realistically reproduce the system behavior [8]. In this work only the robot modeling and simulation was taken in consideration (from valves manifold to robot effector) without the air network and the compressor.

The modelling of robot's components in AS can be either mathematical or graphical. In this work was done graphical and the model of robot which can be controlled in cascading is represented in fig. 1. Once each component has been sized were established overall requirements of the robot by running simulation and operating sequences. During simulation, were adjusted parameters like loads, dimensions of air tubes and characteristics of fluid (compressed air) like pressure control, flow, leakage and kinematic and dynamic variables like position, speed, acceleration, force and torque. Some of the parameters were measured from real robot and other parameters (friction, lekeage) were estimated in terms of behavior as close to the real dynamic of robot. In the result section will be presented a comparison between behavior of realtime simulation and 3DoF pneumatic robot.

B. *3D Implementation robot*

The whole 3D model is designed using CATIA and simulated in virtual reality environment from Automation Studio. Each part of the robot model is created in CATIA and saved in this workspace with extension *.igs*, then the parts are assembled in AS 3D editor, where is possible to visualize, simulate and animate them simultaneously with the technologies that drive the real robot, as detailed in [6].

In Fig. 2 is presented pear real robot and virtual robot implemented in AS. Advantages of using virtual model for industrial process are in link with remote accessibility for learning, for testing with specific target of shorten time commissioning, predictive maintenance - concepts used in paradigm of Industry 4.0.

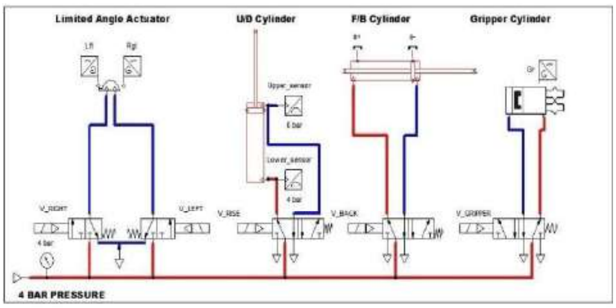


Fig. 1. Pneumatic circuit of 3DoF robot

Fig. 2. Real - Virtual model robot tandem

IV. ROBOT CONTROL USING PLC

The complexity of robot control is in directly link with the binomial correspondence between numbers of DoF and the rules which activate the moving. Taking into consideration the last trend in industrial control approach [7]-[8], decisional component developed in the rule-base in computer is transferred to PLC either as a master device or directly control, like in this research.

The PLC has 18 points of DC inputs and 12 points relay outputs, two auxiliary modules AH02A (1-channel analog input (I/U)/ 1-channel analog output (I/U)) and a communication module EMTA (Fast Ethernet (100Mbps), 1Ch) [2]. The inputs (digital inputs – proximity sensors and analog inputs – pressure transducers) and outputs (solenoid valves actings) used in implementation are presented in the next table.

TABLE I. ASSIGNATION OF PLC INPUTS/OUTPUTS

Input/ Output	PLC Register	Variable	Data type
IN	P00000	START_STOP	BIT
IN	P00001	a_minus	BIT
IN	P00002	A_PLUS	BIT
IN	U09.01.4	_09_AD0_DATA	WORD
IN	U0A.01.4	_0A_AD0_DATA	WORD
IN	P00005	LEFT	BIT
IN	P00006	RIGHT	BIT
IN	P00007	GRIPPER	BIT
OUT	P00040	UP	BIT
OUT	P00041	LEFT	BIT
OUT	P00042	RIGHT	BIT
OUT	P00043	BACKWARD	BIT
OUT	P00044	OPEN	BIT

The PLC's program run the sequence represented in the graph from Fig. 3.

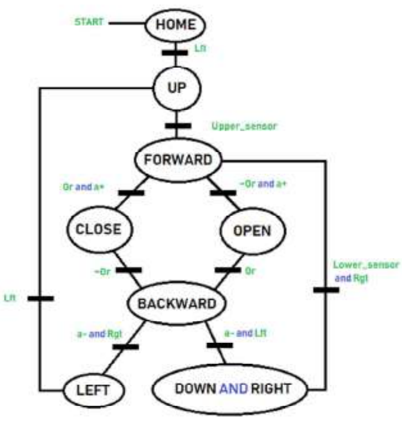


Fig. 3. Transition graph

After pushing the "Start" button the robot returns to the homing position. So we consider the initial state as the homing position defined by the configuration with the robot down, backward, turned to left and the gripper closed.

The cascading movement of robot's actuators implemented according with transition graph is next figure:

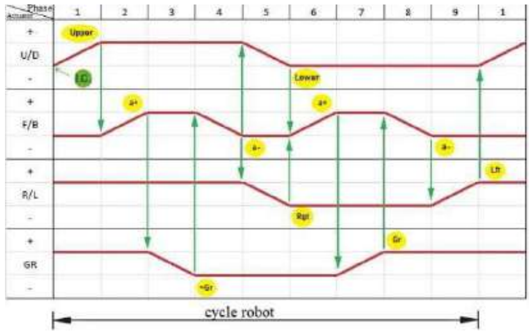


Fig. 4. Graphical representation of movement implemented in PLC

The completely cycle operates in nine phases. In first phase to obtain the stroke positive of U/D cylinder is necessary to initiate cycles – start from home position (I.C.) and output signal from proximity sensor Lft which confirm that robot is turn completely in left and pressure in positive chambre of cylinder is almost 0 and pressure in negative chambre is maximum. The program in ladder diagram start with calibration of signals from analog inputs – signals from pressure transducers. In Fig. 5 appear a capture of PLC program.

V. SYSTEM ARCHITECTURE

The PLC used in controlling of robot is not included in standard PLC Library from AS and to make the link was necessary to introduce an industrial server KEPServerEX 6.3 and to configure in AS an OPC client. The simplified architecture of the real time communication between AS and PLC based on OPC technology for control robot is presented in Fig. 6. Through this gate/tunnel created by OPC server the real-time model of robot implemented in AS are in link both with sensors from real robot (which are PLC inputs) and with activating outputs from PLC which energize virtual solenoid valves from the AS model [2].

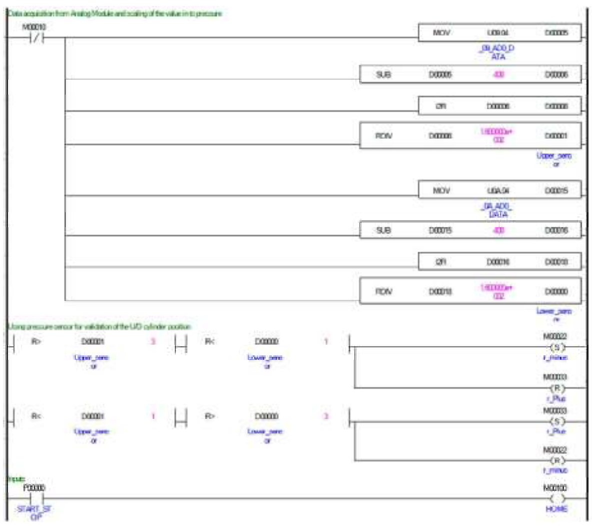


Fig. 5. Capture from LD control robot program

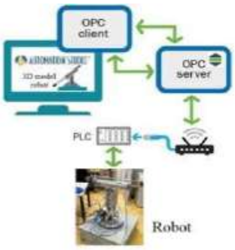


Fig. 6. System architecture

KEPServerEX provides the ability to establish links between data values in different data sources, allowing Machine-to-Machine (M2M) communications as close to the device as possible [9]. In this project the communication with PLC is through Ethernet comumunication module – EMTA. Is creating a channel where is added the device connected through IP. All inputs and outputs are defined firstly in server and after this in client from AS. AS has the facility to connect to OPC Server and it is paired OPC tags and valves solenoid from the virtual model – Fig. 7.

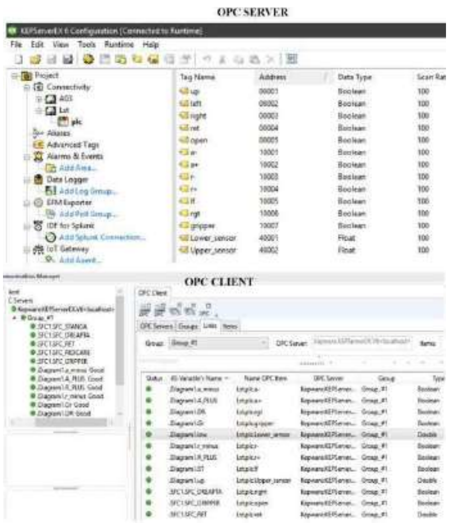


Fig. 7. Tags from OPC Server and OPC client

VI. RESULTS

The target for the work was to validate the virtual model of robot developed in AS by the tools commonly used in the industrial environment. Using the facility of OPC server which syncronize the 3D model and real process in this work the validation is through comparison pressure stabilization in positive (blue curve)/ negative(green curve) chambers of cylinders. In Fig. 8 is presented the differential pressure approach in the cylinder which execute the U/D movement of robot, the main cylinders which carry a big part of chassis robot and in vertical position.

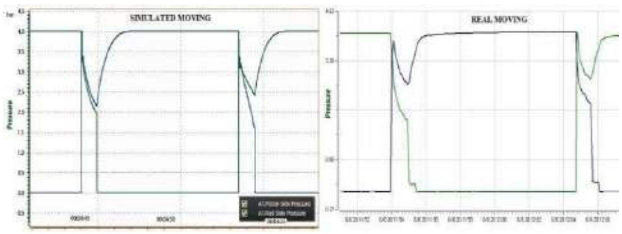


Fig. 8. Simulate and real pressure evolution in U/D robot cylinder

After the experiments which stay on base calibration of parameters which are very hard to measure was started the experiment to record the pressure profiles, Fig. 9, when the robot execute completely cycles descripted in transition graph from Fig. 3. A comparative behavior between the 3D model and the real robot is illustrated in a video (https://youtu.be/LQY9iV6VumY)

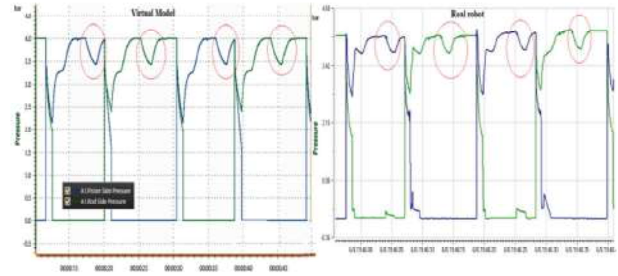


Fig. 9. Simulated process vs Real process pressure

In the graph have been highliting with circles perturbation of stationary regim which are in link with variation of pressure, for a few milisecond in U/D cylinder, when appear consumption of air flow for moving F/B robot cylinder (coupling fenomena). Only between these cylinders there is a coupling because the cylinders volumes are relatively close in values. Small differences can be justified by measurement noises which appear in real process.

VII. CONCLUSION

In this paper is validated a 3D model of a pneumatic robot with real-time comunication. Using this approach with link between real process and AS software through OPC server offers advantages in industrial word for real-time monitoring of productivity through collecting and storing the data in a cloud system. The dynamic behaviour is similar between virtual model and real, small differences can be justified by measurement noises of analog sensors and other constructive restrictions from real word.

REFERENCES

[1] L. Nohacova, K. Nohac, "Possibilities of Computer Simulation in Power Engineering and Environmental Engineering. In: Sobh T. (eds) Innovations and Advances in Computer Sciences and Engineering. Springer, Dordrecht, ISBN978-90-481-3657-5, 2010.

[2] Internet adresses: www.automationstudio.com, www.tech-con.ro, www.camozzi.com, www.lsis.com, www.kepware.com

[3] M.Spong, S. Hutchinson, M. Vidyasagar, Robot Modeling and Control, 2nd ed., John Wiley & Sons, Inc., 2004.

[4] A. Parr, Hydraulics and Pneumatics. A Technician's and Engineer's Guide, 3rd ed., Butterworth-Heinemann, 2011.

[5] I. L. Krivts, G. V. Krejnin, Pneumatic Actuating Systems for Automatic Equipment. Structure and Design, 1st ed., Taylor Francis, ISBN9780429125621, 2006.

[6] C.C. Dosoftei, A. Lupu, C. Pascal, A new approach to create a realistic virtual model of a cylindrical robot using Automation Studio, in press.

[7] M. Patil, "Robot Manipulator Control Using PLC with Position Based and Image Based Algorithm", International Journal of Swarm Intelligence and Evolutionary Computation, ISSN: 2090-4908 Volume 6, Issue 1, 2017.

[8] C. Pascal, L. Raveica, D. Panescu, "Robotized application based on deep learning and Internet of Things" 22th ICSTCC, IEEE, IAN 18274236, 2018.

[9] T. Wanyama, I. Singh, "A training demonstration for experiential learning in OPC based process automation data access" Proceedings of the Canadian Engineering Education Association (CEEA), 2013.

sensors

Article

Kinematics Calibration and Validation Approach Using Indoor Positioning System for an Omnidirectional Mobile Robot

Alexandru-Tudor Popovici [1,*], Constantin-Catalin Dosoftei [2] and Cristina Budaciu [2]

[1] Department of Computer Engineering, "Gheorghe Asachi" Technical University of Iasi, 700050 Iasi, Romania
[2] Department of Automatic Control and Applied Informatics, "Gheorghe Asachi" Technical University of Iasi, 700050 Iasi, Romania
* Correspondence: alexandru-tudor.popovici@academic.tuiasi.ro

Abstract: Monitoring and tracking issues related to autonomous mobile robots are currently intensively debated in order to ensure a more fluent functionality in supply chain management. The interest arises from both theoretical and practical concerns about providing accurate information about the current and past position of systems involved in the logistics chain, based on specialized sensors and Global Positioning System (GPS). The localization demands are more challenging as the need to monitor the autonomous robot's ongoing activities is more stringent indoors and benefit from accurate motion response, which requires calibration. This practical research study proposes an extended calibration approach for improving Omnidirectional Mobile Robot (OMR) motion response in the context of mechanical build imperfections (misalignment). A precise indoor positioning system is required to obtain accurate data for calculating the calibration parameters and validating the implementation response. An ultrasound-based commercial solution was considered for tracking the OMR, but the practical observed errors of the readily available position solutions requires special processing of the raw acquired measurements. The approach uses a multilateration technique based on the point-to-point distances measured between the mobile ultrasound beacon and a current subset of fixed (reference) beacons, in order to obtain an improved position estimation characterized by a confidence coefficient. Therefore, the proposed method managed to reduce the motion error by up to seven-times. Reference trajectories were generated, and robot motion response accuracy was evaluated using a Robot Operating System (ROS) node developed in Matlab-Simulink that was wireless interconnected with the other ROS nodes hosted on the robot navigation controller.

Keywords: OMR; indoor positioning system; accurate localization; calibration; validation

check for updates

Citation: Popovici, A.-T.; Dosoftei, C.-C.; Budaciu, C. Kinematics Calibration and Validation Approach Using Indoor Positioning System for an Omnidirectional Mobile Robot. *Sensors* **2022**, *22*, 8590. https://doi.org/10.3390/s22228590

Academic Editor: Jordi Palacín Roca

Received: 3 October 2022
Accepted: 3 November 2022
Published: 8 November 2022

1. Introduction

The high interest in the analysis of the performances of Omnidirectional Mobile Robot (OMR) navigation platforms is increasing in the scientific community [1–3], as well as in the industry field for different types of applications, starting from monitoring and mapping of the area of interest towards the transportation, intelligent manufacturing [4], and logistic activities [5,6]. In this context, the OMR vehicle performs movement in any direction under any orientation; therefore, it has great advantages over conventional platforms (i.e., car-like Ackermann steering or differential drive system) in warehouse management, where complex trajectory planning associated with task assignment is a demanding requirement.

Currently, the main attention in the advanced mobile robots research field is paid towards transitioning from automated guided vehicles to autonomous mobile robots. This technological challenge is sustained by developing complex sensors and computational processing power, which offer new navigation capabilities in a dynamic environment with predefined or variable constraints [2,3,7].

The prediction is that over four million logistical robots will be developed and placed in approximately 50 K warehouses by 2025 [8]. The favorable factor for this assumption that will have a big impact on performance in the operational logistic domain is represented by the migration of e-commerce to digital commerce. Many parallel research, both from various academic communities and industrial companies, is closely related to the development of a handling and transport solution in a complex logistic environment, the testing and validation of the experimental OMR localization being an important step in the further development of algorithms [9]. The increased interest in omnidirectional systems is primarily due to their maneuverability in logistics and assembly applications [4,6]. Robot pose estimation is based on odometry, which is defined as a simple positioning method based on the wheel velocity measurements and is usually used in real-time experiments.

Path performance evaluation based on odometry is inconclusive due to the mechanical shortcomings of the experimental OMR, even if the control action is well defined. Furthermore, the main concern related to the errors that appear in the localization of the robot is also justified by the fact that the OMR is intended to work in a dynamic warehouse environment. Even if the improvement of odometry by proper calibration reduces the position errors, the accurate knowledge of the current location can be ensured by an Indoor Positioning System (IPS).

In this research study, the data acquisition from the IPS was exploited in order to monitor the motion response of the OMR and calculate the calibration parameters. Experiments involving typical motion paths for the OMR were performed, so that the longitudinal, lateral, rotational, and composed motions were studied in order to establish the calibration requirements for the kinematic transformations used for commanding the motion of the OMR and monitoring its actual execution.

Tracking performance evaluation scenarios for composed motion reference trajectories having Lissajous curve shapes are comparatively analyzed both from the IPS and estimated odometry position [10]. In the current research, a plurality of sets of data was obtained from multiple experimental scenarios, which were carried out starting from simple orthogonal movements towards to complex trajectories.

The architectural concepts together with the basic description of an omnidirectional embedded real-time robotics system was developed by the authors in the current research project, ROSY-LOGISTIC, and the main results were published in previous articles [11–13]. With the development of many open-source technologies, an OMR can be used on almost any scenario due to the availability of software libraries and tools, among which is the Robot Operating System (ROS).

This research tries to capture as comprehensively as possible, in a first phase, the necessary steps for the IPS ROS node integration in the Matlab environment, since these steps are not completely described in the technical literature; thus, for anyone, this can be a time-consuming step. Further in Section 2, the motivation of the paper is more pronounced due to the lack of position accuracy in the experimental path recorded. Section 3 attempts to develop a calibration approach starting from measurements of the motion and odometry errors. Section 4 focuses on the experimental scenarios together with the open-loop motion tracking and performance evaluation. The last section concludes the research and discusses the perspective for the future work.

1.1. Related Work on Trajectory Validation for OMRs in Logistic Areas

The accurate knowledge of the current localization and orientation of the mobile robot can benefit largely from a well-calibrated odometry. In outdoor applications, Global Positioning System (GPS) technology offers good positioning accuracy and is used in many automated driving applications. Unfortunately, for indoor localization, GPS technology cannot offer similar reliability because the signals of the satellites lose much strength when penetrating a building.

Low-cost technologies such as WiFi, ZigBee, and Bluetooth Low-Energy (BLE) are radio-frequency-based systems and are widely used in in mobile robots for indoor localization [14,15]. Although they are very popular due to the availability of the hardware, the accuracy of static measurements is in the order of 1 to 4 m [16].

Ultra-Wideband (UWB) has gained interest in indoor positioning of robots, the system relying on the signal travel time for the distance between the mobile robot and static anchors. The mobile robot location is estimated by exploiting approaches such as multilateration and trilateration [17]. In the paper [14], the authors benchmarked the accuracy of different types of indoor positioning systems including the Marvelmind robot. The experimental results demonstrated that, in larger spaces, there are situations where the Marvelmind system cannot not perform measurements because of corrupted packets identified using CRC methods, even if there are not any apparent sources of interference present. This phenomenon became more pronounced for higher data traffic. Obviously, in recent years, considerable progress has been made in terms of positioning systems for mobile robots through the use of the latest sensors and signal processing techniques.

There are several approaches for IPSs with their own advantages and limitations [18]. Usually, a data fusion algorithm is used in the localization system to combine the pose estimates from the two different sources. Several methods exist, such as inertial, visual, laser, LiDAR, or wheel odometry, and any of the methods can be applied in a multisensor fusion algorithm, e.g., visual–inertial odometry [15,19]. Though multisensor fusion approaches are usually used, there is a real benefit in increasing the confidence in the odometry.

The mobile robots' cost increases significantly with the addition of advanced sensors, but odometry and calibration methods can definitely mitigate the positioning error.

1.2. Contributions of the Paper

This work comes as a natural continuation of the previous work [12] of the ongoing research project, which aims to develop transport solutions in complex logistic environments using a fleet of autonomous omnidirectional mobile robots coordinated by a warehouse management system. In our previous work [12,13], we provided baseline experimental results starting from orthogonal movements and continuing with more complex trajectories.

The main contribution of this paper concerns the OMR localization accuracy analysis and proposing a new offline method for the OMR kinematics experimental calibration. In this regard, the effectiveness of the method was verified by comparing the performances between the reference trajectory and the estimated position.

The position provided by Marvelmind was compared to a separately implemented position calculation based on raw point-to-point distances reported by the IPS to evaluate its reliability. The initial OMR trajectory tracking was evaluated, and in addition to the translation velocity correction coefficients [20], the need for translation–rotation cross-talk compensation coefficients was established.

2. The OMR's Hardware and Software Architecture

The OMR consists of a chassis that contains the set of mechanical elements including the propulsion system, made up of four motor–planetary gearbox assemblies, coupled to the chassis through eight shock-absorbing suspensions, which ensure contact with the ground, at any time, for the four Mecanum wheels, which have a diameter of 6″, as shown in Figure 1. The movement is facilitated by the electrical energy provided with the Li ion batteries, having a nominal DC voltage of 22.2 V. The OMR acts in a working environment to perform different tasks, according to the software component, represented by the implemented control algorithms, which take into account the information received from the perception system. The complexity of the perception system is closely related to the specifics of the operations performed by the robotic platform. The main components with the characteristics of the perception system [21] are presented in Table 1. Last but not least is the control system of the robot.

The driving structure was implemented hierarchically, with a top-down approach, with two controllers: vehicle controller (executive-level) and navigation controller (high-level). The vehicle controller was implemented with an STM32F103RC micro-controller, while the navigation controller was implemented with an NVIDIA Jetson Nano B02 embedded system. The two control systems are directly connected to various components of the robot, as represented in Figure 2, and exchange information through specific control instructions, receiving through the communication protocol both information related to work possibilities and information on the activity and the current state; information analyzed at this decision level allows the robot to further establish the action strategy.

Table 1. Components of the perception system.

Component Model	Specifications
RP LIDAR A2M8 360	2D LiDAR, resolution: 0.5 mm–1.5 m at a maximum range: 12 m
ASTRA PRO Depth	3D camera stereo, distance: 0.6 m–8 m 1280 × 720 @30 fps
MPU6050	IMU 3-axis accelerometer and gyroscope module
EI 500P/R	Quadrature Optical Encoder 500 ppr
HW v4.9-IMU-NIA	MARVELMIND "GPS" indoor system with beacons

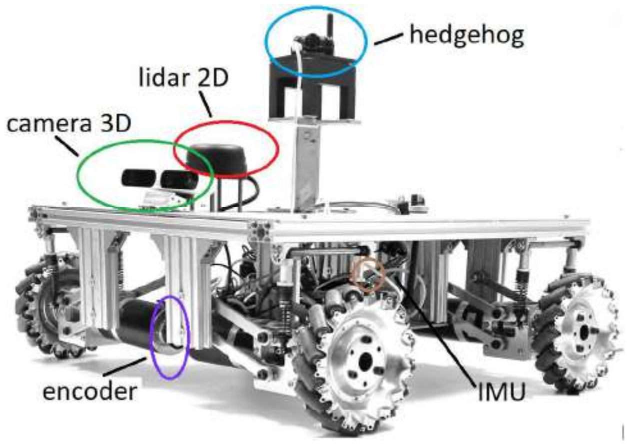


Figure 1. The OMR used in the experiments.

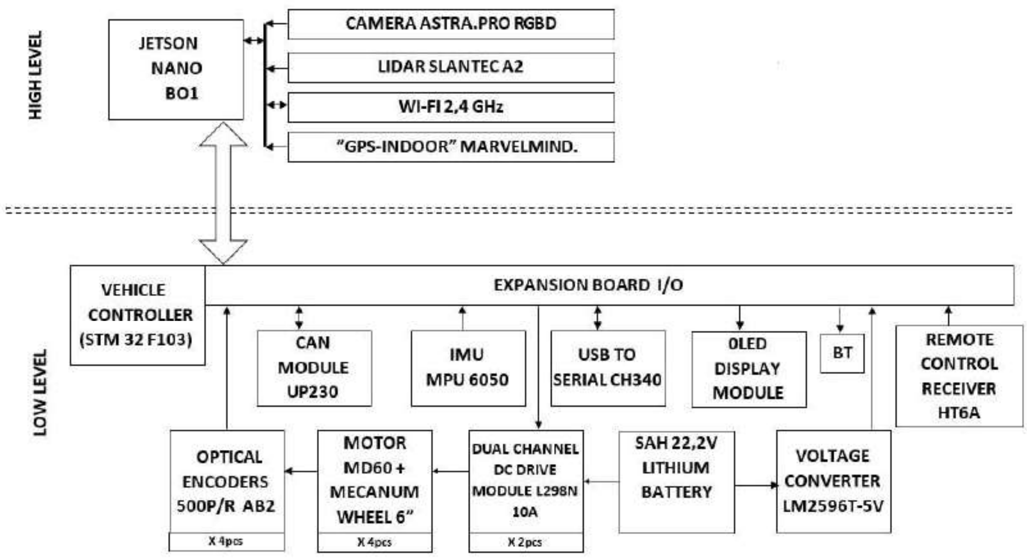


Figure 2. Layered hardware architecture of the OMR.

Given the complexity of the software needed in a dynamic environment, a certain degree of computational resources is required. The path-planning algorithms need to be implemented in order to be executed completely on the OMR platform, without the help of external or remote control.

On the low-level controller runs a customized implementation of FreeRTOS. The firmware includes the inverse and direct kinematics models with all specific OMR parameters. As was demonstrated and detailed in [12], the matrix representation of the inverse kinematics is depicted in Equation (1):

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = J \begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} \tag{1}$$

where ($\omega_i, i = \overline{1,4}$) represents the angular speed of each wheel, v_x/v_y are the instantaneous longitudinal/lateral velocities component of the OMR, Ω is the rotational speed, and J is the inverse kinematic Jacobian matrix of the OMR—expressed in relation to the notational conventions of the OMR elements:

$$J = \frac{1}{R}\begin{bmatrix} 1 & 1 & -(l_x + l_y) \\ 1 & -1 & -(l_x + l_y) \\ 1 & 1 & (l_x + l_y) \\ 1 & -1 & (l_x + l_y) \end{bmatrix} \tag{2}$$

All variables and their numerical values for the experimental platform from Equations (1)–(3) below are highlighted in Figure 3.

The way to determine the lateral/longitudinal and rotation velocities of the OMR starting from each wheel's velocity is known as forwarding kinematics and is used in odometry calculation—Equation (3):

$$\begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} = \frac{R}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \frac{-1}{l_x+l_y} & \frac{-1}{l_x+l_y} & \frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{3}$$

In an Ubuntu environment with popular programming languages and libraries such as C++, Python, OpenGL, and ROS, the navigation level of the OMR developed on the Jetson Nano, which comes with a Quad-core ARM A57 @ 1.43 GHz and 128-core Maxwell GPU, providing enough resources to cater to the computation for mapping and motion planning.

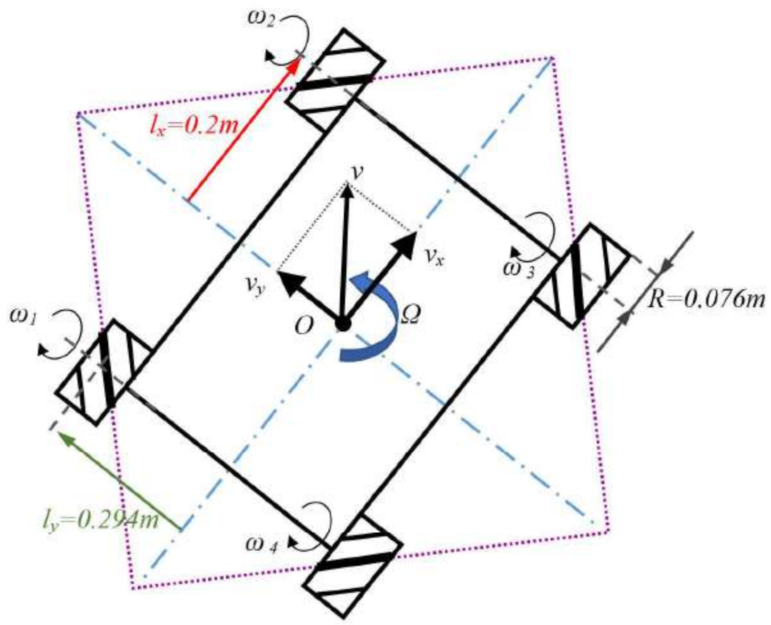


Figure 3. The robot coordinate system and dimensions.

2.1. The Indoor Positioning System—Marvelmind Ultrasound Beacons

The IPS used with the OMR platform was developed by Marvelmind Robotics (Starter Set HW v4.9-IMU-NIA), with a promised accuracy of 20 mm [22], and can be configured to obtain 3D or 2D position solutions. The working principle is based on a network composed of fixed ultrasonic beacons and a mobile beacon, called the hedgehog, installed on the target, which must be localized, in our case, on the OMR. All the beacons are linked by radio interfaces operating in the license-free Industrial, Scientific, and Medical (ISM) band. The mobile beacon on the OMR integrates an Inertial Measurement Unit (IMU) having an accelerometer, a gyroscope, and compass module, which can be used for sensor fusion. The ultrasonic sensor network is completed by a modem having the role of the central controller for the system, which communicates with all the beacons through radio, in the case of the equipment using the ISM 433 MHz band, specific to the European region. Messages with localization information can be received from the hedgehog, but also from the modem, by several communication interfaces (USB Virtual COM Port, I2C, SPI, serial TTL, and others).

There are two operation modes possible for the system, which are called the Inverse Architecture (IA) and Non-Inverse Architecture (NIA). In the IA, the stationary beacons emit ultrasonic signals, while multiple hedgehogs can receive them, the mode being more useful for small areas requiring a minimal number of fixed beacons in order to avoid the reduction of the localization rate. In the NIA, the mode used in the current research, the hedgehogs emit the ultrasonic signals and the stationary beacons receive the propagated wave, and if more hedgehogs are to be monitored in the same area, either Time Division Multiple Access (TDMA) or Multi-Frequency (MF) ultrasonic signals using Frequency Division Multiple Access (FDMA) can be configured for quasi-simultaneous and, respectively, simultaneous tracking. Using TDMA, multiple targets take turns in being localized. Using FDMA, the targets emit simultaneously, but on different ultrasonic frequencies. TDMA implies a reduction of the localization rate, while for FDMA, there is a limited number of frequencies that can be effectively identified using digital filters [23,24] implemented on embedded systems such as the ultrasonic beacons and the coordinating modem.

The location of the OMR is calculated using a proprietary, undisclosed, trilateration algorithm based on the propagation delay, also named Time of Flight (ToF), of the acoustic signal between the hedgehog and up to four nearby stationary beacons of the reference network. The distances between beacons is recommended to be up to 30 m, achieving with just four stationary beacons a coverage area of up to 1000 m^2. Larger or more complex areas can be covered using more stationary beacons in the reference network.

The performance of the localization system depends on many aspects, and a proper configuration of the fixed beacon network is essential for achieving the promised accuracy. In real-world scenarios, it can be difficult to achieve the ideal conditions needed for proper operation. Based on the acquired experience, the localization rate decreases on so-called submaps (subareas covered by up to four fixed beacons), where the maximum distance between the beacons is larger. In a 2D configuration for a maximum distance of 5 m between any two beacons, localization rates of around and more than 25 Hz can be achieved. On the other had, 2D localization precision in systems using multilateration is subject to the Horizontal Dilution of Precision (HDoP) [25], which depends on the relative position of the target regarding the available reference beacons. For this reason, even in spaces where there are no obvious issues (propagation path occlusion, multiple/indirect propagation, interference, or other), the accuracy will vary, generally becoming worse when the target is closer to the periphery of the submap in which it is being tracked.

The experimental OMR platform was equipped with the Marvelmind IPS in 2021, and since, then there have been roughly monthly updates of the firmware and software on the producer website, which keeps improving the solution. The latest version used in this study was 7.202 from the beginning of September 2022.

2.2. IPS ROS Node for Matlab-Simulink Integration

The Marvelmind company offers as an open-source component, the *marvelmind_nav* Robot Operating System (ROS) package, in order to facilitate the usage of their IPS solutions in industrial and robotics applications. Since it is not yet a standard ROS package, it needs to be manually installed to be used on the OMR. A short summary of the installation steps [26] is presented below:

- Open a command line on the target embedded computer;
- Change the directory to the sources folder (*src*) of the used ROS workspace;
- Make a new directory named *marvelmind_nav* in the *src* folder;
- Download manually the latest ROS package from the Marvelmind repository [22] or use *git clone*;
- Execute *catkin_make -only-pkg-with-depts marvelmind_nav* to build the package;
- Run the node with the command *rosrun marvelmind_nav hedge_rcv_bin /dev/ttyACMx*; in the previous command, the *x* in *ttyACMx* must be replaced with the device number (usually 1) of the USB virtual serial port that appears in the */dev/* directory when the hedgehog is connected using the USB cable to the embedded computer.

The ROS node named (*hedge_rcv_bin*) publishes the position data through the topics: *hedge_pos*, *hedge_pos_ang* and *hedge_pos_a*. Marvelmind decided to use custom message types for their published topics, most likely because of the specificity of the IPS application. As a result, until official standard support is offered from ROS and Matlab-Simulink, the following steps will be required for Matlab-Simulink version R2021a in a Windows 10 environment in order to successfully configure the ROS Subscriber blocks for receiving data from the Marvelmind ROS node:

- Check if the message type is available by running in the Matlab Console (MC): *rosmsg list*; if no message name starting with *marvel* appears, than the next steps need to be executed; otherwise, it means that the Simulink model can already subscribe to the position information topics;
- Install CMake 3.15.5 or newer [27];
- Install Microsoft Visual Studio 2017 (VS17), especially the C/C++ development tools and the CMake support (note that, currently, only VS17 is supported for the Matlab ROS toolbox [28]);
- Install Python 2.7 [29] (note that, currently, the Matlab ROS toolbox does not support any other Python version [30]);
- Configure the Python version by executing in the MC (recommended immediately after restarting Matlab): *pyenv('Version','2.7')*;
- Add to the Matlab path the binary installation folder of CMake, by executing in the MC: *addpath('C:\Program Files\CMake\bin')*;
- Set up the Matlab compiler for building the *mex* file type shared libraries with VS17 by executing in the MC (the path depends on the actual MATLAB installation folder): *mex -setup:'C:\Program Files\MATLAB\R2021a\win64\mexopts\ msvcpp2017.xml' C++*;
- Copy in Matlab's current path the folder *msg* from the root of the ROS Marvelmind package [31] that contains *.msg* files;
- Rename the locally copied folder to *marvelmind_ros_messages*;
- Build the needed Matlab *.mex* files for supporting Marvelmind ROS messages by executing in the MC: *rosgenmsg('./marvelmind_ros_messages')*;
- Include the folder containing the support files for custom Marvelmind ROS messages in the Matlab path by executing in the MC: *addpath('./marvelmind_ros_messages')*;
- Save the Matlab path for future restarts by executing in the MC: *savepath*;
- Clear the Matlab workspace classes by executing in the MC: *clear classes*;
- Refresh the Matlab toolbox cache in order to load the new message types by executing in the MC: *rehash toolboxcache*;
- Run the first step again to check that the new message types are now available to Matlab.

Considering that the installation steps have been successfully completed, it should be possible to generate in Simulink the ROS Subscriber blocks to obtain the position of the hedgehog, which can be used to track the OMR for odometry validation. The previous steps are shared in detail because the diversity and the compatibility of the software components needed made it difficult to obtain the complete working solution.

Alternatively, to avoid the cumbersome steps enumerated, the open-source code of the ROS package for Marvelmind can be adapted to generate, in place of the custom Marvelmind message, some standard ROS messages (e.g., point type) that do not need special support, but this path can be challenging also.

2.3. Rapid Control Prototyping Using the ROS Node for Matlab-Simulink Integration

In the initial investigation [13], a simple rapid control prototyping environment was developed using an ROS node designed in Matlab-Simulink, and later, it was developed into the model depicted in Figure 4, where the three main components can be identified: the position acquisition from the odometry and IPS, the position controller used to follow the trajectory described by the waypoint vector, and finally, the ROS blocks used for sending the requested velocity references to the execution layer of the OMR.

The central part of the node is the position controller implemented as a Matlab function block. It was designed to steer the OMR to the prescribed pose received from the waypoint selector block, which keeps track of the current and next target waypoint on the trajectory path. Velocity saturation and acceleration limitation were applied to keep the OMR within nominal parameters and to minimize the risk of damaging collisions.

In the command output stage, manual switches were included so that simple motions and emergency pauses could be requested when the node was executed remotely from a PC, connected through the WiFi network to the rest of ROS nodes running on the OMR. This approach allowed for accelerated testing and improved debugging of the developed node by using Matlab-specific tools such as signal probes and data recordings.

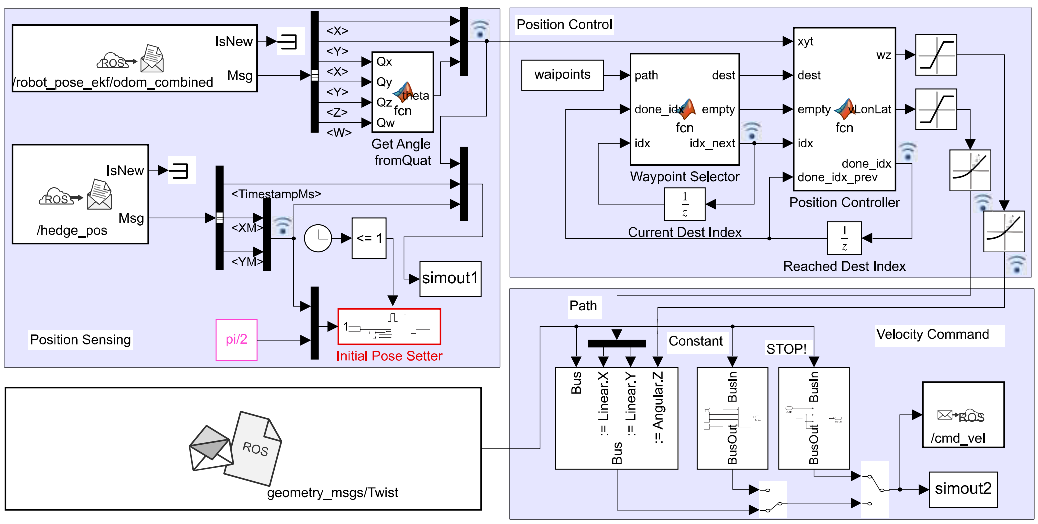


Figure 4. ROS application node implemented in Matlab with odometry initialization from the IPS.

2.4. Initial Tracking Results Using the ROS-Matlab Simulink Approach

With a setup similar to the one described in the previous subsection, some initial experiments were carried out using simple trajectories, such as a square shape [13], to investigate the performance of the odometry calculated by the integration of the raw relative speed to the ground based on the direct kinematics described in Equation (3), and the necessity of applying velocity compensations [20] to improve the results became apparent.

In Figure 5 are illustrated the recorded paths, as measured using the odometry and IPS, against the reference desired path. Since the trajectory controller was configured to work in closed-loop based on the position estimated using odometry, it can be noticed that, although the odometry was tracking the reference, the validation performed with the IPS revealed systematic deviations. Other inconveniences that can be noticed in the path recorded with the IPS, but that were not visible in reality, were the position estimation jitters of the IPS, which appeared especially in certain locations of the experimental area.

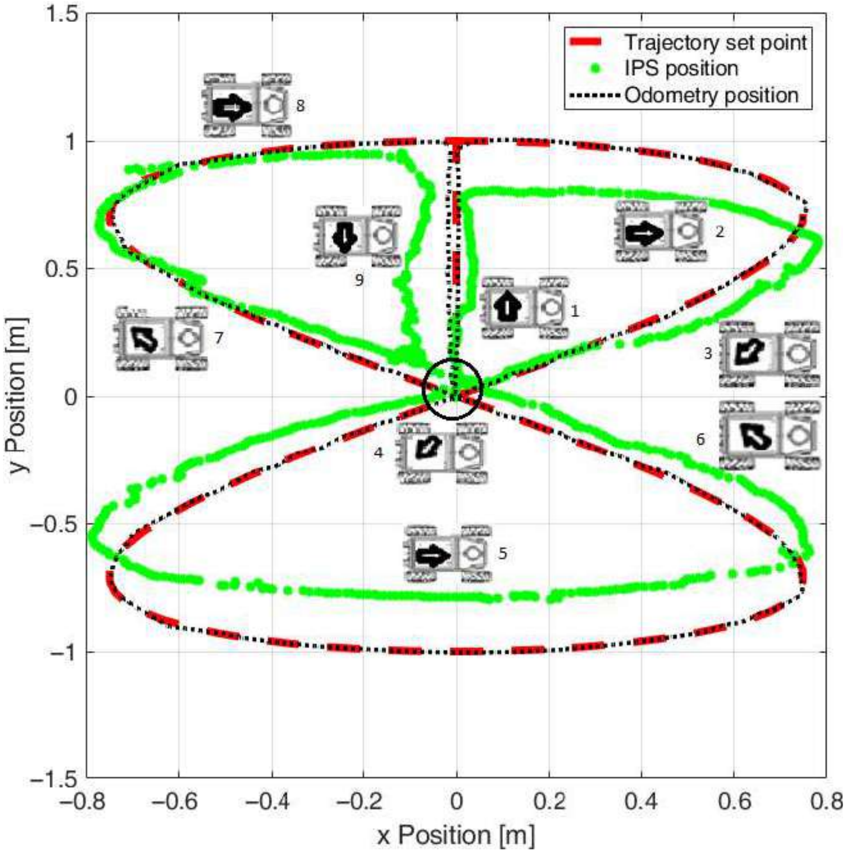


Figure 5. Initial experiments without odometry calibration (the numbered icons indicate the direction and the order of travel on the trajectory segments).

At that moment, the quality stream of the Marvelmind IPS localization was not used; however, it became obvious that the configuration of the IPS beacons was not ideal, and in certain areas, some of the stationary beacons missed the direct signal from the hedgehog and received only indirect propagation reflected on nearby walls. This observations motivated a more careful approach to the placement of the IPS stationary beacons, but also the interest in monitoring the reported position quality and attempting an independent trilateration implementation based on the raw beacon-to-beacon measurements, which can be received from the IPS after each localization in order to obtain a quantitative assessment of the position accuracy.

After implementing support for velocity reference compensations [20] at the execution level of the OMR based on the recorded systematic errors, the results from Figure 6 were obtained which showed a significant improvement, especially for the lateral direction. In Figure 6 also, it is visible that the OMR used a higher reference speed since the IPS position samples were more spread out and the overshoot of the trajectory controller was more obvious in the first direction change when starting to follow the actual Lissajous curve shape. By applying the calibration, the significant systematic lateral translation deviation was reduced from about 22 cm to under 5 cm.

Another issue that was identified after the initial calibration was an undesired rotation of the OMR while following the reference track. This aspect is not visible in Figure 6 because the combined odometry provided by the specialized ROS node that also uses the

IMU of the OMR managed to keep good track of the azimuth change, and as a result, the trajectory controller could perform an accurate-enough tracking.

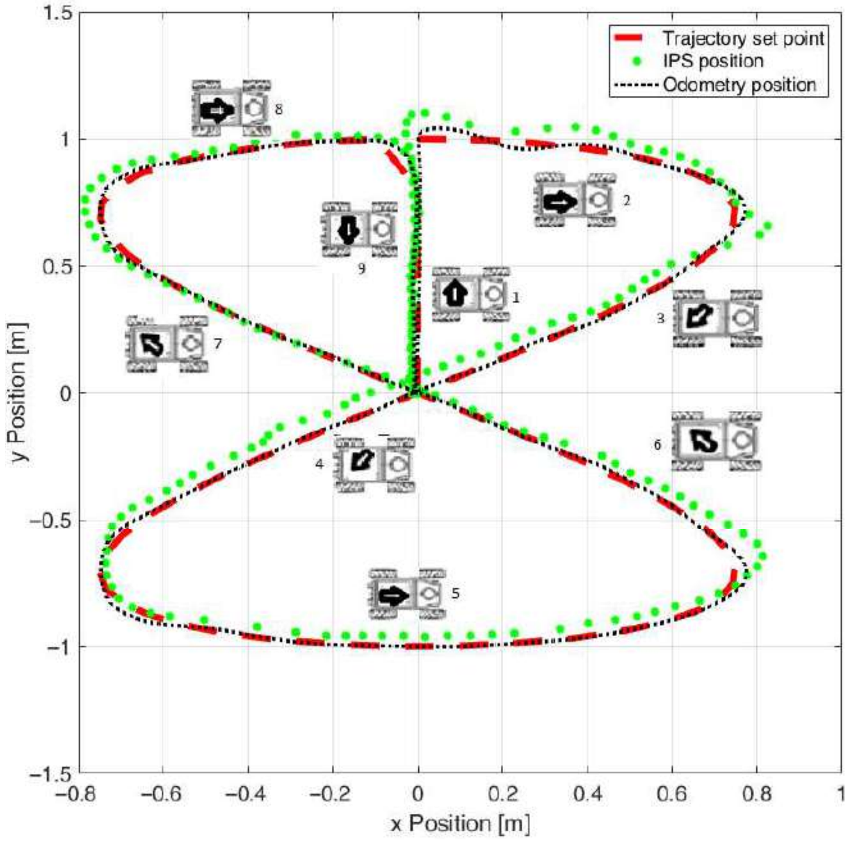


Figure 6. Initial experiments with odometry calibration (the numbered icons indicate the direction and the order of travel on the trajectory segments).

3. OMR Indoor Navigation Improved by Calibration of the Motion and Odometry

Autonomous navigation benefits from the good accuracy of the motion and odometry. In this process, there are three contradictory requirement concerns: high accuracy, robustness for different conditions, and the application of cost-effective sensors and methods. Good performance can be ensured with high mechanical precision for the OMR, which implies higher costs. Better position-finding sensors can solve the positioning control using a closed-loop approach. This also leads to a higher cost, as well as lower robustness.

In this context, a calibration method designed to compensate through software the undesired operation of the OMR due to mechanical imperfections is of interest since it can improve the base performance for motion- and odometry-based localization, which can only be useful for more advanced techniques such as sensor fusion.

Considering the observed issues regarding the initial results obtained in the previous section, which used a complex control structure made of a path planner based on waypoints and a positioning controller, a simplified open-loop control structure was considered for further analysis and calibration of the odometry. In this way, attention can be concentrated on the performance of the OMR, while other possible sources of errors are limited. As a result, a scripted environment based on Python was developed to apply sequences of motion requests (as relative velocity vectors that the OMR platform can execute) and, at the same time, to record the reported odometry data and the localization information from the Marvelmind IPS for later analysis and calibration information extraction.

In order to perform the experimental calibration of the OMR kinematics, certain practical challenges need to be taken in consideration and mitigated: mechanical accuracy and play of the OMR components, wheel slippage during aggressive maneuvers, floor quality, suspension response, integration errors, initial position, and orientation estimation. In this context, a practical approach needs to be constructed around simple operations that can be easily reproduced systematically, which motivated a simpler open-loop motion control solution. The entire experimental procedure followed the flowchart from Figure 7.

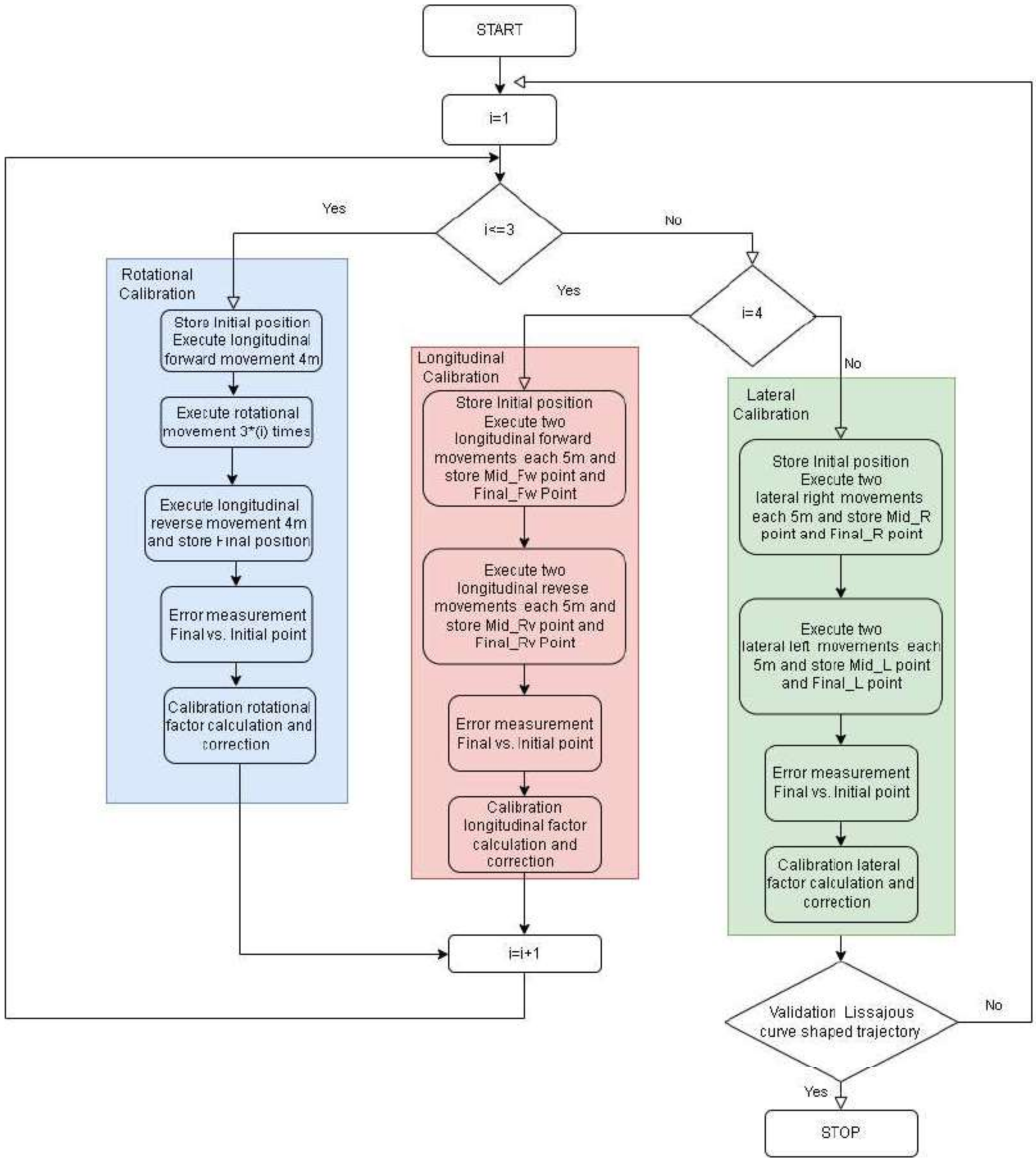


Figure 7. The flowchart of the experimental procedure.

3.1. *Open-Loop Motion Response for Orthogonal Movements*

In order to obtain good performance for complex movements, it is essential to obtain the desired response for simple orthogonal movements, i.e., pure longitudinal, lateral, or rotational motion, respectively, which are specific to the OMR. The research conducted on the OMR based on four Mecanum wheels showed that the direct usage of the mechanical parameters for solving the kinematic equations led to discrepancies, especially for the lateral and rotational motion. The observed motion errors can be classified into scaling errors (moving/rotating more or less than expected) and cross-talk errors (one orthogonal motion, i.e., lateral, produces another undesired orthogonal motion, i.e., rotational).

3.2. *Measurements of the Motion and Odometry Errors*

Depending on the type of motion error, different methods can be applied for experimental determinations. For translation scaling errors, a laser range finder was considered for measuring the actual motion due to its high accuracy. Alternatively, a well-set-up IPS together with an averaging strategy in specific reference points of a test path can be used to obtain a more complete picture of the actual motion in time. For rotational scaling error and the motion cross-talk errors, the methods were based on localization at relevant points from the reference trajectory using the IPS or the usage of the LiDAR to directly measure the orientation variation relative to some available reference (e.g., a wall). Although the IPS supports a special mode called *paired mobile beacons* to establish the orientation of a target that carries them, the method was not pursued due to its additional setup complexity and also because the starter kit used contained only four stationary beacons; converting one of them to a paired mobile beacon meant a reduction of the fixed beacons used for the multilateration of the position.

An important aspect can be the order in which the motion errors are determined, especially because of the noticed cross-talk from translation to rotation motions. In this context, it is easier to first determine the scaling compensation required for the correct rotation and then attempt the determination for translation and the cross-talk between translation and rotation, which cannot be easily separated. By starting with the determination of the rotation correction factor, then the proper translation–rotation cross-talk factor can be directly determined.

3.2.1. Rotational Movement Error

Depending on the amount of inaccuracy for the kinematics involved in rotation, an iterative practical approach may be required for easier determination. The technique for the iterations was similar, it being necessary in all cases to establish a reference direction, followed by a number of rotations in place, and then, after stopping, a final measurement of the orientation direction. The in-place rotation was executed by requesting the OMR platform to rotate at a fixed rate for the theoretical time needed to complete a specific number of rotations. Considering a firmware implementation on the OMR that limits the acceleration and deceleration to the same value to avoid slippage or over-currents, the previously described requested motion should be executed as expected.

For example, in the first iteration, a single rotation was performed to obtain an initial estimation of the correction factor, which was then applied in the inverse and direct kinematic transformations. In the next iterations, the number of in-place turns was increased to benefit from averaging-out the errors, such as those affecting the measurement of the initial and final orientation.

In the case that the IPS is to be used for determining the rotation correction factors, a method for evaluating the initial and final orientation is necessary if the paired mobile beacons option is not considered, as summarized in the left most (blue) column of Figure 7. The solution is to measure (averaging is recommended to improve accuracy) the initial position at stand-still using the IPS and then perform a longitudinal translation forward at a fixed speed and for a specific time, followed by a stop and a second accurate measurement of the intermediate stand-still position. The initial and first intermediate positions were used to determine the initial orientation by using the arc-tangent trigonometric function in both the IPS and odometry reference system. A number of in-place rotations were executed, followed by a complete stop and then a second orientation determination (based on a second intermediate point and a final point), this time by executing a reverse longitudinal motion. The steps described above were performed pragmatically using a script, which made the measurements and also applied the velocity references to the OMR while calculating in the end the correction factor. Assuming that the translation cross-talk is similar for both forward and reverse longitudinal motion and that the number of rotations is high enough, the correction factor should not be significantly affected by the eventual variation of the cross-talk effect or the imprecision of the IPS. The iterative approach is simple because the total correction factor can be calculated as the product of all the previous correction factors currently determined.

To minimize the limitations of the IPS's accuracy, the longitudinal translations are recommended to be as large as the area in which the IPS is least affected by the HDoP or other issues. In Figures 8 and 9 are presented captures from the web interface used for monitoring the OMR during the rotation correction factor determination after the first and second iterations. The grid spacing was 0.5 m, and the black and blue tracks represent the odometry and the IPS recordings, respectively. The IPS recording is drawn as separate points, useful for visually evaluating the dispersion of the position solutions along the path, certain areas being more affected than others. The light red square represents the current (final) location of the OMR according to the odometry, and the blue circle represents the current (final) IPS reported position. It can be noticed that the odometry position was initialized from the IPS, but no effort was made to align the initial orientation of the odometry and IPS coordinate systems, since this aspect was not relevant for the determination. Additionally, it is visible that the return path of the odometry also had a small deviation compared to the forward path, which can be explained by the approximate method used to execute the rotation by applying a reference rotational speed for a predetermined amount of time. The initial deviation for three complete rotations was determined to about 25.5°, and after calibration, it was reduced to under 3°.

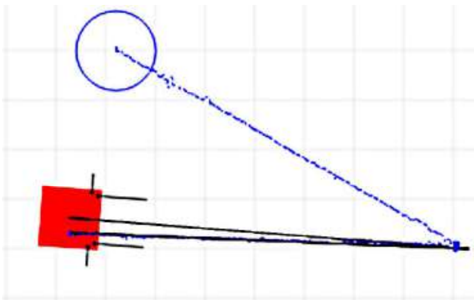


Figure 8. Rotation correction factor determination by using the IPS in the first iteration.

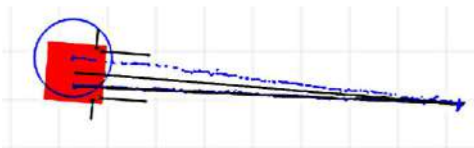


Figure 9. Rotation correction factor determination by using the IPS determination in the second iteration.

In a similar way, the kinematic rotation correction factor can be determined using LiDAR, in this case without it being necessary to perform the longitudinal motions to identify the initial and final orientation.

3.2.2. Translation Movement Error

As mention in [20], the translation movement errors are relatively easy to determine and then to calculate the velocity correction factors in order to compensate the undesired effects. In the current study, the observed rotation during translation, especially in the lateral direction, can slightly complicate the procedure, requiring a combined approach. In the simplest form, a constant reference speed was applied for a certain amount of time, and the initial and final position provided by the odometry and IPS were used to calculate the correction factor. To simplify the automation of determining the correction factor for translations, it is very useful to also perform the reverse motion so that the experiment is reset for a new determination. This also allows for a second set of data to be extracted.

In Figure 10 are represented the captures after two experiments for determining the longitudinal translation correction factor using the same representation conventions as in Figures 8 and 9. It can be noticed that the return paths had a slight deviation compared to the forward paths and were not always the same. This can be attributed to the mechanical play of the OMR's wheel assembly and also to the non-symmetric translation–rotation cross-talk effects for the forward–reverse directions, which is discussed next. In Figure 11 is presented the capture for another iteration of the experiment after the longitudinal correction factors were applied to compensate the kinematics. Besides the longitudinal translation correction factor, two more factors to compensate the rotation during the forward and reverse translations were applied so that the deviations were significantly reduced; more details follow in the next part. The longitudinal translation before calibration was about 25 cm shorter than the 10 m recorded by the odometry, while after calibration, only deviations under 5 cm were obtained.

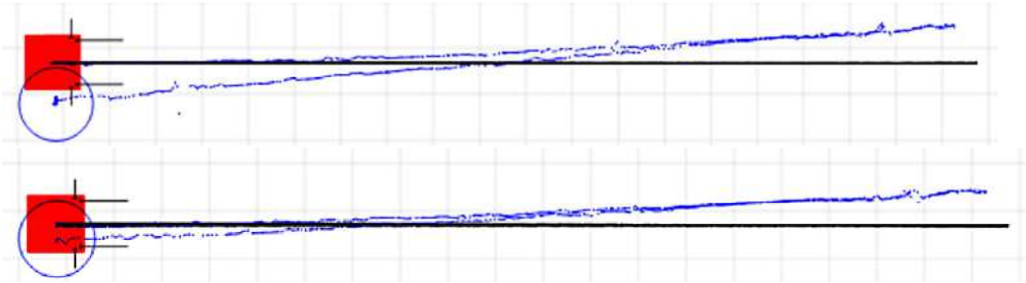


Figure 10. Two experiments for longitudinal translation correction factor determination by the IPS.

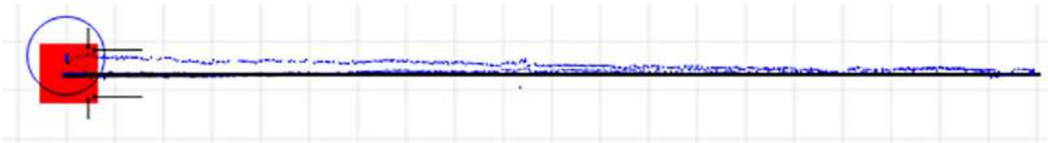


Figure 11. Result after applying the longitudinal translation correction factors obtained by the IPS (translation, forward-translation–rotation, and reverse-translation–rotation correction factors).

3.2.3. Undesired Rotational Movement during Simple Translation

Under the assumption that the coupling between the translation velocities and the rotational velocity side effect is linear, the recording of the paths from the IPS correlated with the odometry, as illustrated in Figure 12, can be used to establish the length and the radii of the arcs specific for each type of translation. Since, in the practical experiments, certain differences were identified between the motions in opposite directions (forward–reverse and left–right), it was of interest to determine four separate translation–rotation cross-talk correction factors, in addition to the three kinematic scaling factors described in [20].

By using the IPS, the arc radii can be determined by fitting the recorded path with an arc and then determining its length. The method has the advantage of reducing the impact of eventual IPS measurement errors. The ratio between the odometry displacement and the arc length gives the inverse kinematic translation correction factor. The curvature of the arc, defined as the reciprocal of the arc's radius, is the translation–rotation cross-talk correction factor used to calculate the compensation rotation velocity to obtain the desired straight motion. In order for the odometry not to measure the injected compensations, the direct kinematics need to be counter-compensated in reverse order with the reciprocal of the correction factors.

As an alternative to fitting, the arc radius can be determined by obtaining three accurate reference points along the track: the start point S, a mid-point M, and the final point F. The disadvantage of the method is that if a mid-stop is executed to obtain an averaged determination based on the IPS, the orientation of the OMR is likely to slightly change and affect the determination. Using the coordinates of the three points, the center C and the radius R of the circumscribed circle can be determined analytically. Furthermore, using the known vertices of the triangle $\triangle SCR$, the angle $\angle SCR$ can be determined and, then, the exact length of the arc using the radius R.

As a metric of performance, translation–rotation cross-talk deviations of up to 7 mrad m^{-1} for longitudinal and 38 mrad m^{-1} for lateral motion were recorded before calibration, while after calibration, the values decreased to 1.5 mrad m^{-1} and 5 mrad m^{-1}, respectively, as illustrated in Figure 13.

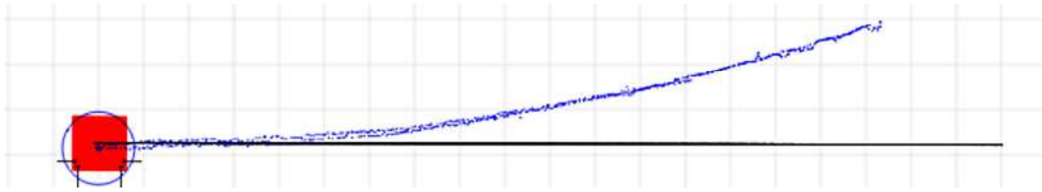


Figure 12. Lateral translation correction factors determination by the IPS (translation, left-translation–rotation, and right-translation–rotation).

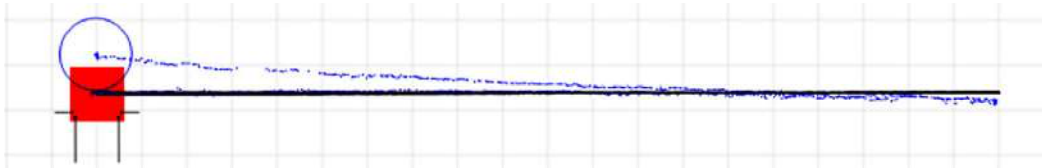


Figure 13. Result after applying the lateral translation and translation–rotation correction factors obtained by the IPS.

3.2.4. Experimentally Obtained Kinematic Correction Factors

By applying the previously described approach, the kinematic correction factors listed in Table 2 were obtained for the OMR used in the experiments. It can be noticed that the additional translation–rotation cross-talk compensation factors were similar, but not equal for translations in opposite directions.

Table 2. Correction factors.

Type	Factor	Unit of Measure
Longitudinal velocity	1.02	scaling—no unit
Lateral velocity	1.13	scaling—no unit
Rotational velocity	1.0236	scaling—no unit
Longitudinal-forward velocity to rotational	0.0072	rad m^{-1}
Longitudinal-reverse velocity to rotational	0.00009	rad m^{-1}
Lateral-left velocity to rotational	0.0389	rad m^{-1}
Lateral-right velocity to rotational	0.0277	rad m^{-1}

4. Experimental Setup and Real-Time Validation Subject to Odometry Calibration

In logistic systems and especially in the calibration phase of the OMR's development, it is beneficial to use an IPS, and it is also essential to measure its accuracy. Considering the unavoidable position solution uncertainties due to the ultrasonic system setup, the manufacturer describes qualitatively the confidence of the latest position solution as a percentage. In the research context of OMR platforms, it can be more useful to obtain a quantitative estimation of the uncertainty that would ensure the assessment of each obtained positioning solution sample.

4.1. Evaluation of the IPS and Improvement Attempts

Since there is no open information given about the position solution calculation method for the IPS used, to our knowledge, it became necessary to apply an external/parallel method to validate the offered position solutions, because during our experiments, significant deviations were observed in certain locations.

Among the useful functionalities provided by the API, there is an option for obtaining the raw point-to-point distances between the mobile beacon (named the hedgehog) and each of the up to four fixed beacons in the current localization zone. Thanks to a relatively open approach regarding the parts of the communication protocols used and an extensive

API, it is possible to develop external dedicated methods based on trilateration, methods that can also provide a better metric for assessing the accuracy.

The mobile beacon, represented by the blue point in the configuration dashboard in Figure 14, can be configured to provide several useful details for external position solution calculation after each internal position measurement: the position solution calculated internally, the raw point-to-point distances between the hedgehog and up to four fixed beacons involved in the localization, and the latest position quality percentage. The setup and configuration presented in Figure 14 allowed for easier replication of the tracking results.

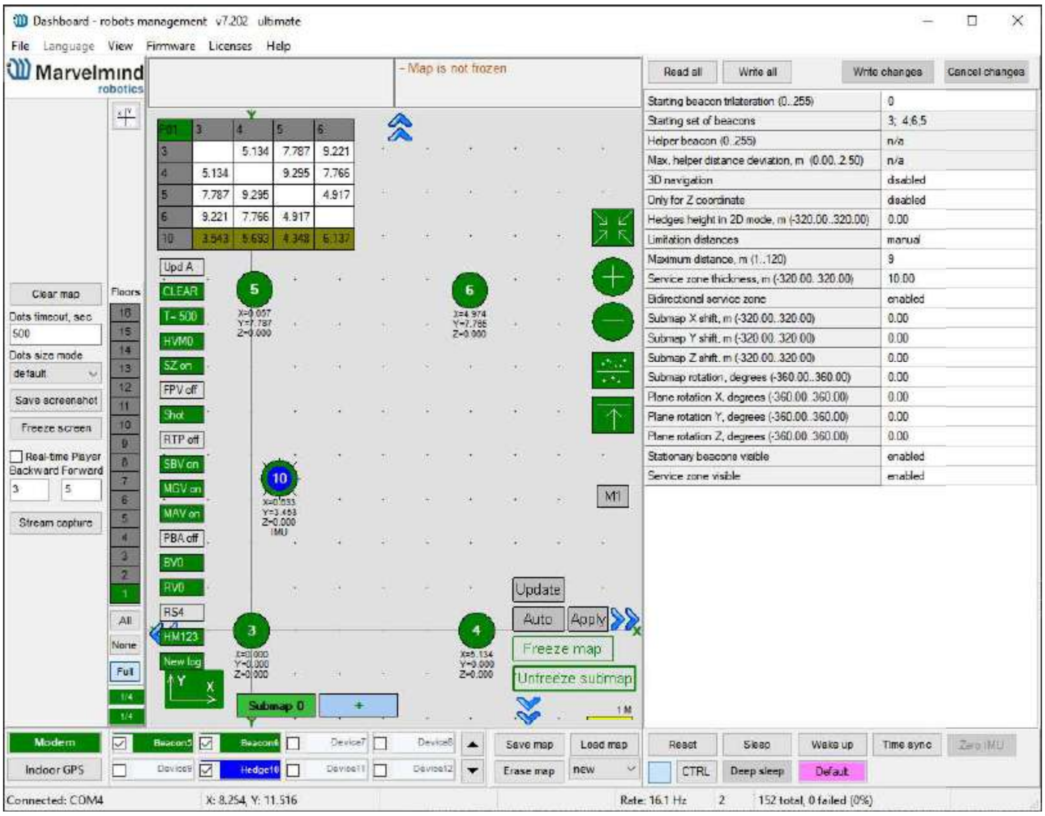


Figure 14. The Marvelmind app dashboard useful in the IPS's configuration.

In addition, the system outputs the position information for all the configured fixed beacons every 10 s, allowing for automatic map updating, which is useful for the external position calculation.

The positioning error is defined as in Equation (4) using the root mean square method over the deviations of the measured point-to-point distances, as reported by the IPS, and the corresponding distances between the estimated position solution and each of the fixed beacons. In Equation (4), n_b is the number of beacons (up to four for this IPS) involved in multilateration, (x_i, y_i) are the coordinates of the beacon i, d_i is the reported distance between the hedgehog and the beacon i, and (x, y) are the coordinates of the estimated position solution.

$$e = \frac{1}{n_b^2}\sqrt{\sum_{i=1}^{n_b}\left(\sqrt{(x_i - x)^2 + (y_i - y)^2} - d_i\right)^2} \quad (4)$$

Trilateration was performed externally using an iterative implementation that searches for the coordinate pair (x, y), at millimeter resolution, that minimizes Equation (4), which also quantitatively characterizes the uncertainty of the estimated position solution.

An example of the early attempts to track the calibrated lateral motion using the IPS and the external multilateration technique is depicted in Figure 15, where it can be seen that the unfiltered multilateration solutions were in certain regions strongly affected by indirect propagation of the ultrasound pulses, but the confidence circles helped point out the lack of precision associated with that determination. Furthermore, in some of the zoomed-in regions, especially the center one, there are areas where the multilateration method performed better than the IPS, a fact that was confirmed by both the less-dispersed track, but also by the very tight confidence circles, which indicate a very small RMSE for the solution.

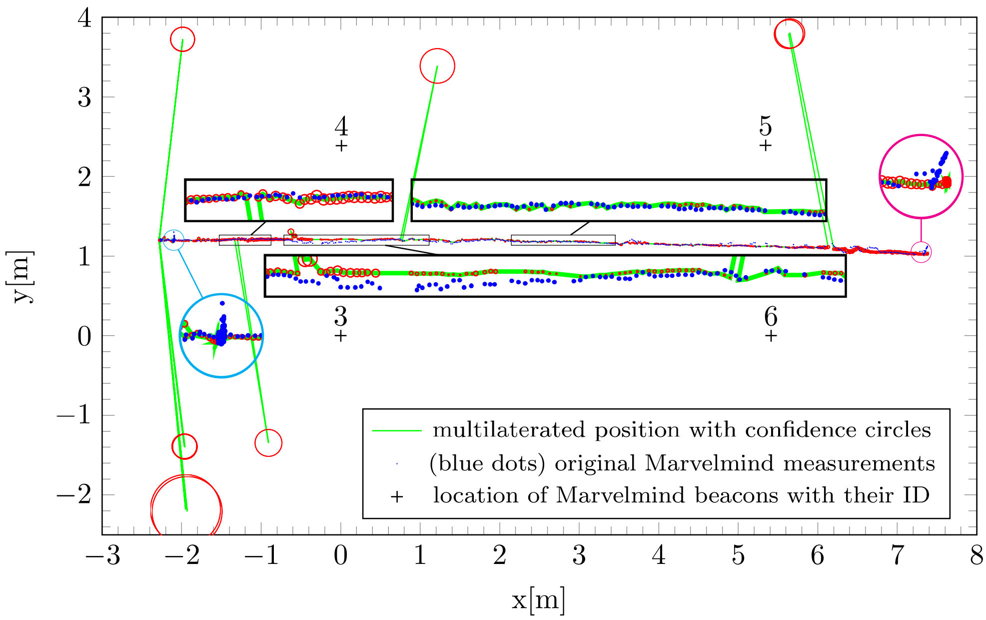


Figure 15. Lateral motion tracking after calibration.

4.2. Open-Loop Motion Tracking Performance Evaluation

In order to obtain smoother trajectories, in place of the generic trajectory controller developed using the Matlab rapid prototyping environment, which uses waypoints for navigation, a simplified open-loop trajectory generation was considered.

As a reference path for complex motion test scenario, a Lissajous-curve-shaped trajectory was chosen [20], described in Equation (5), where S is a scaling factor (the value $S = 1$ was used in the following tests). $\varphi \in [0, 2\pi]$ was used to generate the Cartesian coordinate pairs (x_r, y_r).

$$\begin{cases} x_r = S\sin(2\varphi) \\ y_r = S\cos(\varphi) \end{cases} \tag{5}$$

The needed speed vectors that were applied to the OMR in order to obtain the desired motion, without using a position controller, were calculated using the derivative of Equation (5) with respect to φ as in Equation (6).

$$\begin{cases} v_x(\varphi) = 2S\dot{\varphi}\cos(2\varphi) \\ v_y(\varphi) = -S\dot{\varphi}\sin(\varphi) \end{cases} \tag{6}$$

In order to exclude the negative impact of abruptly starting and stopping the OMR when following the Lissajous curve, the track was split into three segments: an acceleration part for which $\ddot{\varphi} = A$ until $\dot{\varphi}$ reached reference speed Ω, then a part executed at $\dot{\varphi} = \Omega$ until $\varphi = 2\pi - \dfrac{\Omega^2}{2A}$, the point from which $\ddot{\varphi}$ was set to $-A$ for deceleration during the last part.

In Figure 16 are comparatively illustrated the trajectories using the Marvelmind IPS (blue line), the multilateration method (green line), and the odometry track (black line) with respect to the reference trajectory (red) obtained with the OMR rotated 90° counterclockwise.

Lissajous-shaped trajectory are also marked by confidence circles. The maximum deviation of the distances between an estimated and real point was 0.2 m, in the case of uncalibrated odometry. This misalignment is more visible in the time response on the x coordinate, as can be observed in Figure 17. To save representation space, the positions of the IPS stationary beacons are not covered in Figure 16, but their identification numbers, types, and positions are listed in Table 3.

In order to evaluate the trajectory accuracy, the root-mean-squared deviation is universally used, based on pose measurements and reference values. The trajectory data used for RMSE evaluation were the Lissajous shape illustrated in Figure 16, where the calculated travel length based on the Euclidean distance was about 10 m. From Table 4, it can be observed that the mean-squared error revealed greater accuracy for the calibrated data compared to the raw values retrieved from the Marvelmind beacons.

Table 3. Map coordinates of the IPS beacons used for 2D localization.

Beacon ID	Beacon Type	X Position (m)	Y Position (m)	Z Position (m)
3	fixed	0.000	0.000	0.400
4	fixed	5.767	0.000	0.400
5	fixed	0.485	4.809	0.400
6	fixed	5.445	4.813	0.400
10	mobile	-	-	0.400

Table 4. Root-mean-squared trajectory tracking deviation.

RMSE	Uncalibrated Kinematics		Calibrated Kinematics	
	Odometry	**IPS**	**Odometry**	**IPS**
Position (m)	83×10^{-4}	1119×10^{-4}	85×10^{-4}	171×10^{-4}
x (m)	89×10^{-4}	1518×10^{-4}	86×10^{-4}	168×10^{-4}
y (m)	76×10^{-4}	445×10^{-4}	83×10^{-4}	173×10^{-4}

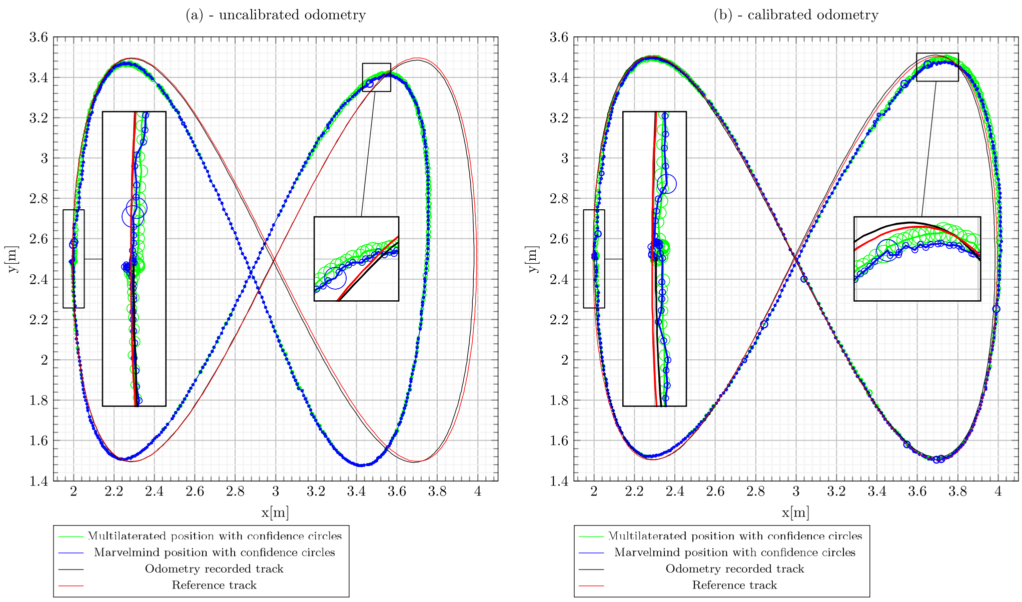


Figure 16. Lissajous-curve-shaped trajectory tracking before (**a**) and after (**b**) kinematics calibration.

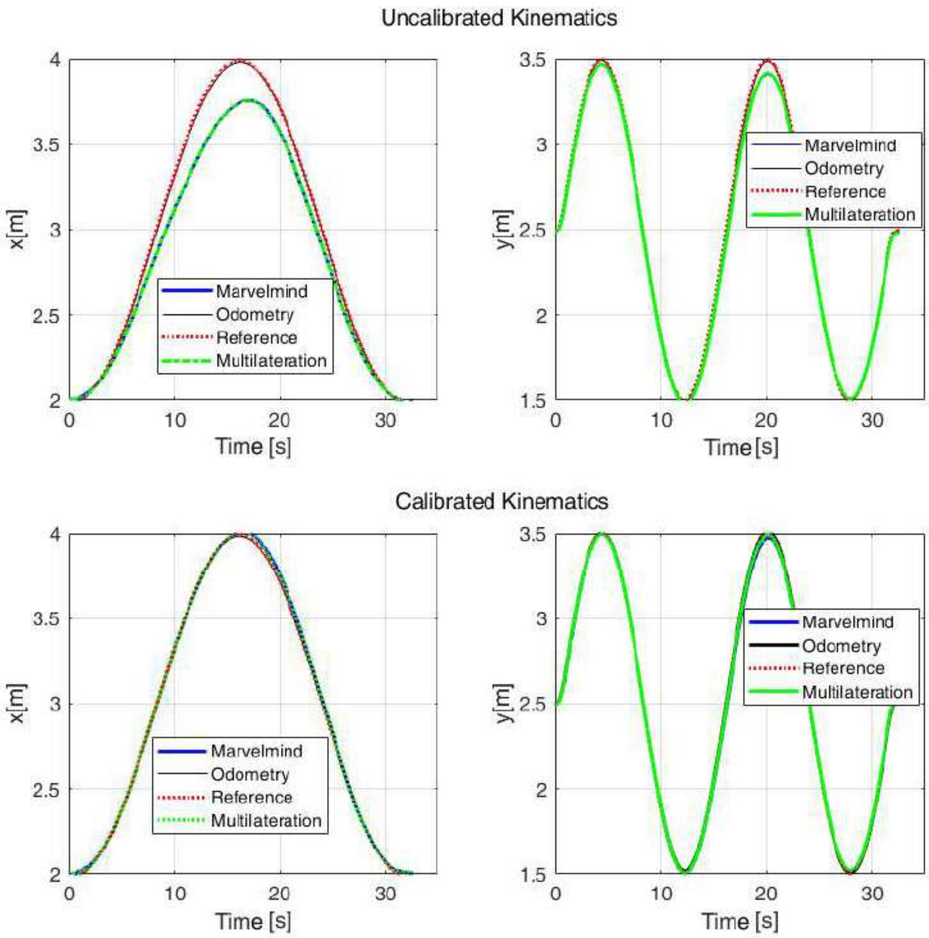


Figure 17. The comparison between the estimated and real position of the OMR on the Lissajous curve trajectory.

5. Conclusions

In this study, path performance evaluation was performed starting from orthogonal movements to complex trajectories. The shortcomings of the experimental OMR can be overcome by proper calibration. The initial experimental results revealed that odometry calibration reduced the error propagation, the results being compared with the IPS data. The experiments for calibration were carried out in open-loop with the aim of not interfering with the performance of the navigation controller. The corrections that affected the calibration factors had the role of compensating to a good extent the various constructive imperfections of the robot, such as the wheel assembly error, uncertain wheelbase, and last but not least, the slippage on the running surface.

The OMR was equipped with the Marvelmind IPS for indoor position validation, the acquisition of the data being carried out by using the Matlab environment through ROS application nodes. The evaluation of the IPS was motivated by the lack of open information about the position solution method. Therefore, an external method based on trilateration was proposed in order to provide a better metric, and the root-mean-squared error was used as the performance criterion both in the odometry and the IPS for kinematics evaluation.

Experiments with the OMR showed that the trajectory of the calibrated robot was closer to the ideal trajectory and, thus, could validate the effectiveness of the proposed approach. The experiments were repeated for different speed values of the OMR, and no correlation was observed between the speed and the correction parameters, under the conditions in which the acceleration was kept within the limits, in which it did not produce slips in the starting and stopping modes.

To overcome the drawback of different sensors for indoor positioning and methods, the future approach is the fusion method for the different sensors of the OMR (LiDAR, IMU, stereo camera, encoders, and IPS) to achieve highly accurate, precise position and navigation data.

The whole thread of the work was from the perspective of a systematic approach, the experimental design being a well-defined research methodology, useful to have greater trust in odometry.

The future research direction involves GPS fused with an IMU, the sensor being slightly attached to the GPS in order to predict and update the vehicle position even in the event of GPS signal loss.

Author Contributions: Conceptualization, C.-C.D., C.B. and A.-T.P.; methodology, A.-T.P., C.-C.D. and C.B.; software, A.-T.P.; validation, C.B. and C.-C.D.; formal analysis, A.-T.P., C.B. and C.-C.D.; investigation, C.-C.D. and A.-T.P.; writing—review and editing, A.-T.P., C.B. and C.-C.D.; project administration, C.-C.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI – UEFISCDI, project no. PN-III-P2-2.1-PTE-2019-0731, contract no. 19/2020, within PNCDI III, 3rd stages and by a research grant of the TUIASI, project number GnaC2018-67/2019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available upon request from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

API	Application Programming Interface
BTSPP	Bluetooth™ Serial Port Profile
CAN	Controller Area Network
FDMA	Frequency Division Multiple Access
GPS	Global Positioning System
HDoP	Horizontal Dilution of Precision
IA	Inverse Architecture
IMU	Inertial Measurement Unit
IPS	Indoor Positioning System
ISM	Industrial, Scientific, and Medical radio band
LiDAR	Light-based Detection and Ranging system
MC	Matlab Console
MF	Multi-Frequency
NiA	Non-Inverse Architecture
o-LED	organic Light-Emitting Diode
OMR	Omnidirectional Mobile Robot
OS	Operating System
RC	Radio remote-Controlled
RMSE	Root-Mean-Squared Error
ROS	Robot Operating System
FreeRTOS™	Free Real-Time Operating System
TDMA	Time Division Multiple Access
ToF	Time of Flight
USB	Universal Serial Bus
VCP	Virtual Serial COM Port
VS17	Microsoft Visual Studio 2017

References

1. Lee, H.J.; Yi, H. Development of an Onboard Robotic Platform for Embedded Programming Education. *Sensors* **2021**, *21*, 3916. [CrossRef] [PubMed]
2. Li, Y.; Dai, S.; Zhao, L.; Yan, X.; Shi, Y. Topological Design Methods for Mecanum Wheel Configurations of an Omnidirectional Mobile Robot. *Symmetry* **2019**, *11*, 1268. [CrossRef]
3. Wang, C.; Liu, X.; Yang, X.; Hu, F.; Jiang, A.; Yang, C. Trajectory Tracking of an Omni-Directional Wheeled Mobile Robot Using a Model Predictive Control Strategy. *Appl. Sci.* **2018**, *8*, 231. [CrossRef]
4. Qian, J.; Zi, B.; Wang, D.; Ma, Y.; Zhang, D. The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System. *Sensors* **2017**, *17*, 2073. [CrossRef] [PubMed]
5. Staal, A.S.; Salvatierra, C.G.; Albertsen, D.D.; Mahendran, M.; Ravichandran, R.; Thomsen, R.F.; Hansen, E.B.; Bøgh, S. Towards a Collaborative Omnidirectional Mobile Robot in a Smart Cyber-Physical Environment. *Procedia Manuf.* **2020**, *51*, 193–200. [CrossRef]
6. Angerer, S.; Strassmair, C.; Staehr, M.; Roettenbacher, M.; Robertson, N. Give me a hand—The potential of mobile assistive robots in automotive logistics and assembly applications. In Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications (TEPRA2012), Woburn, MA, USA, 23–24 April 2012. [CrossRef]
7. Doroftei, I.; Grosu, V.; Spinu, V. Omnidirectional Mobile Robot—Design and Implementation. In *Bioinspiration and Robotics: Walking and Climbing Robots*; M.K. Habib: Vienna, Austria, 2007; pp. 511–528. [CrossRef]
8. ABIresearch. The Tech Intelligence Experts, Homepage. Available online: https://www.abiresearch.com/market-research/product/7778043-commercial-and-industrial-robotics/?src=svcrecent (accessed on 22 April 2022).
9. Azizi, M.R.; Rastegarpanah, A.; Stolkin, R. Motion Planning and Control of an Omnidirectional Mobile Robot in Dynamic Environments. *Robotics* **2021**, *10*, 48. [CrossRef]
10. Carbonell, R.; Cuenca, A.; Casanova, V.; Piza, R.; Salt Llobregat, J.J. Dual-Rate Extended Kalman Filter Based Path-Following Motion Control for an Unmanned Ground Vehicle: Realistic Simulation. *Sensors* **2021**, *21*, 7557. [CrossRef] [PubMed]
11. Dosoftei, C.; Horga, V.; Doroftei, I.; Popovici, T.; Custura, S. Simplified Mecanum Wheel Modelling using a Reduced Omni Wheel Model for Dynamic Simulation of an Omnidirectional Mobile Robot. In Proceedings of the 2020 International Conference and Exposition on Electrical And Power Engineering (EPE), Iasi, Romania, 22–23 October 2020; pp. 721–726. [CrossRef]
12. Dosoftei, C.C.; Popovici, A.T.; Sacaleanu, P.R.; Gherghel, P.M.; Budaciu, C. Hardware in the Loop Topology for an Omnidirectional Mobile Robot Using Matlab in a Robot Operating System Environment. *Symmetry* **2021**, *13*, 969. [CrossRef]
13. Dosoftei, C.C.; Popovici, A.T.; Sacaleanu, P.R.; Budaciu, C. Real-Time Motion Control of an Electric Driven OMR using a ROS to Matlab Bridged Approach. In Proceedings of the 2021 25th International Conference on System Theory, Control and Computing (ICSTCC), Iasi, Romania, 20–23 October 2021; pp. 160–165. [CrossRef]
14. Amsters, R.; Demeester, E.; Stevens, N.; Lauwers, Q.; Slaets, P. Evaluation of Low-Cost/High-Accuracy Indoor Positioning Systems. In Proceedings of the 2019 The Fourth International Conference on Advances in Sensors, Actuators, Metering and Sensing (ALLSENSORS), Athens, Greece, 24–28 February 2019; pp. 15–20.
15. He, X.; Aloi, D.N.; Li, J. Probabilistic Multi-Sensor Fusion Based Indoor Positioning System on a Mobile Device. *Sensors* **2015**, *15*, 31464–31481. [CrossRef] [PubMed]
16. Mainetti, L.; Patrono, L.; Sergi, I. A survey on indoor positioning systems. In Proceedings of the 2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 17–19 September 2014; pp. 111–120. [CrossRef]
17. Onalaja, O.; Adjrad, M.; Ghavami, M. Ultra-wideband-based multilateration technique for indoor localization. *IET Commun.* **2014**, *8*, 1800–1809. [CrossRef]
18. Expósito Jiménez, V.J.; Schwarzl, C.; Martin, H. Evaluation of an indoor localization system for a mobile robot. In Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), Graz, Austria, 4–8 November 2019; pp. 1–5. [CrossRef]
19. Glowinski, S.; Ptak, M. A kinematic model of a humanoid lower limb exoskeleton with pneumatic actuators. *Acta Bioeng. Biomech./Wroc. Univ. Technol.* **2022**, *24*, 145–157. [CrossRef]
20. Li, Y.; Ge, S.; Dai, S.; Zhao, L.; Yan, X.; Zheng, Y.; Shi, Y. Kinematic Modeling of a Combined System of Multiple Mecanum-Wheeled Robots with Velocity Compensation. *Sensors* **2020**, *20*, 75. [CrossRef] [PubMed]
21. Pavel, M.D.; Rosioru, S.; Arghira, N.; Stamatescu, G. Control of Open Mobile Robotic Platform using Deep Reinforcement Learning. In Proceedings of the SOHOMA 2022, 12th International Workshop on Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future, Valencia, Spain, 22–23 September 2022; pp. 1–12.
22. Marvelmind Company. Precise (±2 cm) Indoor Positioning and Navigation for Autonomous Robots, Drones, Vehicles and Humans. Available online: https://marvelmind.com/ (accessed on 29 September 2022).
23. Bârleanu, A.; Băitoiu, V.; Stan, A. Digital filter optimization for C language. *Adv. Electr. Comput. Eng.* **2011**, *11*, 111–114. [CrossRef]
24. Bârleanu, A.; Băitoiu, V.; Stan, A. FIR Filtering on ARM Cortex-M3. In Proceedings of the 6th WSEAS European Computing Conference, Prague, Czech Republic, 24–26 September 2012; pp. 490–494.
25. Specht, M. Experimental Studies on the Relationship Between HDOP and Position Error in the GPS System. *Metrol. Meas. Syst.* **2022**, *29*, 17–36. [CrossRef]

26. Marvelmind Robotics. ROS Marvelmind Package Installation Instructions. Available online: http://marvelmind.com/pics/marvelmind_ROS.pdf (accessed on 2 October 2022).
27. Kitware. CMake Download Page. Available online: https://cmake.org/download/ (accessed on 29 April 2021).
28. Mathworks. Matlab Supported and Compatible Compilers for R2021a—All Products. Available online: https://www.mathworks.com/support/requirements/supported-compilers.html (accessed on 29 April 2021).
29. Python Software Foundation. Python Releases for Windows. Available online: https://www.python.org/downloads/windows/ (accessed on 29 April 2021).
30. Mathworks. Matlab—ROS System Requirements. Available online: https://www.mathworks.com/help/ros/gs/ros-system-requirements.html (accessed on 29 April 2021).
31. Marvelmind Robotics. ROS Marvelmind Package Git Repository Page. Available online: https://bitbucket.org/marvelmind_robotics/ros_marvelmind_package (accessed on 29 April 2021).

A new approach to create a realistic virtual model of a cylindrical robot using Automation Studio

C C Dosoftei, A Lupu and C M Pascal
Gheorghe Asachi Technical University of Iasi, Faculty of Automatic Control and Computer Engineering, Bd. Dimitrie Mangeron, nr. 67, 700050, Iasi, Romania

E-mail: cdosoftei@ac.tuiasi.ro

Abstract. Industry 4.0 is mainly represented by the digitalization technologies of physical industrial assets, thus obtaining the cyber-physical systems. A key role in digitalization developing is indeed played by possibility to develop a realistic virtual model (3D model) and using this model in virtual simulations to test the functionality of the system at possible scenarios from real world. The paper presents the modelling a mechatronic system represented by a three-dimensional cylindrical type manipulator in Automation Studio software and aims creating cinematic animation which will be used in the field of virtual commissioning. This approach in Industry 4.0 paradigm is studied in the academic environment and in the industrial research from last decade time, because generate major benefits such as in remote accessibility for learning, time efficiency from planning to real setup with possibility of detection and debugging of errors in the system, for minimizing time commissioning and predictive maintenance.

1. Introduction

Industry 4.0 (I4.0) creates a lot of opportunities for companies and research institute because it is the first time when an industrial revolution is predicted a-priori, not observed retroactive. The basic components of the concept I4.0 are cyber-physical systems (CPS), Internet of Things (IoT), Internet of Service (IoS) and Smart Factory. A few years ago were defined six design principles that characterize main components of I4.0: interoperability, virtualization, decentralization, real-time capability, service orientation and modularity [1]. From all of this virtualization is the key enabler of I4.0 and means that CPS are able to monitor physical processes.

Developing of specialized tools for computer-aided engineering (CAE) over the last decade come to sustain all virtualization process from CPS and Smart Factory. One of these tools is Automation Studio (AS) conceived by Famic Technologies Inc.. AS software which comes with the advantage given by the fact that is not behind of industrial equipment manufacturer, is developed for both direction academia and industry, being used as a design and simulation tool in the fields of automation and electrical control system, pneumatics, and hydraulics system and for fluid power systems design [2]. This software allows to go in–depth beyond in the direction of virtual engineering through standard modules and libraries which interact with each other during the simulation process to allow user to create complete systems that behave as they would in real life. The standard 3D Editor workshop [3] brings the concept of virtual system to another level. Up to a medium complexity level it is possible to develop an object in the AS editor, while for complex system it is necessary to build new models based on CAD data. This is the case of actual paper when we will build geometry data in

CATIA software and assembling these components via joints and synchronizing this mechatronic mechanism with power and control technologies of the other workshops from AS. In this way the PLC controller implemented and virtual robot equipped with sensors will be connected in closed loop behaviour and validation will be done through software in the loop (SIL). This new approach with Virtual System workshop from AS creates the realistic customized model of any mechatronic system which can be included in the second generation of CPS [4], which collect the information from real sensors and actuators.

2. Hardware structure of robot

Comprehending the complexity of robots and their industrial applications involve a multidisciplinary knowledge starting with mathematics, continued with mechanical and electrical systems and last but not least computer and control technology. Industrial robot technology progresses rapidly and along with it increases diversity of application, the most popular being arc and spot welding, materials handling, machine tending, painting, picking, packing and palletizing assembly [5,6]. There are a lot of applications where the pneumatic robots find their place, coming with few important advantages: firstly is price level than they can move quite smoothly, the maintenance complexity relative simply.
In the present paper is presented the development of mathematic modelling and simulation for the pneumatic robot. The three-degree of freedom (DOF) robot which is an industrial manipulator arm – cylindrical type, one revolute moving and two prismatic moving (RPP), there is in Department of Automatic Control and Applied Informatics and is used in academic aim – figure 1.

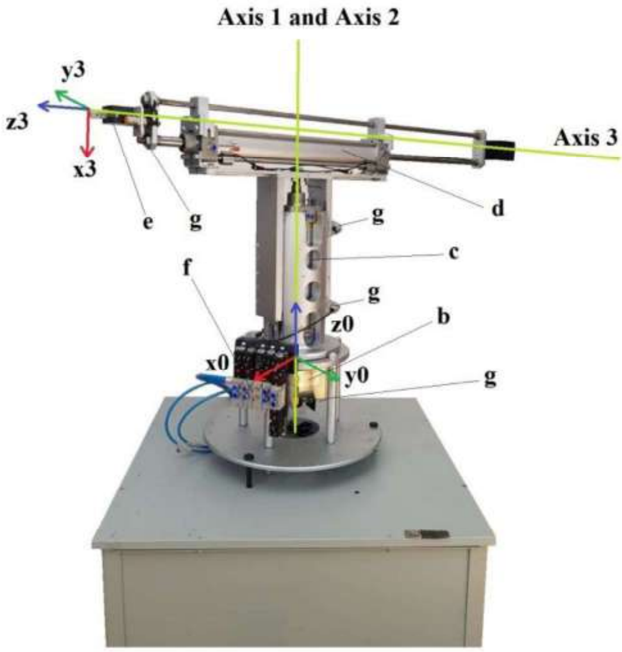


Figure 1. 3DoF Pneumatic Robot.

The main features of unit are: (a) all drives are pneumatic, (b) the revolute moving between 0-90º is realized on Axis 1 with a pneumatic actuator L/R, (c) first translation moving on Axis 2 is realized by a double action cylinder – U/D , (d) second prismatic moving on Axis 3 is implemented with double end-double acting cylinder – F/B, (e) the effector is realized with a 180º angular gripper – G, (f)

directional valves to control the actuators pneumatic are mounted on manifold, (g) proximity sensors and (h) control panel for operational task.
All cylinders have a permanent magnet on the piston being able to send, through reed sensors mounted on the cylinders sliding axis, electrical signals to indicate its position. If appear restriction of spaces as is the case with UD cylinder and robot chassis are used inductive proximity switches that detect metal brackets on the moving parts of the robot. Mainly of pneumatic robot application very useful is the end-stroke cushioning which quieten the impact of the piston on the end block. The control of cylinders is done with solenoid valves with electric command, mechanical spring return, and external servo pilot. For rotary actuator is used a bistable valve 5Ports/3Ways and for linear cylinders and gripper valves used are mono-stable valves 5Ports/2Ways.

3. Forward cinematic model

One widely used representation of forward kinematics is based on the Denavit-Hartenberg (DH) parameters. The obtained model gives the pose of the end-effector frame from robot joint parameters. The chosen kinematic structure is illustrated in figure 2, where O_0 is the origin of the base frame associated with the first link. Frames having origins in O_1 and O_2 refer to the next two links according to the DH formalism. O_3 is the origin of the end-effector frame.

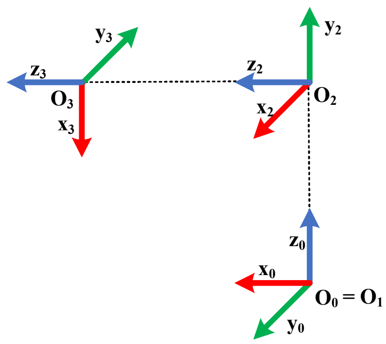


Figure 2. Define link frame.

The entire structure is succinctly described in the table 1 in term of DH parameters, where θ, d1 and d2 are variable. Distances O_1O_2 and O_2O_3 define d1 and d2, respectively.

Table 1. Define DH parameters of each link.

link	θ_i	α_i	a_i	d_i
1	θ	0	0	0
2	90°	90°	0	d_1
3	-90°	0	0	d_2

The "equation (1)" is obtained as the product of three DH transformation matrices. The Cartesian position (x, y, and z) for the origin O_3 as a function of the joint values (θ, d1, and d2) is given by the last column. Using these values, the robot workspace can be determined and visualized. Figure 3 shows the eight reachability points, red dots when θ is 0 and blue ones for 90º.

$$\begin{bmatrix} 0 & -\sin(\theta) & cos(\theta) & d2 * cos(\theta) \\ 0 & cos(\theta) & sin(\theta) & d2 * sin(\theta) \\ -1 & 0 & 0 & d1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

According to the motion for each joint, the resulted trajectory will be neither a straight line nor an arc of circle, when two or three joints are used simultaneously. For example, the pneumatic characteristics of the robot give the three joints trajectory illustrated in figure 3. In some cases the trajectory is proper to move to the end point, but returning home can create an unwanted path.

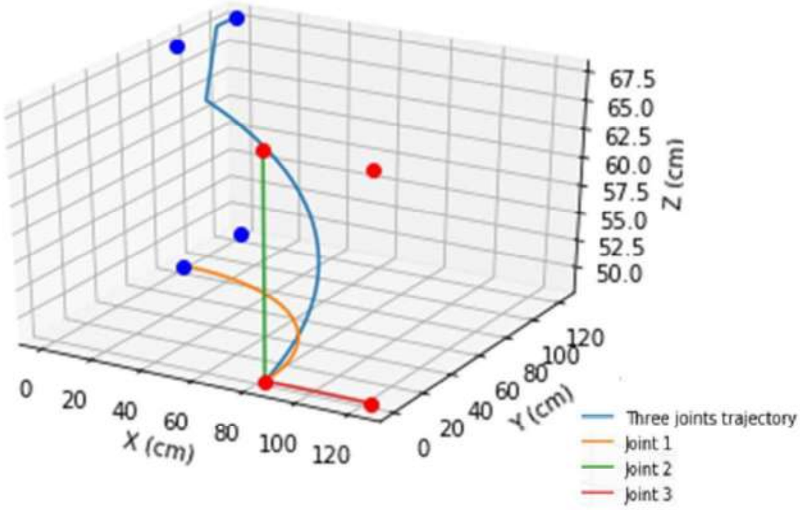


Figure 3. Reachability points and several trajectories.

4. Digital twin of the 3DoF pneumatic robot realized in Automation Studio

A digital twin represents a dynamic virtual model of a thing, system or process that relies input data from real world to understand its state, respond to changes, improve operations and add value. By integrating the virtual and physical worlds, the digital twin continuously enriches the data feed coming from the sensors of real robot enables real-time monitoring of robot and timely analysis of data to quickly diagnose and fix failure, schedule preventive maintenance to reduce/prevent downtimes, optimize robot operation and implementing upgrades [7].

Model-Based Design, commonly defined as MBD, is a mathematical and visual design methodology for designing complex systems in different domains, such as in automotive, aerospace, motion control and industrial equipment applications. From literature review about the MBD concept [7-11] the ubiquitous V-model which describes a relation between each phase of the development life cycle and its associated phase of testing is a specific form, figure 4, for implementation a realistic virtual complex mechatronic model using Automation Studio.

One of the challenges with modelling and simulation software is to push over their limit imposed by high-level modelling approach when composing of virtual systems is done using predefined elements from internal library. This is the case when the approach is low level modelling as is defined in [12], the building of new components/mechatronic systems start from CAD data and followed by assembling and functional interaction between them and / or other workshops such as pneumatic, hydraulic, electrical, control. Based on this flow, in actual work is creating and validating in Automation Studio the model of pneumatic robot guided by the V-diagram from figure 5.

After the requirements have been determined exactly what the robot must do in a given real-world scenario, the second step is to design the geometrical components which compound the robot from base to gripper. All these 3D components must be saved in files with format .IGS. In the third phase must be open a new project in AS with a 3D diagram editor, import files designed in Catia and started to assemble the robot in a cascade way.

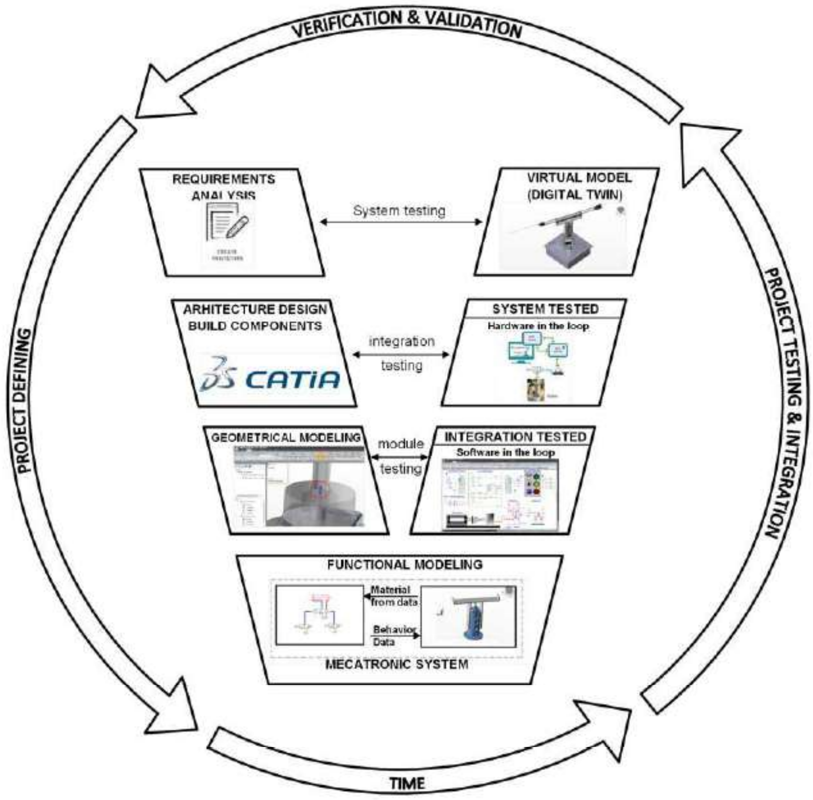


Figure 4. V-model of implementation in Automation Studio.

Figure 5 presents assembling between components that create the kinematic animation of Up/Down cylinder, first translation movement of robot in Axis2.

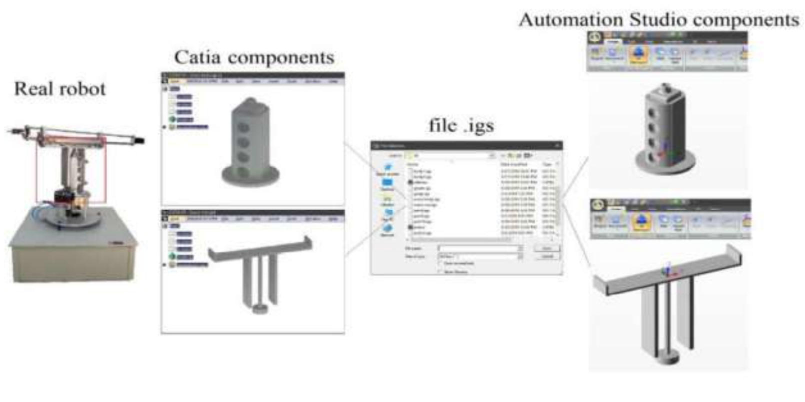


Figure 5. Low level modelling of robot chassis.

AS – 3D workshop contains six positioning tools (three are called “Constraints” and other three are called “Displacements”) necessary to displace and adjust the instances in the 3D animation of an assembly in a certain perspective. After completing the alignment process between the two components, a sliding motion is defined for the translation U / D cylinder, to complete the kinematic animation between the two components: source and target. These two components are made parallel to each other when displacing the source toward the target. The result of this stage with geometrical modelling for U/D cylinder capsulated in the chassis of robot is presented in figure 6:

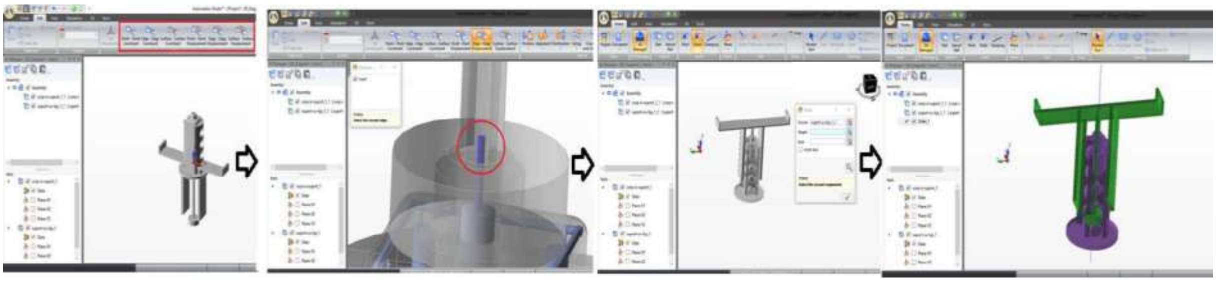


Figure 6. Few steps from aligning to geometrical model U/D cylinder.

In the next step called functional modelling, above geometrical model will be overlaid a standard cylinder from pneumatic library and so the kinematic animation will be related to the displacement of the cylinder controlled by a valve. After choosing the type of cylinder it is necessary to customize the proprieties of cylinder similar with real cylinder code 41M2P080A0200 (e.g. diameter Ø80mm, rod diameter Ø25mm, stroke 200 mm, inclination –vertical, external load 20kg, and more). Continuing this path and add all others components (rotary actuator for joint revolution on Axis 1, double ended piston rod for translation on Axis3 and the end of arm tooling – gripper) is defined the completely model of robot. The digital model, represented in figure 7, is ready to use when is assigned manually electrical inputs from sensors and electric outputs to actuators.

5. Testing robot model - Software in the loop method

On testing branch first step is used an approach based on software in the loop simulation, where mechatronic model and controller are coupled in closed loop behaviour. Because the pneumatic drives in a mechatronic system are suitable for a sequential control, the testing of the robot model operation will be carried out using a PLC.

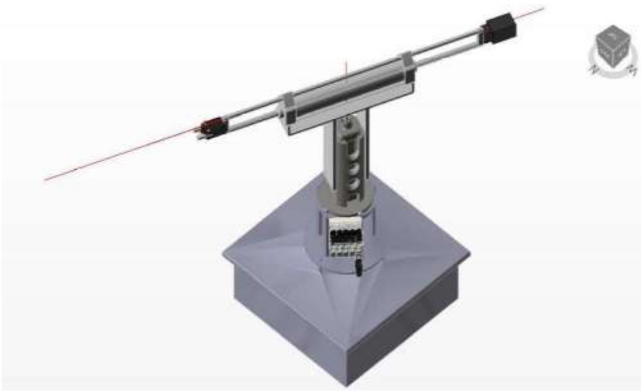


Figure 7. Digital twin robot.

On testing branch first step is used an approach based on software in the loop simulation, where mechatronic model and controller are coupled in closed loop behaviour. Because the pneumatic drives in a mechatronic system are suitable for a sequential control, the testing of the robot model operation will be carried out using a PLC. The global variables defined in functional model, like inputs/outputs, are connected to inputs/outputs of PLC. To interface robot with operators is developed in specialized AS workshop a human-machine interface (HMI), figure 8, from where the control can be settled in two modes: manual and automatic. The algorithm is implemented in this paper in a graphical formalism – figure 8, known as grafcet or SFC, preferred because is a useful and efficient environment to working in projects where different professional groups are involved to understand automation system.

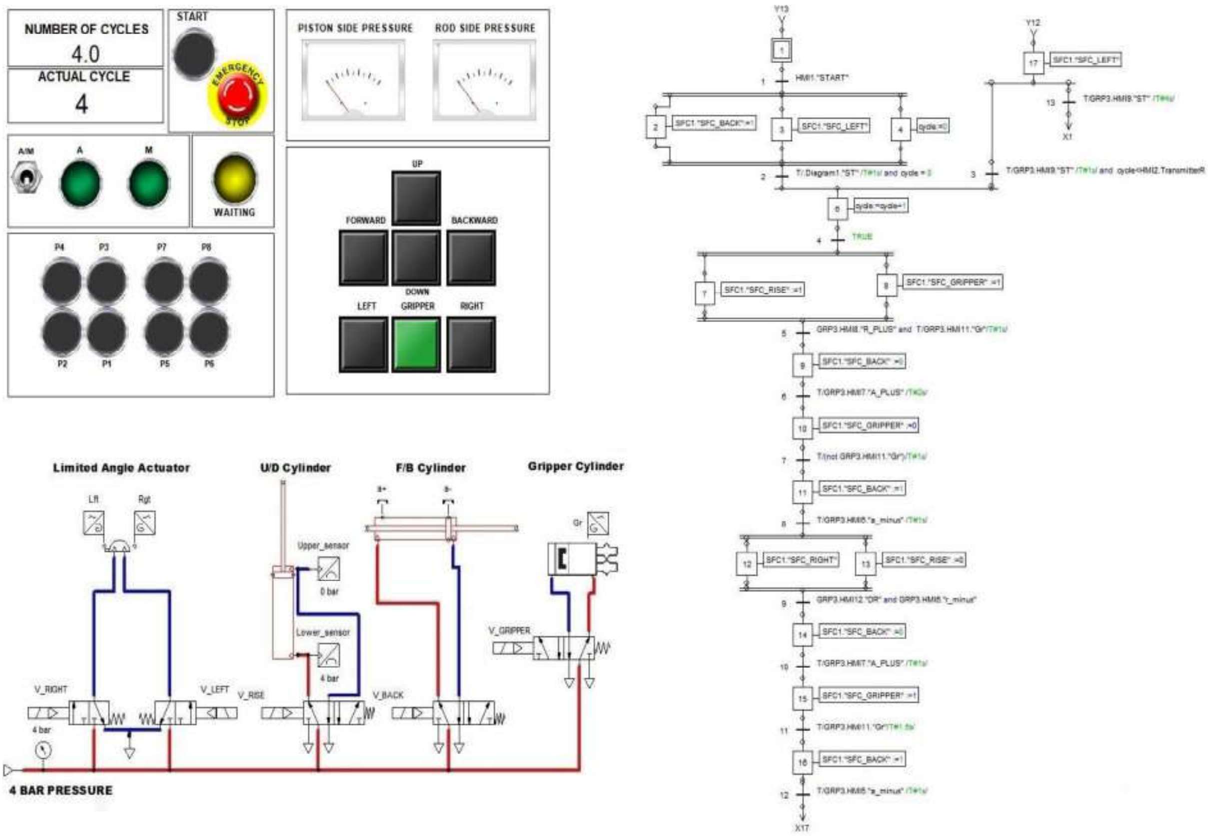


Figure 8. Software in the loop simulation (HMI, pneumatic circuit of robot and grafcet).

In simulation activities all pneumatic components of robot are operated with a pressure stabilized at 4 bars. With specialized tools from AS it is obtained the evolution of different measurements (positioning, speed, acceleration, pressure). In our case in figure 9 is presented the variation of speed for U/D cylinder which is the main cylinder of robot if we take in consideration its task.

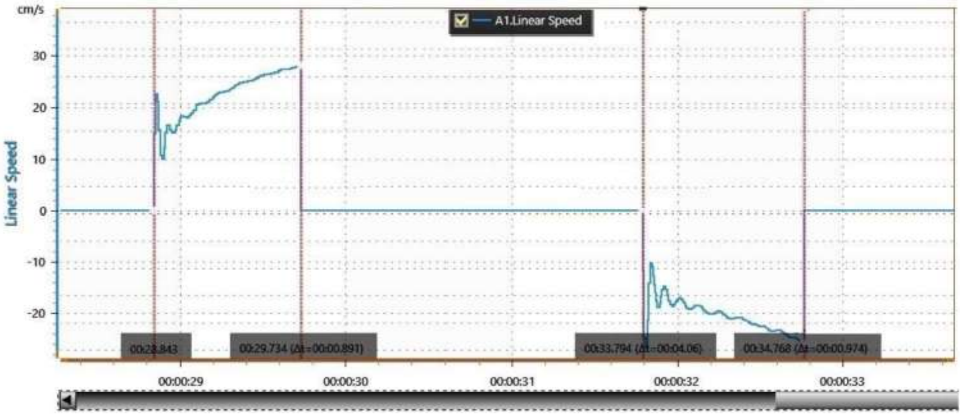


Figure 9. Variation of linear speed of U/D cylinder.

6. Conclusion

The method presented in this paper is a low-level modelling procedure to obtain a digital twin of 3DoF pneumatic robot, by following the V-model adapted for AS software up to level of SIL. Development requires an interdisciplinary approach. The valid model can be used for virtual commissioning where the real PLC is linked by digital model through a tunnel implemented with an OPC Server. In this way is necessary to implement the algorithm that has been validated in this paper in ladder format for PLC and HMI. From the authors point of view, in context of forwarding to Industry 4.0, the near future will bring to delivering real and model of virtual machine, this being an important differentiator between the manufacturing companies.

7. References

[1] Hermann M, Pentek T and Otto B. 2016 Design principles for industrie 4.0 scenarios. In 2016 *49th Hawaii international conference on system sciences (HICSS),* pp 3928-3937, 10.1109/HICSS.2016.488

[2] Nohacova L and Nohac K 2010 Possibilities of Computer Simulation in Power Engineering and Environmental Engineering. *In Innovations and Advances in Computer Sciences and Engineering*, pp. 1-5, Springer, Dordrecht

[3] Automation Studio™. Famic Technologies Inc. https://www.famictech.com/edu/ [accessed 02April 2019]

[4] Zhou J, Li P, Zhou Y, Wang B, Zang J and Meng L 2018 Toward new-generation intelligent manufacturing, *Engineering*, **4(1)**, pp 11-20

[5] Kochan A 1999 New robot applications satisfy the automotive industry's need for even greater flexibility. *Industrial Robot: An International Journal*, **26(5)**, pp 349-353

[6] Spong M W and Vidyasagar M 2008 *Robot dynamics and control*, John Wiley & Sons, New Jersey, USA, p 9

[7] Madni A M, Madni C C and Lucero S D 2019 Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems*, *7*(**1**), pp 1-7

[8] Elm W C, Gualtieri J W, McKenna B P, Tittle J S, Peffer J E, Szymczak S S and Grossman J B. 2008 Integrating cognitive systems engineering throughout the systems engineering process, *Journal of Cognitive Engineering and Decision Making*, **2(3)**, pp 249-273

[9] Wikipedia. "V-Model", https://en.wikipedia.org/wiki/V-Model [accessed 5th March 2019]

[10] Morton S D 2001 The Butterfly Model for Test Development, https://www.agileconnection.com/ [accessed 2nd April 2019]

[11] Tierno A, Santos M M, Arruda B A and da Rosa J N 2016 Open issues for the automotive software testing. In *2016 12th IEEE International Conference on Industry Applications (INDUSCON),* pp. 1-8
[12] Hoffmann P, Schumann R, Maksoud T M and Premier G C 2010 Virtual Commissioning Of Manufacturing Systems A Review And New Approaches For Simplification. In *2010 - 24th European Conference on Modelling and Simulation*, Kuala Lumpur, Malaysia, pp. 175- 181

Acknowledgments
This work was supported by a research grant of the TUIASI, project number GnaC2018_190.

symmetry

Article

Hardware in the Loop Topology for an Omnidirectional Mobile Robot Using Matlab in a Robot Operating System Environment

Constantin-Catalin Dosoftei *, Alexandru-Tudor Popovici, Petru-Razvan Sacaleanu, Paul-Marcelin Gherghel and Cristina Budaciu

Department of Automatic Control and Applied Informatics, "Gheorghe Asachi" Technical University of Iasi, 700050 Iasi, Romania; alexandru-tudor.popovici@academic.tuiasi.ro (A.-T.P.); petru-razvan.sacaleanu@student.tuiasi.ro (P.-R.S.); gherghel.paul@ac.tuiasi.ro (P.-M.G.); cristina.budaciu@academic.tuiasi.ro (C.B.)

* Correspondence: constantin-catalin.dosoftei@academic.tuiasi.ro

Abstract: The symmetry of the omnidirectional robot motion abilities around its central vertical axis is an important advantage regarding its driveability for the flexible interoperation with fixed conveyor systems. The paper illustrates a Hardware in the Loop architectural approach for integrated development of an Ominidirectional Mobile Robot that is designed to serve in a dynamic logistic environment. Such logistic environments require complex algorithms for autonomous navigation between different warehouse locations, that can be efficiently developed using Robot Operating System nodes. Implementing path planning nodes benefits from using Matlab-Simulink, which provides a large selection of algorithms that are easily integrated and customized. The proposed solution is deployed for validation on a NVIDIA Jetson Nano, the embedded computer hosted locally on the robot, that runs the autonomous navigation software. The proposed solution permits the live connection to the omnidirectional prototype platform, allowing to deploy algorithms and acquire data for debugging the location, path planning and the mapping information during real time autonomous navigation experiments, very useful in validating different strategies.

Keywords: logistic robot; omnidirectional mobile robot; kinematic model; symmetrical configuration; hardware in the loop; navigation; ROS environment

Citation: Dosoftei, C.-C.; Popovici, A.-T.; Sacaleanu, P.-R.; Gherghel, P.-M.; Budaciu, C. Hardware in the Loop Topology for an Omnidirectional Mobile Robot Using Matlab in a Robot Operating System Environment. *Symmetry* **2021**, *13*, 969. https://doi.org/10.3390/sym13060969

Academic Editors: Aviv Gibali, Kok Lay Teo and Yonghong Wu

Received: 6 May 2021
Accepted: 28 May 2021
Published: 30 May 2021

1. Introduction

Due to the implementation of new technologies in many industrial plants, the efficiency of production activities has seen a continuous increase and the bottleneck has moved to the logistics area. Logistic includes a variety of processes such as: picking, packing, warehousing, inventory, delivery, and routing. In the last decade, the intelligent robot technology was extended to non-manufacturing sectors as logistic processes, and the logistic robots appeared especially for picking, packaging, palletizing and handling as a link between these operations [1]. Another factor favouring the optimization in the logistics field is represented by the development of autonomous systems. Initially the idea transposed in automated guided vehicles (AGVs), which are systems able to move following a fixed route between some predefined points. The operation of such a system in the field of logistics results in a number of shortcomings such as increased costs of system installation, lack of flexibility of systems to a route change, fleet management, rigidity from the perspective of collaboration with other systems or the human operator [2]. A superior technology to these AGVs is represented by autonomous mobile robots (AMRs), through which all the described disadvantages are eliminated.

The main changes are in the perception system where smart sensing solutions appeared: laser scanners, stereo vision cameras, inertial measurement unit, localization sensors and advanced processing using an additional embedded system capable of fast processing the information. AMRs can make decisions based on the perception and give to the robot an advanced mobility in a dynamic environment, which may be continuously changing. In recent years, AMRs are regular occupants in the modern logistics system, helping warehouse workers to fulfil orders with increased promptness.

The AMRs used in internal logistics contributes to improving processes by increasing productivity, reducing downtime due to lack of components and more efficient use of human resources to focus where it can bring more added value. Nowadays, AMRs is one of the fastest expanding fields of scientific research wherein academic groups and industrial companies frequently work together. AMRs attract attention more and more from the perspective of applications that have already branched off in various areas, as in search and rescue missions [3], planetary exploration [4], medical care [5], intervention in extreme environments like mining [6] and military operations [7], agriculture [8], household and office applications [9], logistics and manufacturing applications [2,10–12], as well as other industrial and non-industrial applications.

Current logistic facilities typically represent high-traffic environments, with narrow aisles among diverse obstacles where omnidirectional mobile robots (OMRs) are a better suited solution for improved manoeuvrability.

The main contribution of the paper is a comprehensive description of a practical approach for developing an OMR system useful to those who want to extend their research from pure simulations into practical higher level applications, since the majority of the recently published research papers focus mainly on simulation results. In this regard, our study might be of interest to researchers who want to design and follow the necessary steps for the realization of the hardware in the loop (HiL) architecture using recent hardware (NVIDIA Jetson Nano, light based detection and ranging system (LiDAR), depth-camera, etc.) and software (Robot Operating System (ROS) nodes developed in MATLAB).

This paper is structured as follows: Section 2 summarizes the related work that uses OMRs in the logistic field, Section 3 describes the ROSY platform prototype with essential features regarding the mechanical design and the associated kinematic model, the hardware configuration, the vehicle controller firmware, followed by the description of the ROS software topology. In Section 4, real-time motion control using ROS nodes implemented in MATLAB is detailed. Moreover, the experimental evaluation methodology is presented and the OMR position control performance is demonstrated by experimental results. Two types of scenarios are proposed: the first is related to the validation of tracking predefined trajectories and the second refers to building a map of the environment using the simultaneous localization and mapping (SLAM) algorithm with inputs from LiDAR and the robot odometry module. The platform is also tested to navigate between points defined in the map which qualified the mobile platform to be equipped with conveying system and used in logistic area. Some limitations and difficulties from experimental tests are also described. Discussions of the results and future work conclude the paper in Section 5.

2. Related Work

Recent independent studies reflect significant research on Mecanum wheeled robots, the study [13] compares recent research results with various OMR prototypes in the context of logistic applications, and the authors provide a review for the development of mobile robots based on Mecanum wheels which were previously used in different research centers.

Considering the issues of traditional mobile robots applications, the Mecanum wheeled platforms gain attention in different groups of researchers, and while the focus was initially on simulation results, currently it is moving towards applied, real-time, implementations. In terms of simulations, there are consistent studies for kinematic modelling [2,7,14], their main focus being the controller design, which can be of different forms based on classical or advanced modern approaches [14–16]. Moreover, path planning is also an important subject, especially in the logistic framework [17,18].

From the practical implementation point of view, different OMR prototypes have been designed and implemented to achieve autonomous navigation, the platforms have to obtain environment perception, localization, path planning and trajectory following abilities [19–22]. In most of the research studies, the STM32 micro-controller is the core of the low level control system, which performs various data calculations and real-time processing for different type of embedded computers [20,21]. The authors from [19] proposed a low cost robot using an Arduino based system for the low level controller and Raspberry Pi as the on board computer which runs a GNU/Linux distribution. Due to the recent trends and the complexity of the tasks required for logistics applications, the currently proposed solution is deployed on a NVIDIA Jetson Nano embedded computer hosted locally on the robot, that runs the autonomous navigation software. The research presented in this paper is part of a larger project, aimed at developing an intelligent logistics system using autonomous OMRs - identified by the name ROSY-Logistic.

The main objective consists of the development of an intelligent logistics system using autonomous robot for flexible exchange of items between different fixed conveyor systems, all operations being coordinated by a warehouse management system. The ROSY platform is meant to transport materials between predefined locations in a dynamic warehouse environment by autonomously optimizing its trajectory plan in response to its surroundings. The initial steps for developing the autonomous OMRs specialised in exchanging items between conveyor systems are detailed in this study. The main purpose of the current paper is to address symmetrically the integration testing level of the V-model process by designing a HiL architecture appropriate for the technologies used in the research project, such as the integration of ROS.

3. Omnidirectional Mobile Robot in a Logistic Application

The workflow of this project follows the well-known symmetric V-model development process methodology [23,24] presented in Figure 1. The approach starts on the design (left) branch by defining project requirements and goes through the lower layers to obtain an actual implementation that is verified and validated by following the symmetric testing branch that deals with module, integration and system testing, respectively.

The initial steps for developing the autonomous OMRs, specialised in exchanging items between conveyor systems are detailed in the following sections. The content of the study is addressed by the integration testing level of the V-model process through designing a HiL architecture appropriate for the proposed technology used in this research project, the ROS software package integration.

Due to the complexity of a mobile robotic structure and especially of an omnidirectional structure, in approaching the development of high-performance structures, a hierarchical leadership structure is adopted, top-down, with a strict organization, each hierarchical level being completely subordinated to the higher hierarchical level. This hierarchical approach also allows a special flexibility in the design and realization of mobile robotic structures, the changes made on a module having minimal influences on the functional structure. The control of a mobile robot can be approached either from a kinematic or from a dynamic perspective. The kinematic perspective consists in decoupling the control in two overlapping loops: the kinematic loop and the dynamic loop. The dynamic approach considers only one loop that ensures a dynamic global control. However, this last approach has a number of disadvantages: the necessary analysis as well as the real-time calculation become very complex. The kinematic approach is simpler and overall stability can be guaranteed [18]. Thus, although for physical reasons the robot model is a dynamic one, this dynamic can be neglected if the actuators used can develop much higher accelerations than required and therefore the prescribed torque is developed instantly relative to the time constants of the system. These considerations allow the development of a sufficiently general control structure, based on the kinematic model of the robot [25]. In Figure 2 it is shown the structure of the controller, derived from a typical cascade structure which contains 3 control loops: the trajectory planning loop, the kinematic loop and the dynamic loop. In this way, if the dynamic loop is much faster than the kinematic loop which, in turn, is much faster than the planning loop, then the stability of the system is guaranteed.

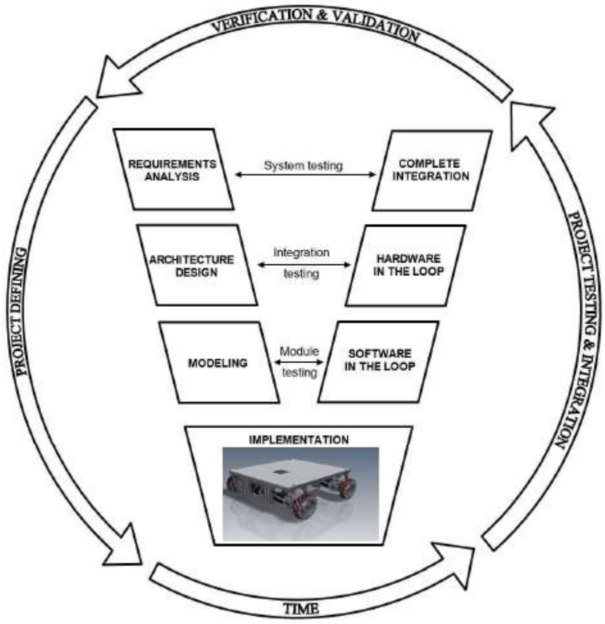


Figure 1. V-model development process methodology.

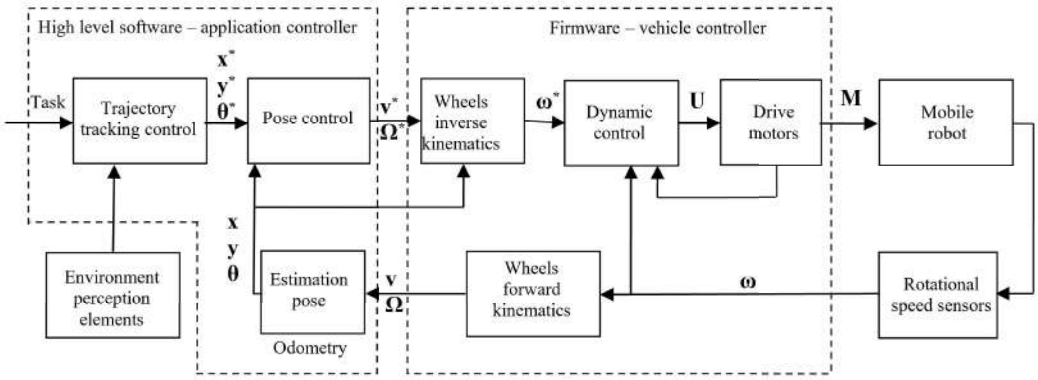


Figure 2. Controller structure of the mobile robot.

The position of the mobile platform which navigates in 2D space can be completely specified by means of the pose, consisting of three scalar sizes:

$$\boldsymbol{p} = [\mathbf{x}\ \mathbf{y}\ \boldsymbol{\theta}]^T \tag{1}$$

The pose derivative generates the speed vector:

$$\dot{\boldsymbol{p}} = [v_x\ v_y\ \boldsymbol{\Omega}]^T \tag{2}$$

Dynamic control can include a current control loop and takes into account the inertial masses of rotation and translation of the robot, as well as strong forces and torques in order to obtain control voltages, **U**, of motors which in turn are used to determine imposed rotation speeds, ω^*. High level control can consider some aspects in the phase of generating the reference path (avoidance of collisions, singular configurations, etc.) depending on the degree of intelligence of the designed mobile robot. Starting from the measured rotational speeds, ω, using the direct kinematic model of the mobile robot, the speed vector from Equation (2) can be obtained and then the position estimation. In this way it is possible to implement the middle level control, which involves a control of the position from a kinematic perspective.

3.1. Mechanical Architecture

The platform used in this study consists of four Mecanum wheels driven by individual motors and has important capabilities, especially for industrial environments, as it is able to move in any direction whilst spinning around its vertical axis. In order to minimize the vibrations caused by the spacers between rollers which affect the stability, the platform is provided with suspension mechanism based on traditional dampers. The robot was completely set up for autonomous navigation utilizing ROS, camera and a 360° LiDAR.

Figure 3 presents the geometrical model of the OMR prototype with the coordinates system assignments to each wheel, platform configuration, and all variables necessary for developing the kinematic model. As described at the beginning of Section 3, the control system deals with two kinematic transformations: the forward model which uses the speeds of all wheels to determine the relative speed of the platform, while the inverse kinematic model takes the components of the decomposed relative speed of the platform in order to obtain the required speed for each wheel.

The building process of the kinematic model of the omnidirectional platform with four Mecanum wheels having the arrangement of rollers direction in a square is a classic bottom-up approach, that starts from the process of composing the movement (v_{g_i}) in the Cartesian coordinate system of the roller that is in contact with the floor ($o_i x_i y_i z_i$). The movement is translated to the wheel's angular velocity (ω_i) with its attached coordinate system ($O_{\omega_i} X_{\omega_i} Y_{\omega_i} Z_{\omega_i}$), and the wheel's radius, R. The resulting velocity vector v is determined by instantaneous translation velocities of the robot (v_x respectively v_y) in the coordinate system of chassis $OXYZ$.

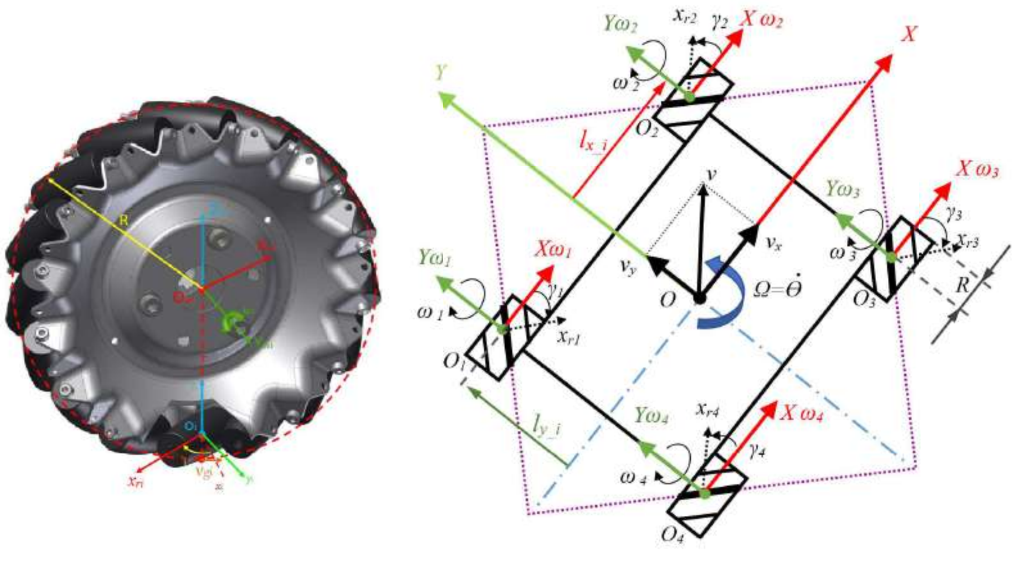


Figure 3. Description of the Cartesian coordinate system from the Mecanum wheel and symmetrical OMR chassis with four Mecanum wheels (all conventions are in concordance with the right-handed Cartesian coordinate system and representation of the wheel in 2D space is judged in terms of wheel tracks left on ground).

The position of each wheel i is uniquely determined in relation to the distance considered as a vector (thus having a sign depending on the direction of the axis to which it refers) as well as rotation angle between the wheel frame and roller frame (for a Mecanum wheel this angle is equal ±45°). In concordance with notations from Table 1 the particularized symmetric values for the structural parameters from Figure 3 are presented in Table 2. Therefore the impact of the linear velocity vector (v_i) of the wheel i and the velocity of the roller in contact with the ground (v_{g_i}) on the velocity vector of the robot can be calculated [26–28] by:

$$v_i + v_{g_i}\cos(\gamma_i) = v_x - l_{y_i}\Omega \tag{3}$$

$$v_{g_i} \sin(\gamma_i) = v_y + l_{x_i}\Omega \quad (4)$$

Because v_{g_i} is an uncontrollable variable of the passive roller, it will be eliminated through substitution from Equations (3) and (4):

$$v_i = v_x - l_{y_i}\Omega - \frac{1}{\tan(\gamma_i)}(v_y + l_{x_i}\Omega) \quad (5)$$

By customizing the parameters from Table 2 for each wheel and transforming linear velocity of each wheel to angular wheel velocity ($v_i = R\omega_i$), the wheels speed equations can be written as:

$$\begin{cases} R\omega_1 = v_x + v_y - (l_x + l_y)\Omega \\ R\omega_2 = v_x - v_y - (l_x + l_y)\Omega \\ R\omega_3 = v_x + v_y + (l_x + l_y)\Omega \\ R\omega_4 = v_x - v_y + (l_x + l_y)\Omega \end{cases} \quad (6)$$

Table 1. Kinematic model variables and their definitions.

Variable	**Definition**
v_x	instantaneous longitudinal velocity component of the robot
v_y	instantaneous lateral velocity component of the robot
Ω	rotational speed
v_{g_i}	velocity of roller from wheel i in contact with the ground
ω_i	angular wheel i velocity
v_i	instantaneous longitudinal velocity of the wheel i
l_x	half distance between front and rear wheel axles
l_y	half distance between left and right wheels
l_{x_i}	x coordinate of wheel 3 i relative to the robot center O
l_{y_i}	y coordinate of wheel i relative to the robot center O
γ_i	rotation angle between the wheel frame and roller frame
R	radius of wheel

Table 2. Mechanical parameters for each wheel i of the rectangular-symmetric ROSY platform.

Wheel Index *i*	**1**	**2**	**3**	**4**
γ_i	$-45°$	$45°$	$-45°$	$45°$
l_{x_i}	$-l_x$	l_x	l_x	$-l_x$
l_{y_i}	l_y	l_y	$-l_y$	$-l_y$

The matrix representation, which is the most used method for inverse kinematics calculations, is presented in Equation (7):

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & 1 & -(l_x + l_y) \\ 1 & -1 & -(l_x + l_y) \\ 1 & 1 & (l_x + l_y) \\ 1 & -1 & (l_x + l_y) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} \quad (7)$$

The inverse kinematic Jacobian matrix J of the OMR is expressed as:

$$J = \frac{1}{R} \begin{bmatrix} 1 & 1 & -(l_x + l_y) \\ 1 & -1 & -(l_x + l_y) \\ 1 & 1 & (l_x + l_y) \\ 1 & -1 & (l_x + l_y) \end{bmatrix} \quad (8)$$

In order to obtain the forward kinematic equations, used for calculating the linear and angular speeds of OMR relative to the ground, the deduction starts from Jacobian matrix, which has 4×3 dimensions, it must be used a pseudo inverse matrix J^+ such that $J^+ \cdot J = I_3$, which it is determined with the formula from [27]:

$$J^+ = (J^T \cdot J)^{-1} \cdot J^T \quad (9)$$

The values calculated for OMR with conventional notations from Table 2 for forward kinematics are:

$$J^+ = \frac{R}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ -(l_x + l_y)^{-1} & -(l_x + l_y)^{-1} & (l_x + l_y)^{-1} & (l_x + l_y)^{-1} \end{bmatrix} \quad (10)$$

And the resultant direct kinematics transformation representing linear and angular velocities of the OMR relative to ground is:

$$\begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} = J^+ \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (11)$$

The matrix equations Equations (7) and (11) are essential components of the vehicle controller firmware implementation that is presented in Section 3.3.

3.2. Hardware Architecture

The architectural design diagram of the hardware platform [29] is presented in Figure 4. The components of the architecture are grouped by their functional role: power supply unit, chassis with the mechanical actuators, perception layer (that includes the environment scanning sensors), and the human machine interface (HMI) elements. The remaining interlinking components, that are not part of the marked groups, are part of the control and communication layer.

Furthermore, the control components can be subdivided in two logical levels, in accordance with the control diagram presented in Figure 2: high level control implemented in the ROS nodes running on the central unit controller and the low level control functions implemented in the firmware of the vehicle controller.

The power supply unit is customized in order to provide flexibility between mobility, required during dynamic tests, and autonomy, needed during development and static tests. For this purpose it features a main power switch that allows to select between the internal power source (Lithium-Ion battery pack that includes a battery management system), an external power source and the power off mode.

The chassis is equipped with four Mecanum wheels. Each wheel is actuated using a separated 24 V DC Motor fitted with a quadrature encoder having 500 increments per mechanical rotation and an 18.5:1 ratio planetary gear box speed reduction for traction torque amplification.

The HMI provides system monitoring functions and several local or remote control options, including an emergency stop button. The embedded vehicle controller state (status, control source, wheels' speed references and actual values, power supply voltage) can be directly observed on the organic light emitting diode (o-LED) mini display attached to the control board. The same state information can be viewed remotely using an Android™smart phone application connected over a BlueTooth™ serial port profile (BTSPP) link. In addition to viewing the state information, the Android™application can also be used for manual remote control of the platform and for tuning the wheel's proportional-integrative speed controllers' parameters. As alternative methods for low level remote control are provided a local (wired) Play Station™2 (PS2) compatible joystick and a standard radio remote controlled (RC) 2.4 GHz 6 channel servo receiver for manual wireless remote operation. It is also possible to achieve HMI interaction using the high level control layer through special PC applications that implement remote control ROS nodes.

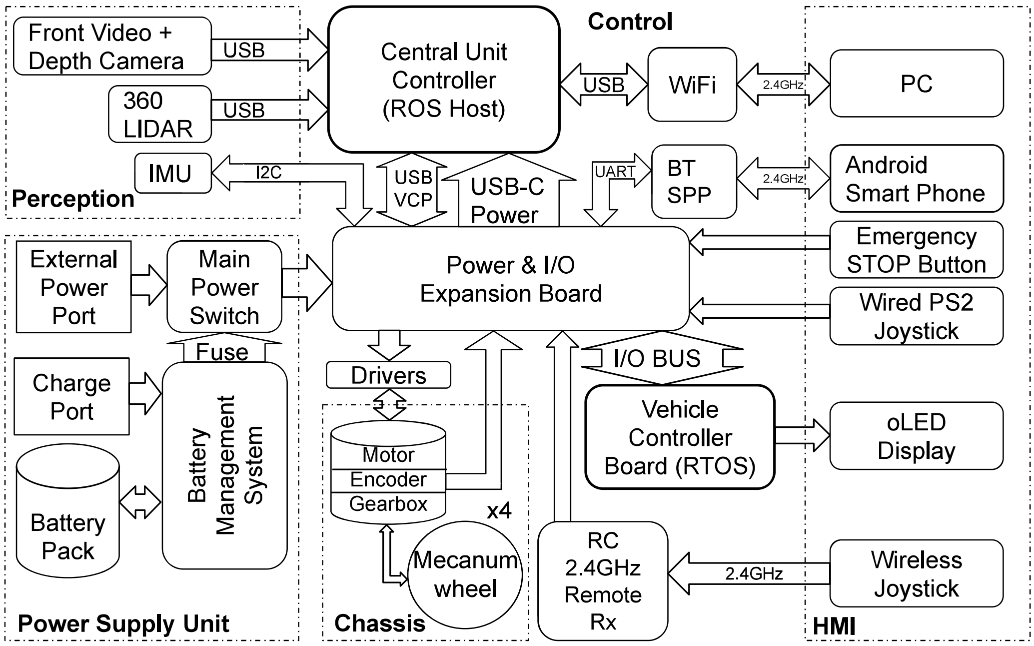


Figure 4. OMR hardware platform modules interconnection diagram.

The perception layer contains the environment scanning sensors: the 360° LiDAR, a video camera with depth perception and an intertial measurement unit (IMU). The LiDAR is used for scanning the general occupancy map of the surroundings. The front video camera with depth perception is used for a more accurate detection of obstacles and objectives on the immediate path of the robot. The IMU provides additional information that can be correlated with the movement commands in order to improve the odometry or to detect abnormal operation.

The central unit controller is a NVIDIA Jetson Nano™ system operated using a GNU/Linux Ubuntu distribution which represents the host for the ROS that runs the nodes dealing with high level control layer. The central unit controller provides access to the perception layer and can dictate the robot's relative motion speeds through a virtual serial COM port (VCP) over the universal serial bus (USB) port on which the vehicle controller is connected, in order to guide it on a specific trajectory. The ROS nodes and the NVIDIA Jetson Nano's Ubuntu operating system can be remotely accessed from a PC using Wi-Fi connection made available through a USB dongle network card.

The vehicle controller is a STM32F103RC micro-controller from ST Microelectronics® which is hosted on an expansion board used for distributing the power and interconnecting with the rest of the attached modules. The central unit controller is powered from the expansion board over dedicated USB-C connector and data connection to the vehicle controller is made through an additional USB connection that tunnels VCP serial connection. The motor drivers, motor encoders, emergency stop button, remote control receiver, bluetooth module and the PS2 joystick are also attached on the expansion board connectors which links them to the vehicle controller.

3.3. Vehicle Controller Firmware

The vehicle controller is designed to be commanded in autonomous mode through the USB VCP connection by the ROS software, which represents the navigation abstraction layer (NAL), running on NVIDIA Jetson Nano™, but also manually through the other HMI channels.

On the vehicle controller it runs a customized implementation of FreeRTOS™, real time operating system (RTOS) distribution maintained by Amazon®. Its main responsibilities are to monitor the command sources (USB VCP, BTSPP, PS2 joystick, RC joystick), arbitrate between them, monitor the motor encoders to obtain the speeds of the wheels, sample the IMU, properly control the motor drivers to obtain the desired speed references, and report system status through different communication channels (VCP, BTSPP, and controller area network (CAN)). This responsibilities are implemented using the following set of tasks running on the RTOS software infrastructure: IMU sampling, PS2 joystick handling, o-LED display management, data reporting, and motion control.

The motion control and data reporting tasks are essential for the implementation of the hardware abstraction layer (HAL) of the OMR. For this reason the firmware includes the parameters of the OMR geometry needed to evaluate the inverse kinematics Equation (7) for controlling the wheel speeds according to the requested relative motion reference speeds, while at the same time the data reporting task is using the same parameters for evaluating the direct kinematics Equation (11) required to obtain the actual OMR relative motion speeds used by the central unit controller for odometry.

3.4. High Level Software and the Control Application of the ROSY Platform

ROS is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. ROS can integrate, run or support any application and also provides: operating system (OS) services, HAL, low-level device control, implementation of commonly-used functionalities, message-passing between different processes and package management. There are many advantages of using ROS, especially due to the fact that it is open-source [30,31].

ROS application is the high level software that allows the developer to integrate the OMR into a HiL process. A ROS application includes individual pieces of software that are integrated into the ROS ecosystem as ROS nodes.

The relationship between nodes running on NVIDIA Jetson Nano is illustrated in Figure 5. The underlying communication of ROS is based on XML-RPC protocol. It allows cross-platform software to make remote calls by sending and receiving messages in XML format.

The main nodes that enable the serial communication and provide information about robot's real-time position while executing a certain task are:

- Matlab navigation ROS node specially designed for the HiL architecture proposed in the paper
- Vehicle Controller serial communication ROS node: enables the serial communication and the data transfer between NVIDIA Jetson Nano and STM32 robot controller. The ROS system receives data sent by the lower level controller and, at the same time, sends instructions to the micro-controller to handle the robot, by specifying the desired orthogonal translation speeds and the rotational velocity.
- LiDAR ROS node "rpLidarNode": enables the LiDAR which communicates with the application controller via a serial port and used, further on, in the ROS software applications.
- Extended Kalman filter (EKF) Odometry ROS node: provides encoder odometer data and IMU data.
- Teleop Keyboard node: used in manual mapping of OMR environment.

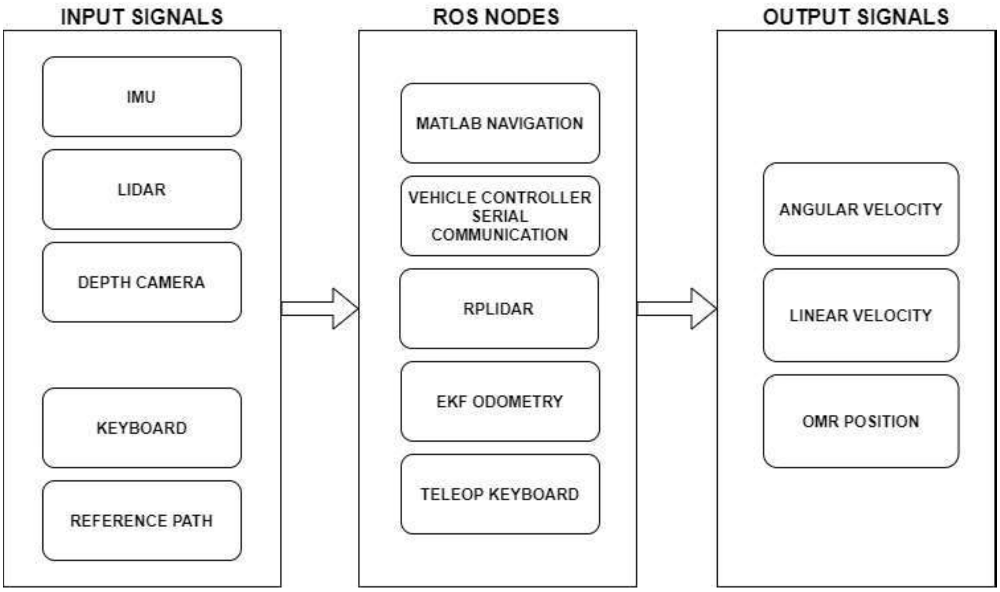


Figure 5. ROS Software Architecture.

4. Real-Time Motion Control Using a ROS Node Implemented in MATLAB

ROS Toolbox provides an interface that allows the connection between Matlab Simulink and ROS ecosystem. Using this facility, there is possible to create a cross-platform network of ROS nodes. The toolbox includes MATLAB functions and Simulink blocks to import, analyze, and play back ROS data recorded in rosbag files [32]. It is also possible to connect to an existing live ROS network to access ROS messages. ROS Toolbox enables the generation of ROS nodes from a Simulink model and also to integrate the simulated ROS node into a ROS network from a physical hardware [30].

The toolbox includes algorithms for collision checking, trajectory generation and forward or inverse kinematics. For mobile robots, such as the OMR, it includes algorithms for mapping, localization, path planning, path following and motion control. The toolbox provides reference examples of common industrial robot applications and it also includes libraries of commercially available industrial robots that can be imported, visualized and simulated. In order to evaluate the performances of the proposed approach, the first goal was to implement a ROS node package using the features that Matlab provides to perform localization of the OMR [33].

4.1. ROS Node Implementation

Figure 6 shows the modules implemented, the main part consists of the *Control* block with the *Trajectory Controller* whose role is to calculate the command for linear velocity and angular speed using two PID type controller in order to reach the destination goal. This command is sent to the next block, being saturated before to be published in the last block throw the topic */cmd_vel*. During navigation through a previously constructed map, the robot behaves as expected with only a few shortfalls. These errors were most likely caused by errors in odometry data.

To generate a route for the robot to move in an optimal way it is needed a map space for a path planner algorithm. In *Waypoint planer*, the coordinates $[x\ y]$ are sent to *pathPoints* as a goal destination. *currentPosition* represents the current position of the robot acquired for *pose* topic and when the goal position was reached the signal *done_cmd* is send for receiving the new position. Because the robot has 60 cm on every side, it is necessary to insert a threshold applied on the edges of the obstacles. This threshold has two effects, one is to create a space bounds in order to avoid collision of the robot with the obstacles and the second is for path planning, where we use a probabilistic roadmap algorithm with the purpose to eliminate areas where it is not necessary to calculate the network graph of the possible path. Also position of the robot on the map is known in real time by subscribe to topic */pose* generated by *wheeltec_robot* node. In the *Perception Subscription* block, position of the robot on the map is known in real time by subscribe to topic */pose* generated by *wheeltec_robot* node. Through the *Planner* block, the *matlab_navigationNode* develop a feasible path from its location to the goal utilizing the probabilistic road map and generates a road path for SLAM algorithm.

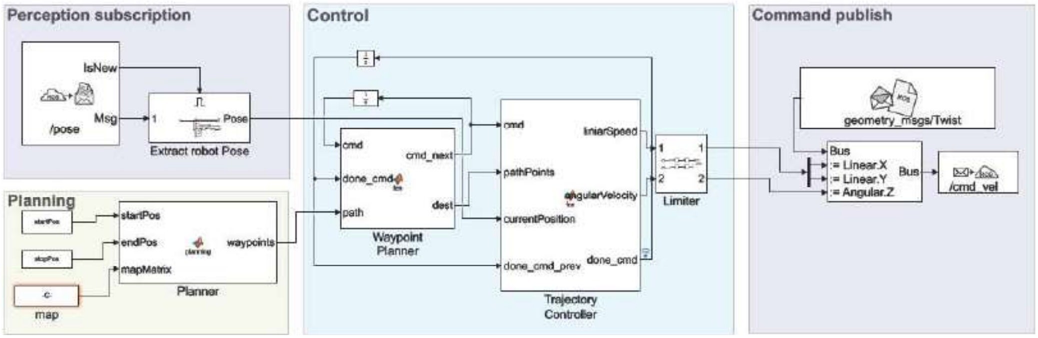


Figure 6. ROS application nodes implemented in Matlab.

4.2. Practical Experiments and Evaluation

This section presents the control performances of the OMR using the proposed approach. Real time experiments are obtained using ROS and Matlab features. The experimental flow has two main parts: (i) test run and data acquisition in Matlab and (ii) data processing and reporting according with the proposed approach used in the Figure 5. The first part, the OMR platform is subject to several scenarios and the data are acquired in real time from IMU and LiDAR sensors. The working environment is a restricted area, the real platform and the environment are illustrated in Figure 7. In the second part, the tests are reported and analysed from the reference tracking performance perspective. The following parameters were monitored on the OMR: raw sensor data (XY movement, linear speed and angular speed) and reference trajectory. The first experiments were performed considering circle and square reference signal, respective, considering reference signals for position x (forward movement) and y (lateral movement). Further, a map of the environment is built based on LiDAR scans, then the OMR navigates inside the map following the reference trajectory.

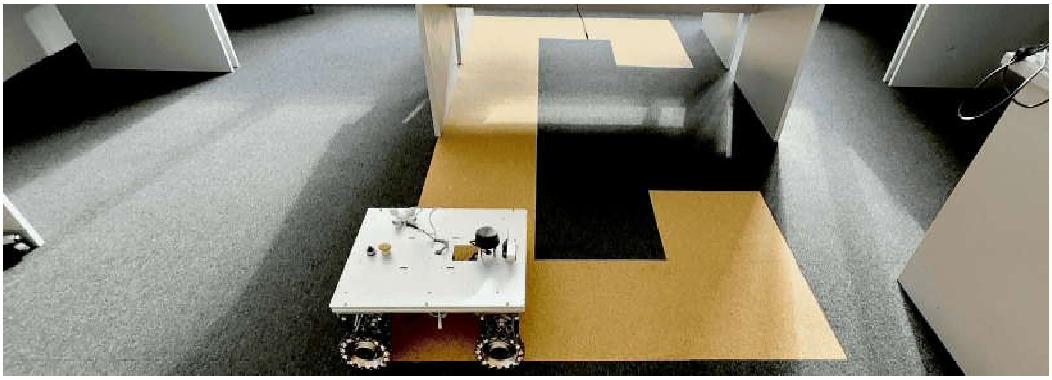


Figure 7. The used OMR hardware platform in its testing area.

The simple motion controller moves the OMR to a new waypoint in two phases: an initial azimuth orientation (if the azimuth differs significantly), followed by a longitudinal translation together with slight azimuth corrections until a vicinity of the destination is reached. Due to the symmetry of the OMR operation, all translations are expected to perform in a similar way, so the longitudinal translation motion was studied for the tracking performance in the square trajectory experiment.

Tests were performed and the platform's relative velocity to ground was recorded using a set of local coordinates (considering x the longitudinal movement axis and y the lateral movement axis), and the angular velocity around the vertical z-axis.

4.2.1. Circular Trajectory

Figure 8 shows with the red line the target circular trajectory with 1 m radius is given as reference and with the black points the actual trajectory of the robot, considering that it starts from the origin of the circle, approaches in a straight line the periphery, then aligns with the closest tangent to the circle and follows the path in the trigonometric direction. Figure 9 illustrates the performances of the OMR, when attempting to follow the circular path. It can be observed that the OMR reaches in the vicinity of the reference points on both x-axis and y-axis, while the control effort is being accomplished by the linear and angular speed controllers.

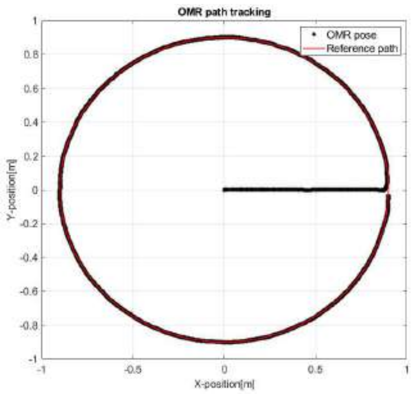


Figure 8. Circle path tracking.

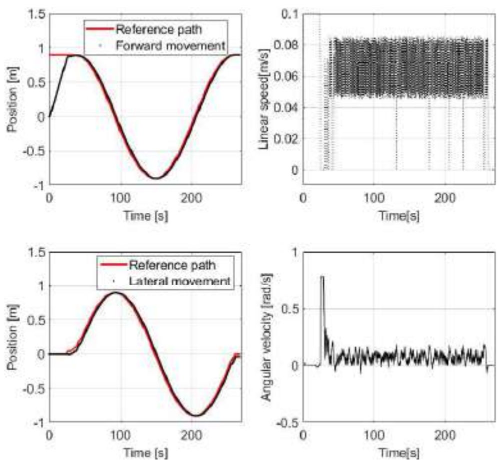


Figure 9. Illustration of control performances for circle scenario.

The linear speed of the platform illustrated in Figure 9 reaches the maximum 0.1 m s^{-1} allowed by the controller when navigating from the origin to the periphery, then it returns to 0 m s^{-1} while the controller rotates the platform tangent to the circle. When the platform orientation becomes tangent, the linear speed controller accelerates towards the next waypoints on the circular path, but it is visible that the linear speed is limited by the controller at about maximum 0.08 m s^{-1} and it oscillates because while following the circular trajectory the orientation of the platform needs continuous realignment, fact that triggers the linear speed reduction to guarantee the proper reference tracking.

4.2.2. Square Trajectory

Figure 10 illustrates with the red line a square reference trajectory with each side measuring 1 m, which is defined by only four way-points representing the corners, while with the black points it is marked the actual trajectory of the OMR in-between them. The tracking performances considering that the robot starts from the origin point, reaches the goal points and follows all four sides in about 2.5 min, as Figure 11 shows.

In Figure 11 it is also easily visible how the linear speed controller and angular speed controller interact: when the desired motion direction differs significantly from the actual direction the linear speed reference is switched to zero in order for the angular speed controller to properly adjust the direction towards the target.

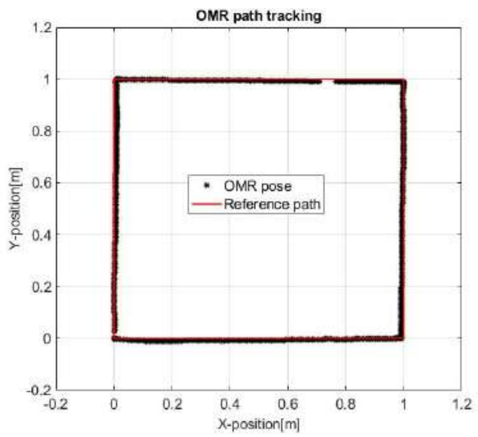


Figure 10. Square path tracking.

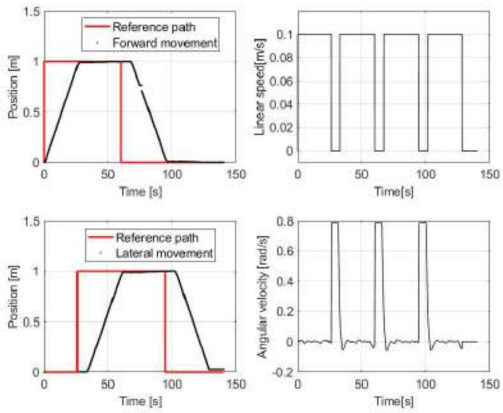


Figure 11. Illustration of control performances for square scenario.

4.2.3. Navigation Using the Map Obtained with SLAM

Autonomous map building focuses on the widely used SLAM approach. The platform keeps track of its motion using odometry while navigating in the known environment as illustrated in Figure 12. The experiments build, in the first phase, the environment map based on LiDAR scans, while in the second phase, robot navigates inside the map which is represented in Figure 13. In order to perform the real time tests, a ROS node package was used, based on the features that Matlab provides to perform localization and mapping.

The approach was to create a LiDAR SLAM object and set the map resolution and max LiDAR range. The maximum LiDAR range was set to 4 m, while it is smaller than the maximum scan range, which is 12 m, as the laser reading are less accurate near maximum range. The grid map has a resolution of 2 cm per division. However, odometry uncertainty confuses the robot about its current position.

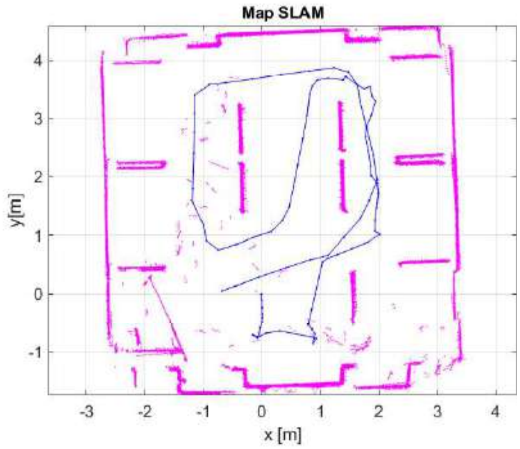


Figure 12. Scene reconstruction by plotting the scans and poses tracked during an initial manual navigation.

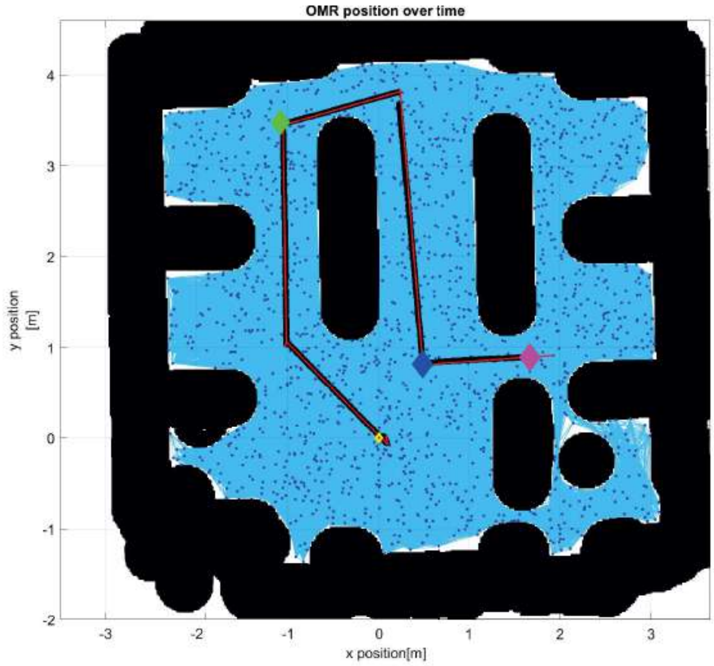


Figure 13. Navigation using the facilities from probabilistic road-map (yellow point is the initial position, the green point is the first destination which produced a secondary way-point to avoid the obstacle, the blue point is the second destination for which an additional way-point was created, and the magenta point is the final destination.

To generate a route for the robot to move in an optimal way, it is needed a map space for a path planner algorithm. Because the robot has 60 cm on every side, it is necessary to insert a threshold applied on the edges of the obstacles, an inflation of the size of the obstacles which can be observed when comparing Figure 12 with Figure 13. This threshold has two effects, one is to create a space bounds in order to avoid collision of the robot with the obstacles and second is for path planner, where a probabilistic road-map algorithm is used, as is illustrated in Figure 13. In order to have a realistic static map of the indoor environment where only the allowed and forbidden areas are represented, the occupational map was binarized and saved in the robot's memory, for a later use in the navigation node.

5. Conclusions

The strategic control level of the OMR and implementation presented in this article use a HiL approach designed for integration in the Matlab environment, in which it will be developed the strategy for managing the robotic structure. The strategy is expected to be complex because it depends on a large number of factors, among which the OMR's architecture, the decision algorithms that take into account the activity to be performed, the necessary operating accuracy, the operating speed and the energy consumption optimization. At this level, the complex action can be decomposed into elementary operations that can be transmitted to be executed by the lower hierarchical level represented by the vehicle controller.

MATLAB is recognized as a very powerful tool for testing control systems satisfying the need for rapid control prototyping. Facilities offered by the Robotics Systems Toolbox has contributed to increasing interoperability with mobile robotic systems that prefer to use the ROS environment more and more.

The preliminary experimental results from laboratory for testing of the four Mecanum wheels mobile platform was done from three perspectives. The first is related to the validation of tracking predefined trajectories, usually used in the testing of mobile robots: circle and square. The second perspective refers to building a map of the environment based on LiDAR scans and the position using SLAM algorithm, procedure used partly in commissioning of industrial mobile robot. The last perspective is the performing of navigation between points defined in the map which qualified the mobile platform to be equipped with conveying system and used in logistic environment as it is the aim of ROSY-Logistic project.

Future research will be directed towards the development of reactive control algorithms in order to cope with the dynamic changes that may occur in the workspace captured with the static map inclusively equipping the platform with other perception systems to obtain accuracy of navigation.

Author Contributions: Conceptualization, C.D., C.B. and T.P.; methodology, C.D. and C.B.; software, T.P., R.S. and P.G.; validation, C.B., C.D. and T.P.; formal analysis, T.P. and C.B.; investigation, C.D., R.S. and T.P.; data curation, C.B.; writing—original draft preparation, C.D.; writing—review and editing, C.B., T.P. and P.G.; project administration, C.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI – UEFISCDI, project no. PN-III-P2-2.1-PTE-2019-0731, contract no. 19/2020, within PNCDI III, second stages.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

HiL	hardware in the loop
AGV	automated guided vehicle
AMR	autonomous mobile robot
OMR	omnidirectional mobile robot
UGV	unmanned ground vehicle
UAV	unmanned aerial vehicle
ROS	Robot Operating System
HMI	human machine interface
o-LED	organic light emitting diode

BTSPP	BlueTooth™ serial port profile
PS2	Play Station™2
IMU	inertial measurement unit
LiDAR	light based detection and ranging system
USB	universal serial bus
VCP	virtual serial COM port
NAL	navigation abstraction layer
HAL	hardware abstraction layer
RTOS	real time operating system
CAN	controller area network
SLAM	simultaneous localization and mapping
RC	radio remote controlled
OS	operating system
API	application programming interface
EKF	extended Kalman filter

References

1. Wang, C.; Du, D. Research on logistics autonomous mobile robot system. In Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China, 7–10 August 2016; pp. 275–280. [CrossRef]
2. Fragapane, G.; de Koster, R.; Sgarbossa, F.; Strandhagen, J.O. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *Eur. J. Oper. Res.* **2021**. [CrossRef]
3. Semenas, R.; Bausys, R. Modelling of Autonomous Search and Rescue Missions by Interval-Valued Neutrosophic WASPAS Framework. *Symmetry* **2020**, *12*, 162. [CrossRef]
4. Chatila, R.; Lacroix, S.; Siméon, T.; Herrb, M. Planetary Exploration by a Mobile Robot: Mission Teleprogramming and Autonomous Navigation. *Auton. Robots* **1995**, *2*, 333–344. [CrossRef]
5. Kriegel, J.; Rissbacher, C.; Reckwitz, L.; Tuttle-Weidinger, L. The requirements and applications of autonomous mobile robotics (AMR) in hospitals from the perspective of nursing officers. *Int. J. Healthc. Manag.* **2021**, 1–7. [CrossRef]
6. Boloz, L.; Bialy, W. Automation and Robotization of Underground Mining in Poland. *Appl. Sci.* **2020**, *10*, 7221. [CrossRef]
7. Williams, A. *Autonomous Systems: Issues for Defence Policymakers*; Nato Comunications and Informations Agency: Norfolk, VA, USA, 2015.
8. Gonzalez-de Santos, P.; Fernández, R.; Sepúlveda, D.; Navas, E.; Emmi, L.; Armada, M. Field Robots for Intelligent Farms—Inhering Features from Industry. *Agronomy* **2020**, *10*, 1638. [CrossRef]
9. Sahin, H.; Guvenc, L. Household robotics—Autonomous devices for vacuuming and lawn mowing. *Control Syst. IEEE* **2007**, *27*, 20–96. [CrossRef]
10. Rubio, F.; Valero, F.; Llopis-Albert, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *Int. J. Adv. Robotic Syst.* **2019**, *16*. [CrossRef]
11. Angerer, S.; Strassmair, C.; Staehr, M.; Roettenbacher, M.; Robertson, N. Give me a hand—The potential of mobile assistive robots in automotive logistics and assembly applications. In Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications (TEPRA2012), Woburn, MA, USA, 23–24 April 2012; IEEE Computer Society: Woburn, MA, USA, 2012. [CrossRef]
12. Alatise, M.; Hancke, G. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access* **2020**. [CrossRef]
13. Abd Mutalib, M.A.; Azlan, N.Z. Prototype development of mecanum wheels mobile robot: A review. *Appl. Res. Smart Technol. (ARSTech)* **2020**, *1*, 71–82. [CrossRef]
14. Wang, C.; Liu, X.; Yang, X.; Hu, F.; Jiang, A.; Yang, C. Trajectory Tracking of an Omni-Directional Wheeled Mobile Robot Using a Model Predictive Control Strategy. *Appl. Sci.* **2018**, *8*, 231. [CrossRef]
15. Muir, P.; Neuman, C. Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In Proceedings of the 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, USA, 31 March–3 April 1987; Volume 4, pp. 1772–1778. [CrossRef]
16. Zijie, N.; Qiang, L.; Yonjie, C.; Zhijun, S. Fuzzy Control Strategy for Course Correction of Omnidirectional Mobile Robot. *Int. J. Control Autom. Syst.* **2019**, *17*, 2354–2364. [CrossRef]
17. Azizi, M.R.; Rastegarpanah, A.; Stolkin, R. Motion Planning and Control of an Omnidirectional Mobile Robot in Dynamic Environments. *Robotics* **2021**, *10*, 48. [CrossRef]
18. Indiveri, G.; Nuchter, A.; Lingemann, K. High Speed Differential Drive Mobile Robot Path Following Control With Bounded Wheel Speed Commands. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 2202–2207. [CrossRef]

19. Soares, J.; Fischer Abati, G.; Duarte Lima, G.; Machado de Souza Junior, C.; Meggiolaro, M. Project and Development of a Mecanum-wheeled Robot for Autonomous Navigation Tasks. In Proceedings of the XVIII International Symposium on Dynamic Problems of Mechanics (DINAME 2019), Búzios, 10–15 March 2019. [CrossRef]
20. Li, T.; Zhang, F.; Gao, X.; Xu, H.; Ji, S. *The Control System Design of a Omni-Directional Mobile Logistics Sorting Vehicle Based on stm32*; Series D, Mechanical Engineering; Scientific Bulletin—"Politehnica" University of Bucharest: Bucharest, Romania, 2020; Volume 82.
21. Mu, F.; Liu, C. Design and Research of Intelligent Logistics Robot based on STM32. *Recent Adv. Electr. Electron. Eng.* **2021**, *14*, 44–51. [CrossRef]
22. Uriarte, C.; Kunaschk, S. Omnidirectional Conveyor System Module, Modular Omnidirectional Conveyor System and Omnidirectional Conveyor System. German Patent DE102012014181A1, 23 January 2014.
23. Dosoftei, C.C.; Lupu, A.; Pascal, C.M. A new approach to create a realistic virtual model of a cylindrical robot using Automation Studio. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *591*, 012078. [CrossRef]
24. Rassõlkin, A.; Sell, R.; Leier, M. Development case study of the first estonian self-driving car, iseauto. *Electr. Control Commun. Eng.* **2018**, *14*, 81–88. [CrossRef]
25. Gracia, L.; Tornero, J. Kinematic control of wheeled mobile robots. *Latin Am. Appl. Res.* **2008**, *38*, 7–16.
26. Dosoftei, C.; Horga, V.; Doroftei, I.; Popovici, T.; Custura, S. Simplified Mecanum Wheel Modelling using a Reduced Omni Wheel Model for Dynamic Simulation of an Omnidirectional Mobile Robot. In Proceedings of the 2020 International Conference and Exposition on Electrical And Power Engineering (EPE), Iasi, Romania, 22–23 October 2020; pp. 721–726. [CrossRef]
27. Maulana, E.; Muslim, M.A.; Hendrayawan, V. Inverse kinematic implementation of four-wheels mecanum drive mobile robot using stepper motors. In Proceedings of the 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 20–21 May 2015; pp. 51–56. [CrossRef]
28. Li, Y.; Dai, S.; Zhao, L.; Yan, X.; Shi, Y. Topological Design Methods for Mecanum Wheel Configurations of an Omnidirectional Mobile Robot. *Symmetry* **2019**, *11*, 1238. [CrossRef]
29. Company, W. Wheeltec Company Website. Available online: https://wheeltec.net (accessed on 29 April 2021).
30. Feng, Y.; Ding, C.; Li, X.; Zhao, X. Integrating Mecanum wheeled omni-directional mobile robots in ROS. In Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), Qingdao, China, 3–7 December 2016; pp. 643–648. [CrossRef]
31. Quigley, M.; Gerkey, B.; Conley, K.; Faust, J.; Foote, T.; Leibs, J.; Berger, E.; Wheeler, R.; Ng, A. ROS: An Open-Source Robot Operating System. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009.
32. MathWorks. Get Started with ROS. Available online: https://www.mathworks.com/help/ros/ug/get-started-with-ros.html (accessed on 28 April 2012).
33. Corke, P. Integrating ROS and MATLAB [ROS Topics]. *IEEE Robot. Autom. Mag.* **2015**, *22*, 18–20. [CrossRef]

Comparative Analysis of Advanced Cooperative Adaptive Cruise Control Algorithms for Vehicular Cyber Physical Systems

A. Tiganasu, C. Lazar, C. F. Caruntu, C. Dosoftei

"Gheorghe Asachi" Technical University of Iasi, Romania
e-mail: {tiganasu.alexandru, clazar, caruntuc, cdosoftei}@ac.tuiasi.ro

Abstract: Nowadays, the number of vehicles on the roads is progressively increasing, this leading to a saturation of the traffic. In order to reduce the travel times and to increase the drivers' comfort, a series of Advanced Driver Assistance Systems (ADAS) that assist the drivers in cities or on highways were developed. The need of increasing the roads' capacities conducted to a concept named vehicle platooning in which the vehicles are grouped in convoys and they move as a single entity with the same velocity on the same lane. Beside simple radar devices that measure the distance to the vehicle in front, the studied platoons contain followers equipped with wireless communication systems (WCS). This feature offers to the followers the possibility to anticipate the behaviour of their predecessor considering that they receive the velocity or acceleration from the front vehicle through WCS. This type of vehicle platoon can be viewed as being composed of two layers: a virtual one, called cyber plane, consisting of the communication messages themselves and a real one, called physical plane, represented by the vehicles in the platoon. The paper presents a comparative analysis of two cyber-physical systems implemented with dedicated algorithms from literature (generalized predictive controller (GPC) and linear quadratic regulator (LQR)) at which were added a series of doctoral researches that cover the case study related to vehicle platooning for city and highway travelling. Each considered platoon is a hybrid system composed of a cruise control (CC) system for the leader and cooperative adaptive cruise control (CACC) systems for the followers. All proposed algorithms were simulated in MATLAB/Simulink and the results were analyzed providing some conclusions related to their efficiency.

Keywords: cooperative control, inter-vehicle communications, cyber-physical system, intelligent driver assistance, connected vehicles.

1. INTRODUCTION

Considering that the worldwide population is continuously growing the cities become more and more crowded and the climate is changing faster due to the pollution, there is an urgent need of reducing the negative effects of the present transportation systems. To solve these problems, many engineers from the automotive industry and researchers from the academia are working hard to find the most efficient solutions. These solutions can be from implementing smart control systems at vehicle level (advanced driver assistance systems) to creating vehicle networks at traffic level, using wireless communications and implying the infrastructure itself.

A concept that addresses the above-mentioned solutions is represented by the Intelligent Transportation Systems (ITS) designed to improve the urban and extra-urban mobility. In this category could be included the four well-known transport networks: road, rail, air and water. ITS involve vehicles, drivers, passengers and road operators that interact with each other and, at the same time, with the environment. Due to the fact that most of the accidents happen in the case of road transport the main focus is on ITS for this type of mobility. To successfully operate, data must be sent accurately and in timely manner and must be correctly received by the corresponding recipient that knows how to interpret it (Williams, 2008). The great potential offered by ITS technologies must be focused mainly on the safety needs than on comfort, considering that the human factors are still present with an important influence on the traffic flow. All possible human errors must be treated by these smart systems (Regan et al., 2001). ITS consists of different transportation systems, such as advanced traveler information system, advanced traffic management system, advanced transit system, and so on (Zhao et al., 2018).

The complexity of road traffic dynamics is characterized by a series of properties as non-linearity, non-uniformity and adaptability. The key reasons of complexity in road traffic are individual driver behaviour and unpredictable movement choices. The traffic is a complex phenomenon being described by the interaction of heterogeneous road users like vehicles, pedestrians, and cyclists (Riaz and Niazi, 2016). The high complexity of road traffic can conduct to undesired events as collisions considering that the human drivers are implied in the car travel on the roads. Human drivers are the major reason of accidents due to various careless activities such as talking on phone or texting.

The cyber-physical systems (CPS) are structures with a tight interaction between physical models and computational (cyber) units and a good collaboration between software engineering, control strategy, embedded systems and real-time systems. They are allowing individual entities to work together in order to form complex systems with new capacities in an efficient and safe way. CPS technology can be applied in various fields, offering a lot of opportunities: critical infrastructure control, safe and efficient transport, environmental control, medical devices (Lungoci et al., 2015), social networking, gaming, agriculture and alternative energy (Sanislav and Miclea, 2012).

In order to obtain an efficient way of vehicle traveling from multiple aspects as pollution and fuel consumption reduction or increasing of roads capacities, many times the grouping of the vehicles in platoons is considered an appropriate solution. A platoon is a complex physical system in which the drivers must act cooperatively to control and manage it, including formation, merging, splitting or maintenance. An example from the literature of a vehicle platoon strategy is the one proposed in (Wei et al., 2017) that consists of a bidirectional platoon control system composed by n vehicles that takes into account the uncertainty in the engine time, the actuator delay and the actuator saturation. All of them are able to measure the relative distance and velocity with respect to the nearest neighbours (in front and behind them) using on-board sensors. The authors tested the developed algorithms by means of simulation tools and performing some experiments with 5 cars equipped with radio-controlled Arduino hardware.

To reduce the drivers' effort spent in the driving process a series of new technologies were developed during time as the adaptive cruise control system (ACC) that is able to measure the distance to the vehicle in front and autonomously maintain it to a safety value using sensors and actuators. These technologies applied individually can be used to manage the traveling in platoons also, and if modern wireless communication systems are considered a type of ITS is obtained (Jia et al., 2016). Vehicles with communication capability can dynamically form a mobile wireless network on a road, called vehicular ad hoc network (VANET), which can offer two types of wireless communications: vehicle to vehicle (V2V) communication and vehicle to infrastructure (V2I) communication. A platoon-based vehicular cyber-physical system (VCPS) is the synergistic integration of networking, computation and physical processes that are working together to assure both safety and comfort to the driver and passengers (Patil et al., 2018). The VCPS offers assistance to the humans being designed not including the driver behaviour characteristics. Autonomous vehicles (AVs), adaptive cruise control (ACC), lane departure warning, and early collision avoidance systems are different types of VCPS. A more innovative application presented in (Abid et al., 2011) is the combination of VCPS and cloud computing paradigm that forms a V-Cloud architecture. In VCPS all vehicles communicate via vehicular networking and are driven in a platoon-based pattern, with a closed feedback loop between the cyber process and physical process. An example of such a VCPS is the cooperative adaptive cruise control (CACC) system that has the capability to maintain a desired inter-vehicle or inter-platoon distance using the technology from ACC combined with V2V communications. The CACC system can be modelled as a networked control system in which both platoon mobility and VANET are coupled. A negative impact on control performance can exist if the uncertainties of practical VANET, as packet loss and transmission delay, are considered. Also, some possible network attacks as jamming, V2V data injection or sensor manipulation can be taken into account in the design phase of a CACC system (van der Heijden et al., 2017). The performance of a platoon-based VCPS is jointly determined by both networking process and control process, which closely combines communication, computation and control together.

During the last years, many researchers studied, implemented and tried to improve the first versions of the CACC system. In the literature, many control strategies for CACC vehicles were found. In (Wang et al., 2018), a series of control algorithms that have the purpose of CACC implementation are presented: the model predictive control (MPC) (Varada, 2017) which refers to a class of algorithms that utilize an explicit process model to predict the future response of a plant being formulated in the state space for a single-agent system; the distributed consensus control which implies many agents that cooperatively reach an agreement with respect to a certain interest that depends on the states of all agents; the optimal control strategy design that can be equivalent with a structured convex optimization problem and can consider nonlinearity and constraints in contrast to the consensus control approach. The authors of (Wei et al., 2018) propose a supervised reinforcement learning (SRL) algorithm for the CACC problem that is an enhanced version of the strategy described in (Gao et al., 2019) and (Desjardins and Chaib-draa, 2011). The fundamental of this algorithm is learning an optimal policy as a mapping from states to actions that optimizes some performance criterion. The fact that the driver is missing from the learning process of the actions leads to the design of a supervisor that must provide hints to the agents (vehicles) about which action may or may not perform for a specific state. In (Lu et al., 2002), a sliding mode controller is used to treat the case of CACC systems presenting two options for the sliding surface computation, while (Öncü et al., 2014) proposes a feedback/feedforward control structure that includes a proportional-derivative (PD) controller together with a cooperative element that receives the acceleration of the preceding vehicle and processes it in order to reduce its eventual negative effect. A range-estimation algorithm is described in (Ward et al., 2019) and has as main function to combine the low frequency estimated GPS (global positioning system) position with the higher frequency of radar measurements. By using both measurements, the range estimator provides a high update rate with high accuracy. This algorithm is mainly used in heavy-duty vehicles platoons. In (Flores and Milanés, 2018), a specific fractional-order control technique is used to develop the CACC controller. This is the generalization of the proportional-integral-derivative (PID) controller that takes as a basis its mathematical form stating that the integral and derivative operators are not necessarily of first order. A control system based on the attenuation of the acceleration diffusion using inter-vehicle communication (IVC) is proposed in (Omae et al., 2013). This control method attenuates the acceleration variability using signals obtained through IVC. Another CACC control strategy from literature is the one presented in (Wu et al., 2019) that is based on an adaptive Kalman filter together with a computation method of the preceding vehicle measurement vector.

This paper has as purpose a comparative analysis of two advanced cooperative control algorithms: a generalized predictive feedback/feedforward controller (GPC) developed in (Tiganasu et al., 2017) and a linear quadratic (LQR) control strategy proposed in (Lazar et al., 2018). These methods provided promising simulation results on their application on a simulated CACC homogeneous platoon of vehicles. Firstly, a summary of the controllers' design is presented and after that the simulation results are described. A MATLAB/Simulink simulator was implemented for each of the methods using as much as possible the same parameters. For a better comparison between these two cooperative control strategies, besides the signals provided by the simulation, some key performance indicators were used to highlight the algorithms' efficiency.

The rest of the paper is organized as follows. In Section 2, a VCPS-oriented platoon organization and two control architectures are presented, while Section 3 is dedicated to the cooperative adaptive cruise control algorithms. Section 4 presents a series of results obtained through simulation. The paper ends with the conclusions that are found in the dedicated section.

2. PLATOON BASED VCPS

In this chapter, two ways of representing a vehicle platoon are briefly presented. In the first one, the convoy is viewed from a macroscopic level, the interactions between vehicles being important in the platoon's movement as a single entity. The second representation is from microscopic level in which the control of each individual vehicle influences the overall behaviour of the platoon. These structures are further used as templates for the simulation of a homogeneous vehicle platoon.

2.1. CPS-oriented platoon organization

The CPS-oriented design of a vehicle platoon is useful to understand how the two planes are combined to obtain a CACC with an in-chain communication system. In Fig. 1 such a representation is depicted. In this case, only V2V communications are considered to fulfil the task of transmitting the velocity to the next vehicle introduced as a measurable disturbance in the system of the successor.

The scope of this signal transmission is that the controlled vehicle can anticipate the behaviour of its predecessor. The negative effect of this disturbance can be compensated if some feedforward mechanisms are implemented.

The CACC system helps the controlled vehicles to have a more efficient movement in a platoon than in the case of using only ACC. The cooperative element of the CACC is represented by wireless communications. The platoons in which the CACC vehicles are grouped can be called cyber-physical systems (CPSs) (Tiganasu et al., 2017). In these vehicle bundles, the physical plane includes all the vehicles in the platoon, the followers being equipped with radar/lidar devices, and the cyber plane is composed of all wireless message transmissions from one vehicle to another.

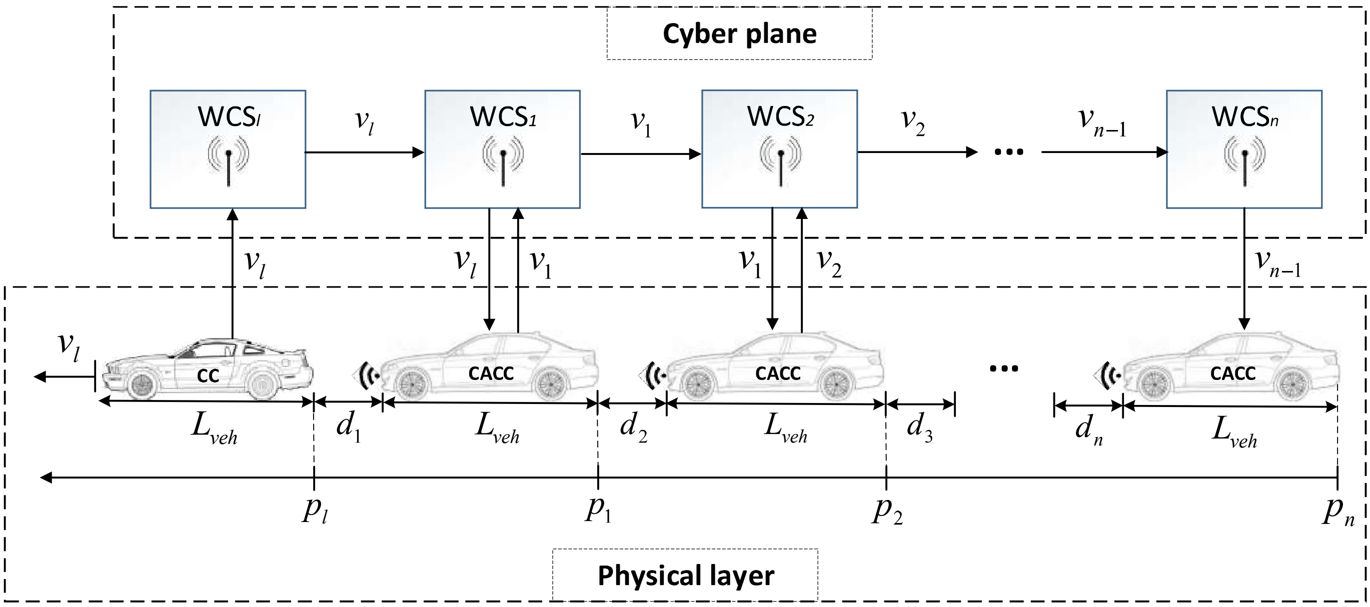


Fig. 1. Structure of a CPS vehicle platoon.

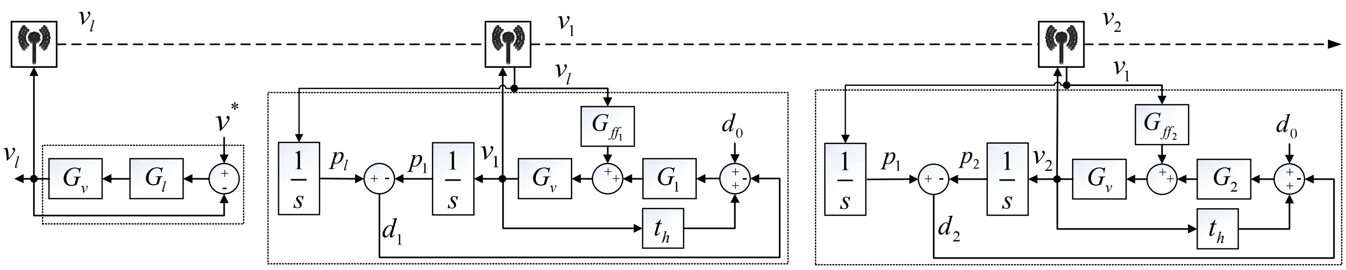


Fig. 2. CACC system architecture.

In Fig. 2, the CACC system architecture is illustrated. In this structure the leader is characterized by a CC system having as speed reference v^*. The followers are consisting in two main components: feedback (G_i) and feedforward (G_{ffi}) controllers. The feedback controllers are included in an ACC structure having the objective to control the distance d_i between vehicles and the feedforward controllers are designed to reject the disturbance introduced by the front vehicle speed.

3. CACC ADVANCED CONTROL ALGORITHMS

The follower vehicles in a platoon must be controlled by a CACC system to be able to maintain the same distance between them and their predecessors using as sensing device a radar and as cooperative element a wireless communication system.

Two advanced control algorithms were designed for controlling the following vehicle longitudinal motion: the predictive control based on the GPC algorithm with a feedforward component (Tiganasu et al., 2017) and the optimal control based on the LQR algorithm (Lazar et al., 2018).

Both LQR and GPC algorithms contain two components:

- a feedback one that is dedicated to the physical vehicle itself embedded in an ACC system;
- a feedforward one for compensating the negative effect of the disturbance introduced by the front vehicle velocity received through the cyber plane.

3.1. Car Following Model

For both advanced control algorithms, the linearized model for the vehicle longitudinal dynamics from (Ulsoy et al., 2012) was used, neglecting the disturbances introduced by the rolling-resistance force and the aerodynamic force and considering a zero-slope road:

$$V_i(s) = G_v(s)U_i(s) = \frac{K_v}{s\tau_v + 1}U_i(s) \tag{1}$$

where the input $U_i(s)$ is the Laplace transform of the traction force, K_v the vehicle gain, τ_v the vehicle time constant, and the output $V_i(s)$ is the Laplace transform of the i^{th} vehicle speed.

The model from (1) was developed using the linearized form of the longitudinal motion equation (Ulsoy et al., 2012):

$$m\frac{dv}{dt} = F_x - mg\sin\theta - fmg\cos\theta - 0.5\rho A_{rc}C_d\left(v + v_w\right)^2 \tag{2}$$

where m is the vehicle mass, F_x is the traction force, v is the vehicle velocity, v_w is the wind speed, g is the gravitational acceleration, θ is the road slope, ρ is the air density, C_d is the drag coefficient, f is the rolling resistance coefficient and A_{rc} is the vehicle frontal area.

Being a homogeneous vehicle platoon system, each vehicle dynamics is described by the transfer function $G_v(s)$. Using (1) and taking into account the position p_i of the i^{th} vehicle, the next two models were obtained, for GPC, the transfer function:

$$P_i(s) = \frac{1}{s}V_i(s) = \frac{1}{s}G_v(s)U_i(s) = \frac{K_v}{s(s\tau_v + 1)}U_i(s) \tag{3}$$

and, respectively, for LQR, the state-space model:

$$\begin{cases} \dot{p}_i = v_i \\ \dot{v}_i = -\dfrac{1}{\tau_v}v_i + \dfrac{K_v}{\tau_v}u_i \end{cases} \tag{4}$$

The distance di between following vehicles is measured with a radar/lidar sensor, resulting:

$$d_i = p_{i-1} - L_{veh} - p_i \tag{5}$$

where p_{i-1} and p_i are the positions of the predecessor and the follower, respectively, both being reported at the rearmost point of these vehicles according to Fig. 1, and L_{veh} is the length of vehicle i.

Deriving equation (5) and using the second state equation from (4), the follower vehicle model for LQR was obtained:

$$\begin{cases} \dot{d}_i = -v_i + v_{i-1} \\ \dot{v}_i = -\dfrac{1}{\tau_v}v_i + \dfrac{K_v}{\tau_v}u_i \end{cases} \tag{6}$$

and applying Laplace transform to the first state equation from (6) and using relation (3), the follower vehicle model for GPC design was found:

$$D_i(s) = -\frac{1}{s}G_v(s)U_i(s) + \frac{1}{s}V_{i-1}(s) \tag{7}$$

where $D_i(s)$ is the Laplace transform of the inter-vehicle distance, and $V_{i-1}(s)$ is the Laplace transform of the velocity v_{i-1}.

The preceding vehicle introduces a disturbance through its position $P_{i-1}(s) = \frac{1}{s}V_{i-1}(s)$ which becomes a measurable one by communicating the speed value of the previous vehicle via the wireless communication system. The negative effect of this disturbance will be reduced by a feedforward controller.

All of the CACC vehicles in a platoon have the same objective, i.e., to follow their leading vehicle with a certain distance, which is the safety inter-vehicle distance determined by the spacing policy. The velocity dependent spacing policy was chosen, which determines the desired inter-vehicle distance d_i^* based on vehicle velocity (Dey et al., 2015):

$$d_i^* = d_0 + t_h v_i, \tag{8}$$

where d_0 is the standstill distance and t_h is the time-headway.

The available measurement data from the radar/lidar sensor are used in a feedback setup by an ACC controller.

3.2. Predictive Feedback/Feedforward Control

The design of the generalized predictive control algorithm (Tiganasu et al., 2017) is done starting from the discrete form of the follower vehicle model given in equation (7):

$$d_i(k) = -G_I(z^{-1})G_v(z^{-1})u_i(k) + G_I(z^{-1})v_{i-1}(k) \tag{9}$$

where $G_I(z^{-1}) = T_s z^{-1}/(1-z^{-1})$ is the transfer function of the discrete-time integrator and

$$G_v(z^{-1}) = \frac{b_1 z^{-1}}{1+a_1 z^{-1}} = \frac{\overline{B}(z^{-1})}{\overline{A}(z^{-1})} \tag{10}$$

is the discrete form of the vehicle model from (1).

For each follower vehicle, the CARIMA model can be developed from equations (9) and (10):

$$A(z^{-1})d_i(k) = B(z^{-1})u_i(k-1) + P(z^{-1})v_{i-1}(k) + \frac{e(k)}{\Delta(z^{-1})} \tag{11}$$

with: $A(z^{-1}) = (1-z^{-1})\overline{A}(z^{-1})$, $B(z^{-1}) = -T_s\overline{B}(z^{-1})$ and $P(z^{-1}) = z^{-1}T_s\overline{A}(z^{-1})$, where T_s is the sampling period and $e(k)$ is a zero mean white noise.

Equation (11) can be re-written as follows:

$$\tilde{A}(z^{-1})d_i(k) = B(z^{-1})\Delta u_i(k-1) + P(z^{-1})\Delta v_{i-1}(k) + e(k) \tag{12}$$

where $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$ with $\Delta(z^{-1}) = 1 - z^{-1}$.

Using (12), the following j-step-ahead predictor was derived similarly to (Camacho and Bordóns, 2007):

$$\begin{aligned}\hat{d}_i(k+j\,|\,k) &= G_j(z^{-1})\Delta u_i(k+j-1) + G_j'(z^{-1})\Delta v_{i-1}(k+j)\\ &+ H_j(z^{-1})\Delta u_i(k-1) + H_j'(z^{-1})\Delta v_{i-1}(k)\\ &+ F_j(z^{-1})d_i(k)\end{aligned} \tag{13}$$

based on the Diophantine equations:

$$\begin{cases}1 = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_j(z^{-1})\\ E_j(z^{-1})B(z^{-1}) = G_j(z^{-1}) + z^{-j}H_j(z^{-1})\\ E_j(z^{-1})P(z^{-1}) = G_j'(z^{-1}) + z^{-j}H_j'(z^{-1})\end{cases} \tag{14}$$

Considering the set of the j-step-ahead predictors for $j = \overline{1,p}$ in equation (13), where p is the prediction horizon, the predictor matrix form resulted:

$$\hat{\mathbf{d}}_i = \mathbf{G}\mathbf{u}_i + \mathbf{G}'\mathbf{v}_{i-1} + \mathbf{H}'\mathbf{v} + \mathbf{H}\Delta u_i(k-1) + \mathbf{F}d_i(k) \tag{15}$$

where the matrices $\mathbf{G}$, $\mathbf{G}'$, $\mathbf{H}$, $\mathbf{H}'$ and $\mathbf{F}$, given in (Tiganasu et al., 2017), are composed of the coefficients of the polynomials from the predictor's expression (13).

The last three terms in (15) depend on the past only and their sum represents the free response $\mathbf{f}$. The term $\mathbf{G}'\mathbf{v}_{i-1}$ is considered equal to zero, due to very small values of $\Delta v(k+j|k)$ over the prediction horizon p.

Taking into account that $\mathbf{G}'\mathbf{v}_{i-1} = 0$, yields the predictor:

$$\hat{\mathbf{d}}_i = \mathbf{G}\mathbf{u}_i + \mathbf{f} \tag{16}$$

The optimal future control sequence is obtained by minimizing the cost function:

$$J = (\hat{\mathbf{d}}_i - \mathbf{w}_i)^T(\hat{\mathbf{d}}_i - \mathbf{w}_i) + \lambda \mathbf{u}_i^T\mathbf{u}_i = \frac{1}{2}\mathbf{u}_i^T\mathbf{M}\mathbf{u}_i + \mathbf{b}^T\mathbf{u}_i + \mathbf{f}_{0i} \tag{17}$$

where $\mathbf{w}_i$ is the reference trajectory for $\mathbf{v}_i$, and $\mathbf{M}$, $\mathbf{b}^T$ and $\mathbf{f}_{0i}$ are given in (Camacho and Bordóns, 2007).

The function from equation (17) is equalled to zero after a derivation procedure resulting the optimal future control sequence computed at discrete-time k:

$$\mathbf{u}_i(k) = \mathbf{M}^{-1}\mathbf{b} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{w}_i - \mathbf{f}) \tag{18}$$

Applying the receding horizon principle on equation (18), the following control relation results:

$$\begin{aligned}\Delta u_i(k\,|\,k) &= \boldsymbol{\gamma}^T(\mathbf{w}_i - \mathbf{f}) = \sum_{j=1}^{p}\gamma_j w_i(k+j\,|\,k) - \sum_{j=1}^{p}\gamma_j F_j(z^{-1})d_i(k)\\ &-\sum_{j=1}^{p}\gamma_j H_j(z^{-1})\Delta u_i(k-1) - \sum_{j=1}^{p}\gamma_j H_j'(z^{-1})\Delta v_{i-1}(k)\end{aligned} \tag{19}$$

where $\boldsymbol{\gamma}^T$ is the first row of $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T$.

The relation (19) can be rewritten as:

$$\begin{aligned}R(z^{-1})\Delta u_i(k\,|\,k) &= T(z^{-1})w_i(k+p\,|\,k) - S(z^{-1})d_i(k)\\ &-\underbrace{V(z^{-1})\Delta v_{i-1}(k)}_{feedforward}\end{aligned} \tag{20}$$

where the main GPC polynomials are:

$$\begin{aligned}R(z^{-1}) &= 1 + \sum_{j=1}^{p}\gamma_j z^{-1}H_j(z^{-1});\ S(z^{-1}) = \sum_{j=1}^{p}\gamma_j F_j(z^{-1});\\ T(z^{-1}) &= \sum_{j=1}^{p}\gamma_j z^{-p+j};\ V(z^{-1}) = \sum_{j=1}^{p}\gamma_j H_j'(z^{-1}).\end{aligned} \tag{21}$$

In equation (20), the feedforward part of the GPC controller is underlined.

3.3. Optimal Control based on LQR Algorithm

The accuracy analysis of the GPC controller was done by a quantitative comparison with another advanced control algorithm based on LQR for VCPS proposed by the authors in (Lazar et al., 2018). Below it is a brief presentation of the LQR algorithm.

The control algorithm based on LQR is used to control the follower vehicles in a CACC architecture and has two components:

$$u_i(k) = u_{ifb}(k) + u_{iff}(k) = \mathbf{K}_i \mathbf{x}_i(k) + G_{ff}(z) v_{i-1}(k). \quad (22)$$

where $\mathbf{K}_i$ is a gain matrix for the feedback regulator and $G_{ff}(z)$ is the transfer function of the feedforward controller.

The feedback component, $u_{ifb}(k)$, is intended for distance control based on an LQR controller and the feedforward one, $u_{iff}(k)$, for rejection of the disturbance introduced by the speed of the front vehicle. The disturbance is considered known, the speed of the front vehicle being transmitted to the follower via a V2V communication system.

The feedback LQR control law was designed using an augmented model obtained from equation (6) by adding two integrators to solve the regulation control problem in relation with the disturbance introduced by the front vehicle:

$$\begin{aligned} \dot{x}_{3i} &= d_i^* - d_i \\ \dot{x}_{4i} &= x_{3i} \end{aligned} \quad (23)$$

Discretizing the augmented model, the discrete-time model for the i^{th} vehicle was obtained:

$$\begin{cases} \mathbf{x}_i(k+1) = \mathbf{A}_d \mathbf{x}_i(k) + \mathbf{b}_d u_i(k) + \mathbf{d}_{av} v_{i-1}(k) + \mathbf{h}_a d_i^*(k) \\ d_i(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_i(k) \end{cases} \quad (24)$$

where $\mathbf{x}_i(k) = \begin{bmatrix} d_i(k) & v_i(k) & x_{3i}(k) & x_{4i}(k) \end{bmatrix}^T$ and $\mathbf{A}_d$, $\mathbf{b}_d$, $\mathbf{d}_{av}$ and $\mathbf{h}_a$ are given in (Lazar *et al.*, 2018).

By minimizing the cost function:

$$J = \sum_{k=0}^{\infty} \left(\mathbf{x}_i^T(k) Q \mathbf{x}_i(k) + u_i^T(k) N u_i(k) \right). \quad (25)$$

where Q and N are weight matrices, the feedback LQR control law was found:

$$u_{ifb}(k) = \mathbf{K}_i \mathbf{x}_i(k) = k_{1i} d_i(k) + k_{2i} v_i(k) + k_{3i} x_{3i}(k) + k_{4i} x_{4i}(k) \quad (26)$$

where the coefficients k_{1i} to k_{4i} are the elements of the gain matrix $\mathbf{K}_i$ obtained after the optimization of the quadratic cost function (25).

The feedforward component of the controller, u_{iff}, can be determined considering the closed-loop model:

$$\begin{cases} \mathbf{x}_i(k+1) = (\mathbf{A}_d + \mathbf{b}_d \mathbf{K}_i) \mathbf{x}_i(k) + (\mathbf{b}_d G_{ff}(z) + \mathbf{d}_{av}) v_{i-1}(k) \\ \qquad + \mathbf{h}_a d_i^*(k) \\ d_i(k) = \mathbf{c}^T \mathbf{x}_i(k) \end{cases} \quad (27)$$

Applying Z transform on the equation (27) and considering as input the disturbance v_{i-1}, the closed-loop transfer function $G_{0d}(z)$ results:

$$d_i(k) = \underbrace{\mathbf{c}^T (\mathbf{I}z - (\mathbf{A}_d + \mathbf{b}_d \mathbf{K}_i))^{-1} (\mathbf{b}_d G_{ff}(z) + \mathbf{d}_{av})}_{G_{od}(z)} v_{i-1}(k) \quad (28)$$

To eliminate the undesired effect of the measurable disturbance v_{i-1}, $G_{0d}(z)$ should be equalled to zero. Thus, the feedforward controller with the following transfer function is determined:

$$G_{ff}(z) = \frac{\left(\tau_v + T_s\left(1 - k_2 K_v\right)\right) z - \tau_v}{K_v T_s z} \quad (29)$$

4. SIMULATION STUDY

In order to analyse the performance of the proposed algorithms (GPC and LQR) they were implemented in MATLAB/Simulink considering a scenario in which a set of identical vehicles grouped as a homogeneous platoon are travelling in a city and on a highway. This case study takes into account the following premises:

- no speed restrictions are considered;
- the platoon is not changing the lane;
- there are no obstacles on the lane on which the platoon is travelling and no other things that can cause the splitting of the platoon;
- the driver of the platoon's lead vehicle is controlling its speed using the on-board CC system;
- the followers are autonomously reacting to the accelerations or decelerations of the leader.

The created platoon simulators are initialized with the start positions of the vehicles:

$$\begin{aligned} p_l^0 &= (n+1) L_{veh} + n d_0 \\ p_i^0 &= (n-i+1) L_{veh} + (n-i) d_0, \; i = \overline{1, n}. \end{aligned} \quad (30)$$

Each control algorithm was simulated including it in a platoon with one leader and fifteen followers. The simulations were realized using a set of parameters that are described in the next tables. In (Tiganasu et al., 2017), the leader's vehicle dynamics contains an actuator characterized by the parameters K_a and τ_a, given in Table 1, that are used in the design of its PID controller. On the other side, the variables t_{set} and ζ from Table 2 are used in (Lazar et al., 2018) to determine the expression of the leader's proportional-integral (PI) controller.

Table 1. GPC-CACC specific platoon parameters.

Parameter	Value	Description
λ	0	Weight factor
p	12	Prediction horizon
K_a	10	Actuator proportional factor (leader's PID controller)
τ_a	0.2s	Actuator time constant (leader's PID controller)

Table 2. LQR-CACC specific platoon parameters.

Parameter	Value	Description
t_{set}	32s	Settling time (leader's PI controller)
ζ	0.9	Damping factor (leader's PI controller)

In Table 3 a series of common parameters for the simulated vehicle platoons is described.

Table 3. CACC platoon common parameters.

Parameter	Value	Description
T_s	0.01s	Sample time
v_w	2 m/s	Wind speed
v_0	0 m/s	Initial vehicle speed
ρ	1.202 kg/m^3	Air density
m	1000 kg	Vehicle mass
C_d	0.5	Resistance coefficient
A_{rc}	1 m^2	Vehicle frontal area
t_h	0.7 s	Time-headway
d_0	1 m	Standstill distance
L_{veh}	5 m	Vehicle length
v^*	NEDC (Fig. 3)	Leader's reference speed (introduced in the system as a specific speed profile)
K_v	0.8319 (m/s)/N	Proportional gain for vehicle model
τ_v	831.94 s	Vehicle model time constant

The speed reference v* used in this simulation study is the New European Driving Cycle (NEDC) (Pacheco et al., 2013) illustrated in Fig. 3 that is a driving cycle designed for the assessment of the fuel economy and emission levels of passenger vehicles engines. The NEDC is composed of two parts: Urban Driving Cycle, repeated 4 times, which is plotted from 0 s to 780 s and Extra-Urban Driving Cycle that is plotted from 780 s to 1200 s.

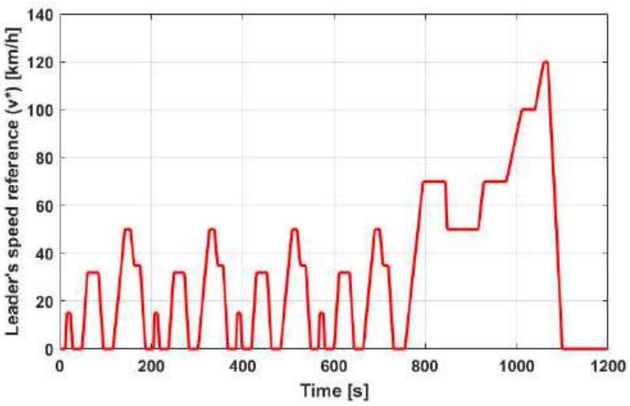


Fig. 3. New European Driving Cycle (speed reference).

4.1. GPC-based Platoon Simulation Results

For the vehicle platoon whose followers contain GPC controllers, the signals obtained through simulation are depicted in the next figures as follows:

- Fig. 4 illustrates the inter-vehicle distances with zooms in two specific areas;
- In Fig. 5 the vehicle velocities together with v^* are depicted;
- In Fig. 6 the path of the vehicles' movement given by their positions together with the total travelling distance can be observed;
- Fig. 7 contains the distance errors as being the difference between the inter-vehicle distances d_i and distance reference d_i^*.

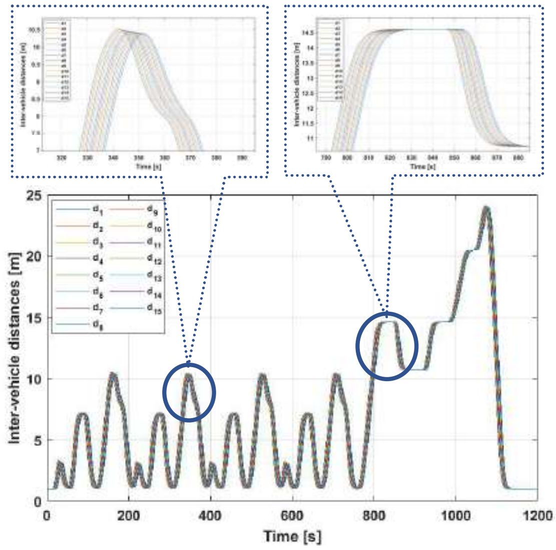


Fig. 4. Inter-vehicle distances for GPC-based VCPS.

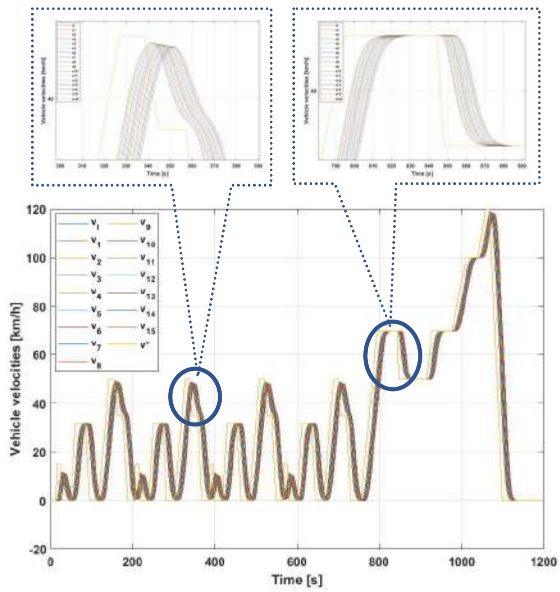


Fig. 5. Vehicle velocities for GPC-based VCPS.

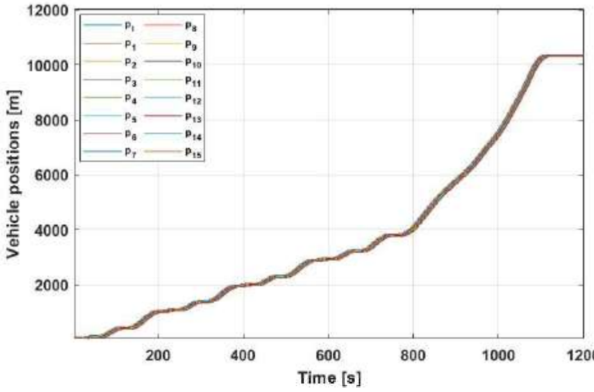


Fig. 6. Vehicle positions for GPC-based VCPS.

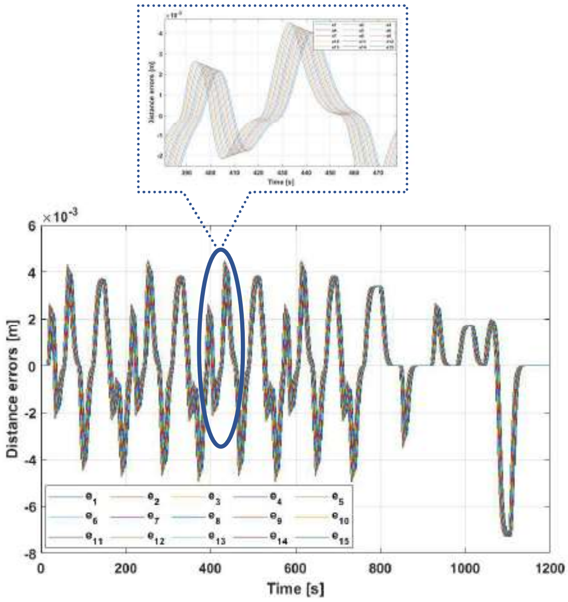


Fig. 7. Distance errors for GPC-based VCPS.

4.2. LQR-based Platoon Simulation Results

In the case of the platoon implemented with LQR controllers for followers, the signals resulted after simulation can be observed as follows:

- Fig. 8 contains the distances between vehicles during their movement;
- The vehicle speeds are depicted in Fig. 9, in which can be seen that they follow the speed profile introduced as reference for the leader;
- Both the platoon travelling distance and the vehicle positions are depicted in Fig. 10;
- The distance errors obtained in this simulation case can be viewed in Fig. 11;

In each figure there are zooms on specific areas that can help to easily analyse the results.

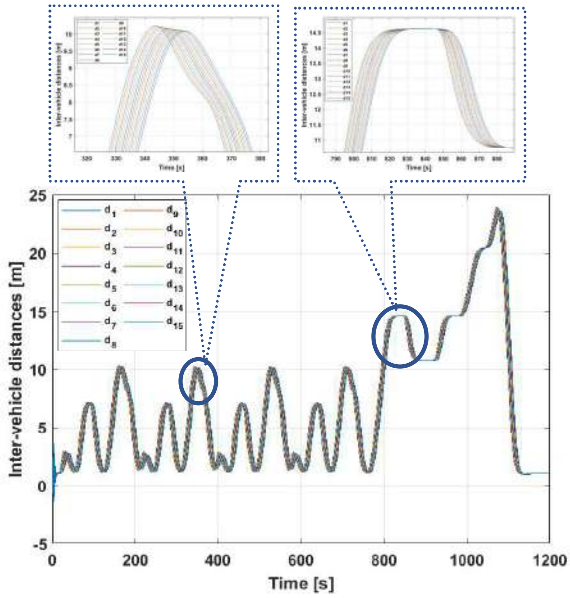


Fig. 8. Inter-vehicle distances for LQR-based VCPS.

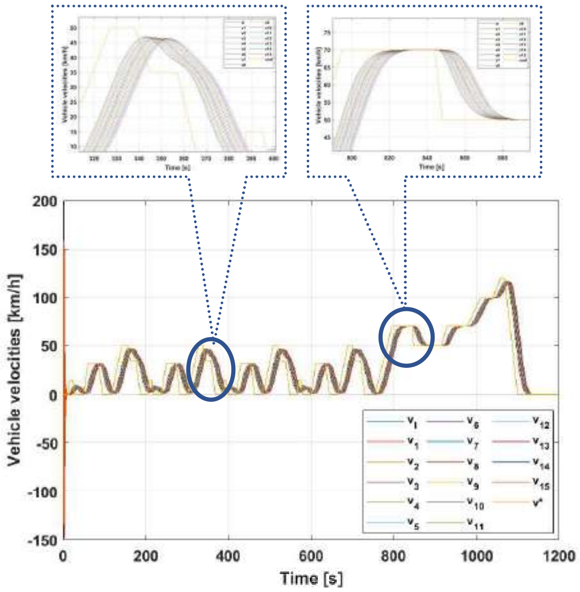


Fig. 9. Vehicle speeds for LQR-based VCPS.

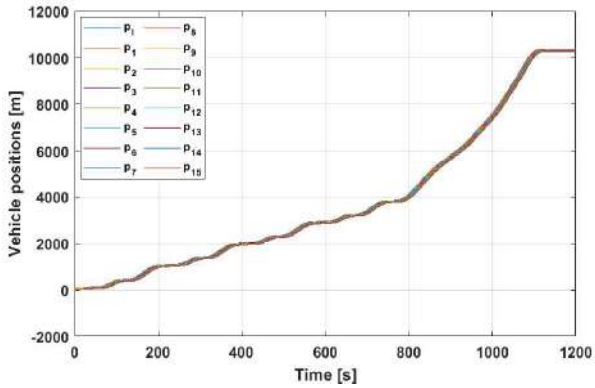


Fig. 10. Vehicle positions for LQR-based VCPS.

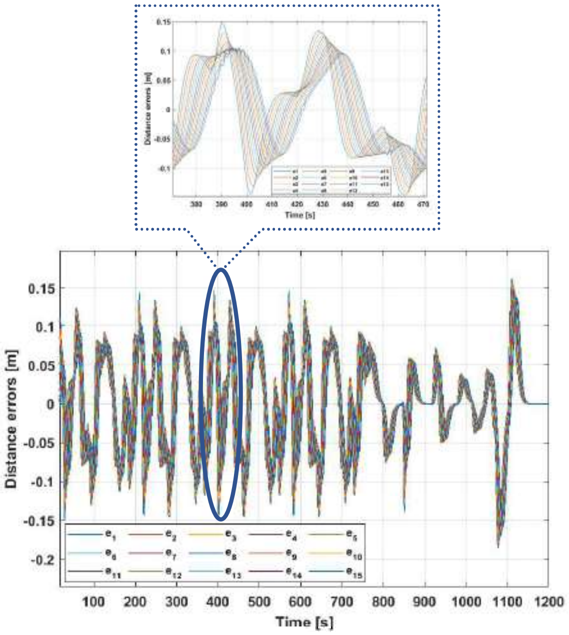


Fig. 11. Distance errors for LQR-based VCPS.

4.3. Comparative Analysis

In order to perform an appropriate comparison between the two simulated VCPS platoon systems proposed by the authors in (Tiganasu et al., 2017) and (Lazar et al., 2018), the following key performance indicators (KPIs) were used:

$$
\begin{aligned}
J_1 &= \sum_{k=T_{start}}^{T_{end}} \sum_{i=1}^{n} \left(\left(d_i^*(k) - d_i(k) \right)^2 + \alpha \left(u_i(k) \right)^2 \right) \\
J_2 &= \sum_{k=T_{start}}^{T_{end}} \sum_{i=1}^{n} \left(\left(d_i^*(k) - d_i(k) \right)^2 \right) \\
J_3 &= \sum_{k=T_{start}}^{T_{end}} \sum_{i=1}^{n} \left(\left| d_i^*(k) - d_i(k) \right| + \alpha \left| u_i(k) \right| \right) \\
J_4 &= \sum_{k=T_{start}}^{T_{end}} \sum_{i=1}^{n} \left(\left| d_i^*(k) - d_i(k) \right| \right)
\end{aligned}
\tag{31}
$$

where T_{start} and T_{end} are start and end time of the simulations, n is the number of followers in the platoon (n = 15 in this case study), α is a weight factor (for GPC-based platoon, $\alpha = \lambda$; for LQR-based platoon, α = 0.5). $d_i^*(k)$ is the speed-dependent distance reference for vehicle i at k moment of time, $d_i(k)$ is the inter-vehicle distance between vehicles i and i-1 and $u_i(k)$ is the vehicle i controller's command. These indicators represent the measure of the total spacing error at the level of the entire platoon weighted in two cases (J_1 and J_3) by the controllers' command values.

In Table 4, the simulation values of the KPIs from equations given in (31) are illustrated and, at a simple look, it is obvious that the LQR-based platoon has huge indicators compared to the GPC VCPS. This means that the first proposed algorithm (GPC) is more appropriate to be used in the design of a real VCPS having overall a better performance.

Table 4. Performance indicators for GPC and LQR platoons comparison.

Indicator	GPC-based VCPS	LQR-based VCPS
J_1	8.52	7.27 * 10^{10}
J_2	8.52	6.1 * 10^3
J_3	2.71 * 10^3	1.85 * 10^8
J_4	2.71 * 10^3	8 * 10^4

Besides the analysis of Table 4, the following comments related to the simulation results (Fig. 4 to Fig. 11) can be used also to formulate a conclusion on the algorithms efficiency:

- For the LQR-based platoon, an initialization phase of approximately 12 s is needed to bring the signals in a steady state while in the case of GPC there are no oscillations at the beginning of the simulation.
- Distance errors for GPC (e.g., at a speed transition from 0 km/h to 32 km/h the error = [0.004, 0.0045] m) (Fig. 7) are smaller than in the case of LQR (e.g., at a speed transition from 0 km/h to 32 km/h the error = [0.095, 0.134] m) (Fig. 11). The first follower in the platoon presents the greatest distance error between it and the leader.

In both cases, the spacing errors are decreasing along the platoon which suggests the string stability of both platoons. In the LQR case, the spacing errors are decreasing faster from one vehicle to another this being visible comparing the zooms from Fig. 7 and Fig. 11.

Fig. 12 depicts the velocity of the first follower vehicle for both GPC and LQR algorithms together with the reference speed of the leader. The purpose of this graph is to make the efficiency of each algorithm more visible from the vehicle following perspective. It can be observed that both vehicles are accelerating and decelerating in order to follow the imposed speed profile.

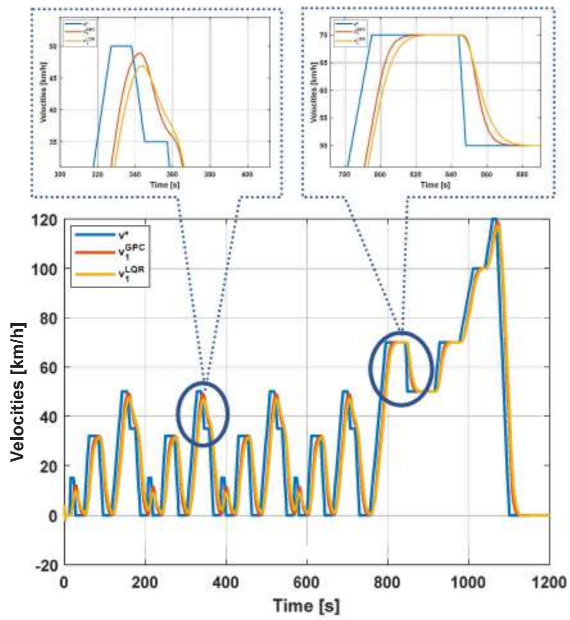


Fig. 12. Velocities of the first follower.

The vehicle with the GPC controller has a faster response than the one with LQR and with values closer to the reference. In both cases no overshoot is visible. At the beginning of the simulation the vehicle with LQR controller presents small oscillations that are disappearing after around 3s for this first follower.

Considering all the aspects mentioned in this subsection, the platoon based on followers implemented with GPC controllers is more performant than the LQR-based platoon.

5. CONCLUSIONS

This paper had as purpose the comparative analysis of two CACC control algorithms proposed by the authors in (Tiganasu et al., 2017) and (Lazar et al., 2018) used to build cyber-physical systems in the form of vehicle platoons. A CPS-oriented platoon organization, that illustrated the split between cyber (communication systems' layer) and physical (the vehicles themselves) planes was illustrated. Also, a diagram with a CACC system architecture was presented. The short description of the control algorithms themselves was included in this paper too, the detailed versions being found in the original papers (Tiganasu et al., 2017) and (Lazar et al., 2018). The results obtained after the simulation of the platoons created based on the architecture from Fig. 2 and on the control algorithms were presented. The use case considered was the travelling in a city and on a highway having the leader equipped with a CC system and the followers with CACC controllers. After analysing the results, it was shown that both algorithms provided good results, but the comparative analysis proved that the GPC-based platoons have better performances than the LQR-based ones.

ACKNOWLEDGEMENT

This work was supported by a research grant of the TUIASI, project number GnaC2018_190.

REFERENCES

Abid, H., Phuong, L. T. T., Wang, J., Lee, S., and Qaisar, S. (2011). V-Cloud: Vehicular Cyber-Physical Systems and Cloud Computing, *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, No. 165, pp. 1-5.

Brookhuis, K. A., De Waard, D., and Janssen, W. H. (2001). Behavioural impacts of Advanced Driver Assistance Systems – an overview, *European Journal of Transport and Infrastructure Research*, Vol. 1, No. 3, pp. 245 – 253.

Camacho, E. F., and Bordóns, C. (2007). *Model Predictive Control*, Springer-Verlag, London, UK.

Desjardins, C., and Chaib-draa, B. (2011). Cooperative Adaptive Cruise Control: A Reinforcement Learning Approach, *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, December.

Dey, K. C., Yan, L., Wang, X., Wang, Y., Shen, H., Chowdhury, M., Yu, L., Qiu, C., and Soundararaj, V. (2015). A Review of Communication, Driver Characteristics and Controls Aspects of Cooperative Adaptive Cruise Control (CACC), *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 2, pp. 491-509.

Flores, C., and Milanés, V. (2018). Fractional-Order-Based ACC/CACC Algorithm for Improving String Stability, *Transportation Research Part C: Emerging Technologies*.

Gacovski, Z., and Deskovski, S. (2014). Different Control Algorithms for a Platoon of Autonomous Vehicles, *International Journal of Robotics and Automation (IJRA)*, Vol. 3, No. 3, pp. 151-160.

Gao, W., Gao, J., Ozbay, J., and Jiang, Z.-P. (2019). Reinforcement-Learning-Based Cooperative Adaptive Cruise Control of Buses in the Lincoln Tunnel Corridor with Time-Varying Topology, *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3796-3805, October.

Hanai, T. (2013). Intelligent Transport Systems, *ITS Japan, Society of Automotive Engineers of Japan*.

Jia, D., Lu, K., Wang, J., Zhang, X., and Shen, X. (2016). A Survey on Platoon-Based Vehicular Cyber-Physical Systems, *IEEE Communication Surveys & Tutorials*, Vol. 18, No. 1, pp. 263-283.

Lazar, C., Tiganasu, A., and Caruntu, F. C. (2018). Arterial Intersection Improvement by Using Vehicle Platooning and Coordinated Start, *15th IFAC Symposium on Control in Transportation Systems (CTS 2018),* Savona, Italy, 6th – 8th June.

Lu, X.-Y., Hedrick, J. K., and Drew, M. (2002). ACC/CACC – Control Design, Stability and Robust Performance, *Proceedings of the American Control Conference Anchorage*, 8th – 10th May.

Lungoci, C., Moga, D., Muresan, V., Petreus, D., Stroia, N., Moga, R., Munteanu, M., Raus, I., Muntean, V., and Mironiuc, A. I. (2015). Hyperthermic Intraperitoneal Chemotherapy Approach Based on Cyber-Physical System Paradigm, *Journal of Control Engineering and Applied Informatics*, Vol. 17, No. 3, pp. 50-59.

Naus, G. J. L., Ploeg, J., Van de Molengraft, M.J.G., Heemels, W.P.M.H., and Steinbuch, M. (2010). Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach, *Control Engineering Practice*, Vol. 18, pp. 882-892.

Omae, M., Fukuda, R., Ogitsu, T., and Chiang, W.-P. (2013). Spacing Control of Cooperative Adaptive Cruise Control for Heavy-Duty Vehicles, *7th IFAC Symposium on Advances in Automotive Control*, Tokyo, Japan, 4th - 7th September.

Öncü, S., Ploeg, J., van de Wouw, N., and Nijmeijer, H. (2014). Cooperative Adaptive Cruise Control: Network-Aware Analysis of String Stability, *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, August.

Pacheco, A. F., Martins, M. E. S., and Zhao, H. (2013), New European Drive Cycle (NEDC) simulation of a passenger car with a HCCI engine: Emissions and fuel consumption results, *Fuel*, vol. 111, pp. 733-739.

Patil, A., More, K., and Kulkarni, S. (2018). A Review on Vehicular Cyber Physical Systems, *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 2, pp. 1138-1143.

Regan, M. A., Oxley, J. A., Godley, S. T., and Tingvall, C. (2001). Intelligent Transport Systems: Safety and Human Factors Issues, *Royal Automobile Club of Victoria*, Noble Park, Victoria, Australia.

Riaz, F., and Niazi, M. A. (2016). Road collisions avoidance using vehicular cyber-physical systems: a taxonomy and review, *Complex Adaptive Systems Modeling*, Springer Open.

Sanislav, T., and Miclea, L. (2012). Cyber-Physical Systems – Concept, Challenges and Research Areas, *Journal of Control Engineering and Applied Informatics*, Vol. 14, No. 2, pp. 28-33.

Tigadi, A., Gujanatti, R., and Gonchi, A. (2016). Advanced Driver Assistance Systems, *International Journal of Engineering Research and General Science*, Vol. 4, No. 3, pp. 151-158, May-June.

Tiganasu, A., Lazar, C., and Caruntu, F. C. (2017). Cyber Physical Systems – oriented Design of Cooperative Control for Vehicle Platooning, *21st International Conference on Control Systems and Computer Science (CSCS21)*, Bucharest, 29th - 31st May.

Ulsoy, A. G., Peng, H., and Çakmakc, M. (2012). *Automotive Control Systems*, Cambridge University Press, USA.

van der Heijden, R., Lukaseder, T., and Kargl, F. (2017). Analyzing Attacks on Cooperative Adaptive Cruise Control (CACC), *IEEE Vehicular Networking Conference (VNC)*, Torino, Italy, 27th – 29th November.

Varada, K. (2017). Distributed Cooperative Control of Heterogeneous Multi-Vehicle Platoons, *Open Access Master's Theses*.

VDA (Verband der Automobilindustrie) (2015). Automation – From Driver Assistance Systems to Automated Driving, Berlin, September.

Wang, Z., Wu, G., and Barth, M. J. (2018). A Review on Cooperative Adaptive Cruise Control (CACC) Systems: Architectures, Controls, and Applications, *21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, USA, 4th – 7th November.

Ward, J., Smith, P., Pierce, D., Bevly, D., Richardson, P., Lakshmanan, S., Argyris, A., Smyth, B., Adam, C., and Heim, S. (2019). Cooperative Adaptive Cruise Control (CACC) in Controlled and Real-World Environments: Testing and Results" *NDIA Ground Vehicle Systems Engineering and Technology Symposium*, Novi, Michigan, 13th – 15th August.

Wei, S., Zou, Y., Zhang, T., Zhang, X., and Wang, W. (2018). Design and Experimental Validation of a Cooperative Adaptive Cruise Control System Based on Supervised Reinforcement Learning, *MDPI Applied Sciences Journal*, 21st June.

Wei, Y., Guo, Ge, and Wang, L. Y. (2017). Bidirectional Platoon Control of Arduino Controlled Cars with Actuator Saturation and Time-varying Delay, *Journal of Control Engineering and Applied Informatics*, Vol. 19, No. 1, pp. 37-48.

Williams, B. (2008). *Intelligent Transport Systems Standards*, Artech House, Boston, USA.

Wu, C., Lin, Y., and Eskandarian, A. (2019). Cooperative Adaptive Cruise Control with Adaptive Kalman Filter Subject to Temporary Communication Loss, *IEEE Access*, vol. 7, 11th July.

Zhao, H., Sun, D., Yue, H., Zhao, M., and Cheng S. (2018). Dynamic Trust Model for Vehicular Cyber-Physical Systems, *International Journal of Network Security*, Vol. 20, No. 1, pp. 157-167.

applied sciences

Article

Simulation Power vs. Immersive Capabilities: Enhanced Understanding and Interaction with Digital Twin of a Mechatronic System

Constantin-Catalin Dosoftei

Department of Automatic Control and Applied Informatics, "Gheorghe Asachi" Technical University of Iasi, 700050 Iasi, Romania; constantin-catalin.dosoftei@academic.tuiasi.ro

Abstract: Automation Studio, a specialised simulation software, offers virtual commissioning capabilities and robust tools for modelling the behaviour and performance of a pneumatic robot controlled by a PLC. Conversely, Unity is a versatile platform primarily used for creating high-quality 3D games and interactive simulations, providing immersive experiences with DTs through mixed-reality environments. This paper provides a study that compares and contrasts the simulation power of Automation Studio and the immersive capabilities of Unity in the context of developing digital twins for a mechatronic system. This research explores how these complementary approaches enhance the development, validation, understanding, and interaction with digital twins. By examining both platforms in this context, the article provides valuable insights for engineers, developers, and researchers looking to create digital twins for mechatronic systems, but not only. This study demonstrates the potential of leveraging the combined power of simulation and immersive capabilities to improve the interaction between the real robotic arm manipulator cylindrical type and its digital twin in different scenarios, using an OPC approach for mirroring. The combination of Automation Studio and Unity provides a powerful platform for applied science education in the field of the digital twin of mechatronic systems.

Keywords: digital twin; mechatronic system; virtual commissioning; mixed reality; cyber-physical system

check for updates

Citation: Dosoftei, C.-C. Simulation Power vs. Immersive Capabilities: Enhanced Understanding and Interaction with Digital Twin of a Mechatronic System. *Appl. Sci.* **2023**, *13*, 6463. https://doi.org/10.3390/app13116463

Academic Editors: Francesco Longo, Antonio Padovano, Vittorio Solina and Giovanni Mirabelli

Received: 30 April 2023
Revised: 20 May 2023
Accepted: 22 May 2023
Published: 25 May 2023

1. Introduction

Industry 4.0 is a new reality of the modern economy because innovation and technological development play important roles in all organisations. As the first industrial revolution improved the operation of manufacturing industries, the second one introduced electricity into the industry, the third automated the uniform tasks of line workers, and the fourth aimed at improving information management and decision making. Cyber-physical systems (CPS), the Internet of Things (IoT), the Internet of Service (IoS), and smart factories are the four main pillars of Industry 4.0 [1]. These components work together to create highly automated and efficient production environments. Seven design principles further define Industry 4.0: interoperability, virtualisation, decentralisation, real-time capability, service orientation, modularity, cost reduction, and efficiency [2]. All components of the industrial network have a flawless connection thanks to interoperability. Decentralisation spreads decision-making and control, while virtualisation makes it possible to represent real assets digitally. Rapid adaptation to shifting circumstances is made possible by real-time capabilities, whereas service orientation emphasises the integration of modular, reusable services. Developing systems with easily interchangeable components is called modularity. Cost reduction and efficiency are focused on optimising industrial operations to increase productivity and profitability while minimising waste and resource consumption. Smart manufacturing systems can improve process efficiency, reduce downtime, and enhance supply chain management by leveraging evolutive technologies such as

IIoT, AI, big data analytics, additive manufacturing, distributed ledgers, cloud computing, and robotics. By integrating the latest technologies into their operations, organisations can reduce costs, increase productivity, and enhance sustainability, all while delivering high-quality products and services to customers. Businesses may develop highly automated, interconnected, and effective smart manufacturing systems by implementing Industry 4.0 ideas and technology. Organisations must undergo this change to be competitive in a world market that is always changing. Businesses may provide value to consumers and shareholders while fostering sustainable economic growth by utilising the most recent digital technology to enhance operations and achieve long-term success.

To define the term I4.0 only through one word, this is digitalisation. The concept of "digitalisation" alludes to our society's greater reliance on automation, networked, and web-enabled technology [3]. The main technology pushing digital transformation across all industries, and not only, is called digital twin (DT), representing a concept associated with cyber–physical integration. The DT concept came to life in 2002 at the University of Michigan when Prof. Grieves introduced, during a PLM course, the concept of virtual space that was equivalent to a physical product and with a bidirectional communication channel between these two spaces [4]. Presently, there are several different definitions of a digital twin [5–10]. Nonetheless, the essence of all of them stays with the initial concept of a mirrored spaces model with bridges of communication between the physical world and the virtual world, and the conceptual model is presented in Figure 1.

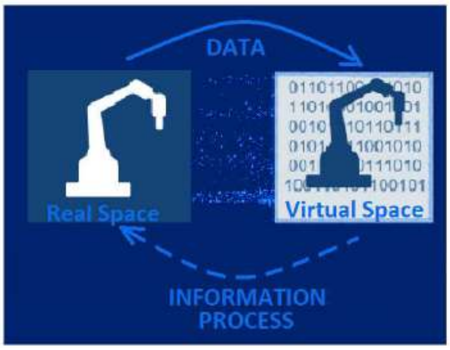


Figure 1. Conceptual model of the digital twin.

Despite the DT philosophy and model proposed in 2002, its application remained limited in the first decade for various reasons [11]. During this time, another concept associated with integrating CPS made significant progress, and CPS was hailed as a main pillar of Industry 4.0. Some perspectives view DT as a subset of CPS, [12]; however, notable differences exist. Introduced in 2006, CPS primarily functions as a scientific category, addressing complex systems that traditional IT terminology could not sufficiently describe. CPS represents a profound integration of computation, networking, and physical processes. Embedded computers and networks monitor and control the physical processes, with feedback loops where physical processes affect computations and vice versa. In contrast, a digital twin (DT) is a high-fidelity virtual model that mirrors a physical entity. A DT simulates the physical entity's behaviour in a virtual environment by utilising real-time data, allowing system performance monitoring, prediction, and optimisation.

To underscore this distinction, consider a traffic monitoring and control system. In the CPS context, this system involves surveillance cameras, traffic sensors, traffic lights, and embedded computers communicating with each other to adjust the traffic flow based on real-time road conditions. On the other hand, a DT for the same system would be a detailed virtual model of the traffic network, replicating the real-world situation in real-time based on data collected from sensors and cameras. While CPS and DT aim to integrate the physical and digital worlds, their functions and applications differ. CPS focuses on the real-time monitoring and control of physical systems using computational processes, while DT centres around virtual modelling and simulation. Even though CPS and DT share the goal of bridging the gap between the physical and digital domains, their primary distinction lies in their categorisation. As a scientific category, CPS focuses on integrating computational and physical processes. Conversely, as an engineering category, DT emphasises using a digital copy of a physical system for real-time optimisation. This difference influences their respective roles and applications in industrial practices, with DTs being more prevalently used for enhancing precision and management.

Let us consider another example—a wind turbine system to illustrate these differences. In the CPS context, integrated sensors and controllers within the wind turbine system continuously measure variables such as wind speed, blade position, and power output. These data are then processed to adjust the turbine's blade position and other operational parameters to optimise the real-time power output. On the other hand, a DT of the same wind turbine would be a virtual model that uses real-time input data from the CPS to simulate the turbine's operation in a virtual environment. Engineers can use this model to anticipate and plan maintenance, assess the impact of different weather scenarios on turbine performance, and test new control strategies or equipment configurations in a safe and controlled environment.

These examples illustrate how CPS and DT complement each other to maximise the efficiency and reliability of modern industrial systems. While both technologies allow for greater integration between the physical and digital worlds, they approach this task from different perspectives and with various tools, proving their immense value in the era of Industry 4.0. manufacturing; their symbiosis promises a leap forward in the efficiency, flexibility, and intelligence of production systems.

1.1. DT Applications and Benefits

DTs are more commonly used in applied sciences due to their practical applications and direct implementation in various industries [8,13]:

- Manufacturing—used to improve production efficiency, reduce downtime, optimise the flow of materials and resources, and enhance product quality;
- Aerospace—utilised for monitoring, weight measuring, the exact specification of climatic factors, flying time measurement, and fault diagnosis in aircraft;
- Automotive—used to simulate the behaviour of various components and systems, allowing for the early detection of potential problems and reducing the need for physical prototyping. Nowadays, it can be used to develop autonomous vehicles and support the development of new technologies such as electric powertrains and advanced driver assistance systems;
- Agriculture—enables farmers to create virtual representations of their crops, allowing for the real-time monitoring and optimisation of growth and yield;
- Healthcare—utilised to model and simulate the human body. Doctors can test various therapies and forecast outcomes before actually providing them by creating a virtual clone of the patient;
- Construction—allows architects and engineers to create virtual representations of buildings, which can be used to identify potential problems and optimise performance before construction even begins;
- Energy—used to simulate and optimise power generation and distribution systems;
- Education—allows for the creation of holographic replicas of actual things and systems, resulting in a more engaging and participatory learning environment for actual native digital students.

DTs have experienced significant growth in recent years, both conceptually and practically. As DTs evolve, they are expected to create a "cyber-physical continuum" that blurs the distinction between the virtual and the real [14,15]. With the introduction of autonomous decision-making, the applications of DTs will only continue to grow, creating new opportunities for innovation and progress.

DT technology has many advantages to offer enterprises, including increased reliability, productivity, reduced risk, lower maintenance costs, faster production, and better customer service. DT also creates new business opportunities and provides insights into product performance throughout its lifecycle. When looking at the DT benefits, it is possible to see that this technology positively impacts several aspects of the manufacturing industry and the supply chain as a whole. Some of the benefits of introducing digital twinning technologies within the company include improved and the research, design, and development of new products and services, greater process efficiency and productivity, reduced waste, increased supply chain visibility, and more efficient unit-level traceability, reduced risk of counterfeit and adulterations, and increased product safety [16]. Access to more data about a product's journey to the end user and customer information is also a significant benefit of DT technology.

1.2. DT Adoption in Romania

The range of DTs' applicability has expanded from the product design stage to the production, operation, and service stages, garnering considerable interest from academics and businesses alike.

DT technology is in the early stages of adoption in Romania, but its potential applications are being explored in various industries. Examples include implementing DT-based smart city components in a few cities; DT applications in automotive manufacturing started from robotic applications and utility companies, especially from the energy and healthcare sectors. The adoption of DT technology in Romania is being facilitated by the participation of representatives from major universities and the uptake of DT research themes from variousindustries [17–20]. In addition, university involvement in digital transformation hubs promotes the widespread adoption of DTs in the Romanian environment [21,22]. This engagement by the academic community and collaboration with digital transformation initiatives create opportunities for the wider adoption of DTs in Romania. On the other hand, the development of the IT industry in Romania has created the conditions for the opening of subsidiaries of multinational companies involved in software development and the implementation of the DT concept in different sectors, as well as local companies that integrate DT in special near-robot applications from the manufacturing industry.

1.3. Synergy between DT, Virtual Commissioning, and Mixed Reality

Digital twin is a key technology for improved industrial performance, as it integrates or merges with many other technologies from the Industry 4.0 paradigm. These include IIoT, big data and analytics, artificial intelligence, cloud computing, virtual commissioning, modelling and simulation, extended reality, and autonomous robots, as depicted in Figure 2. The particular technology employed in digital twin technology varies based on the nature of the application and its specifications.

The research presents a systematic approach combining digital twin technology with VC to optimise the design and operation of a cylindrical robot while incorporating mixed reality for seamless interaction with the real robot and its hologram. This synergy facilitates real-time adjustments, enhancing the system's overall performance and minimising the need for physical prototyping. Furthermore, this integrated solution streamlines the development process, enabling students and engineers to identify potential issues, validate control algorithms, and test various scenarios in a secure virtual space before the actual implementation. Additionally, this synergy emphasises a multidisciplinary approach encompassing the robot's design, functionality, and performance, addressing challenges that may arise in diverse working scenarios.

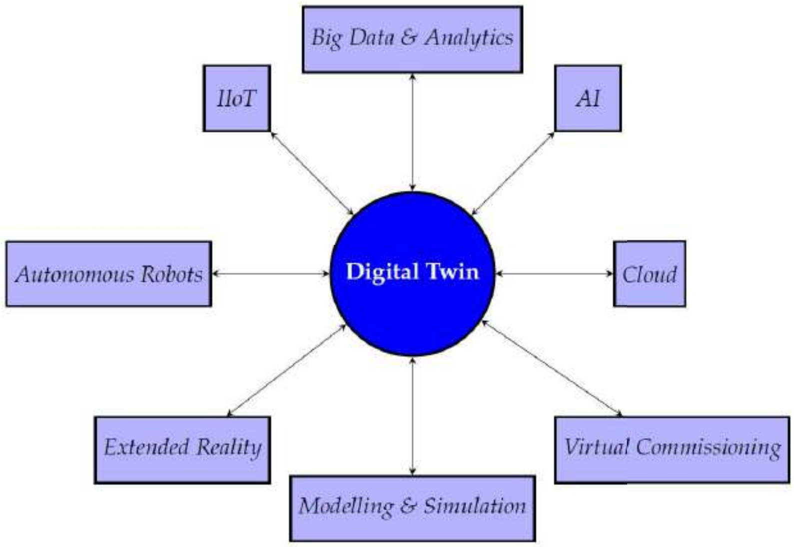


Figure 2. Synergy between DT and other components of Industry 4.0.

This paper's organisation is as follows: Section 2 covers DT and VC applications in robotics, software tools for DT implementation, and the role of XR technologies, particularly MR, in data accessibility and analysis. Section 3 overviews the development workflow for a 3DoF pneumatic cylindrical robot's DT application. Section 4 presents the implementation and validation of a digital twin in AS environment. Section 5 presents the implementation of the same robot in Unity and facilities a holographic approach. Section 6 presents the comparative advantages of both implementations. Section 7 comes with a few conclusions and directions for future research.

2. Related Work

DTs can improve design, production, operation, maintenance, training, and end-of-life administration over the whole lifecycle of a component or system, and [23] provides a comprehensive list of the applications of the digital twin classification throughout the lifecycle. This paper demonstrates that most digital twin applications and use cases belong to "development" or "operations" categories. The second is the main application area, and the function of the digital twin for both cycles received almost no consideration to present DevOps. Considering this gap, the current work is positioned precisely in this area because it includes the VC area after the creation of the virtual model of the robot. Applications in mechatronic systems represent an important category in scientific papers which present VC and DTs. Combining mechatronic components, control software, and digital technologies such as VC and DTs can help enterprises optimise their operations [24–26].

The application of DTs and VC in robotics provides significant benefits in terms of cost savings, improved performance, and streamlined development processes. These advantages have led to the rapid adoption of DTs and VC technologies in the robotics field, making it a top area of interest for both industries and researchers [27–30].

Regarding software platforms and tools available for DT implementation in different architectures, industry and academia have different choices due to the distinct needs and goals [31]. While academia favours more adaptable and configurable tools to serve a wide range of research topics (Matlab, LabView, Automation Studio, Ciros, Ansys, Unity, V-REP, Gazebo), the industry often looks for comprehensive, scalable solutions that can be quickly integrated into existing systems (Siemens PLM Software, FastSuite-2, Dassault's 3D experience, PTC's ThingWorx, AVEVA's digital twin, GE's Predix, RoboDK, CoppeliaSim, VisualComponents, Emulate3D) [1,10,23,28,32].

The huge volume of data around DTs can be difficult for non-expert users to evaluate and analyse. In order to increase the accessibility and usability of the data produced by DTs, XR technologies are used to bridge the gap between the virtual and real worlds. The potential of combining the strengths of DT and XR (specifically MR, [33]) to create more efficient and effective approaches for various applications, such as design, operation, maintenance, and training, was recognised by industries and researchers, which led to the beginning of the strong connection between these technologies. The benefits of leveraging mixed reality as an effective front–end for digital twin models in a more intuitive, immersive, and interactive manner are presented in [34–38].

In the specialised literature, there appears to be a lack of research exploring the combined approach of implementing digital twins using both Automation Studio and Unity.

3. Materials and Methods

The development workflow of the case study DT application for a 3DoF cylindrical robot type (or RPP robot, which contains three linkages and three joints—one revolute joint that rotates about the base, and two prismatic joints), as presented in Figure 3, is based on three main elements: (1) 3D asset creation; (2) DT and VC application development—the ecosystem around AS; and (3) DT and MR application development—the ecosystem around Unity. The interactions between architectural components, physical and cyber layers, are a part of the DT approach and are realised through the OPC approach, which is usually used in horizontal and vertical communication automation processes because it provides semantic interoperability for all layers.

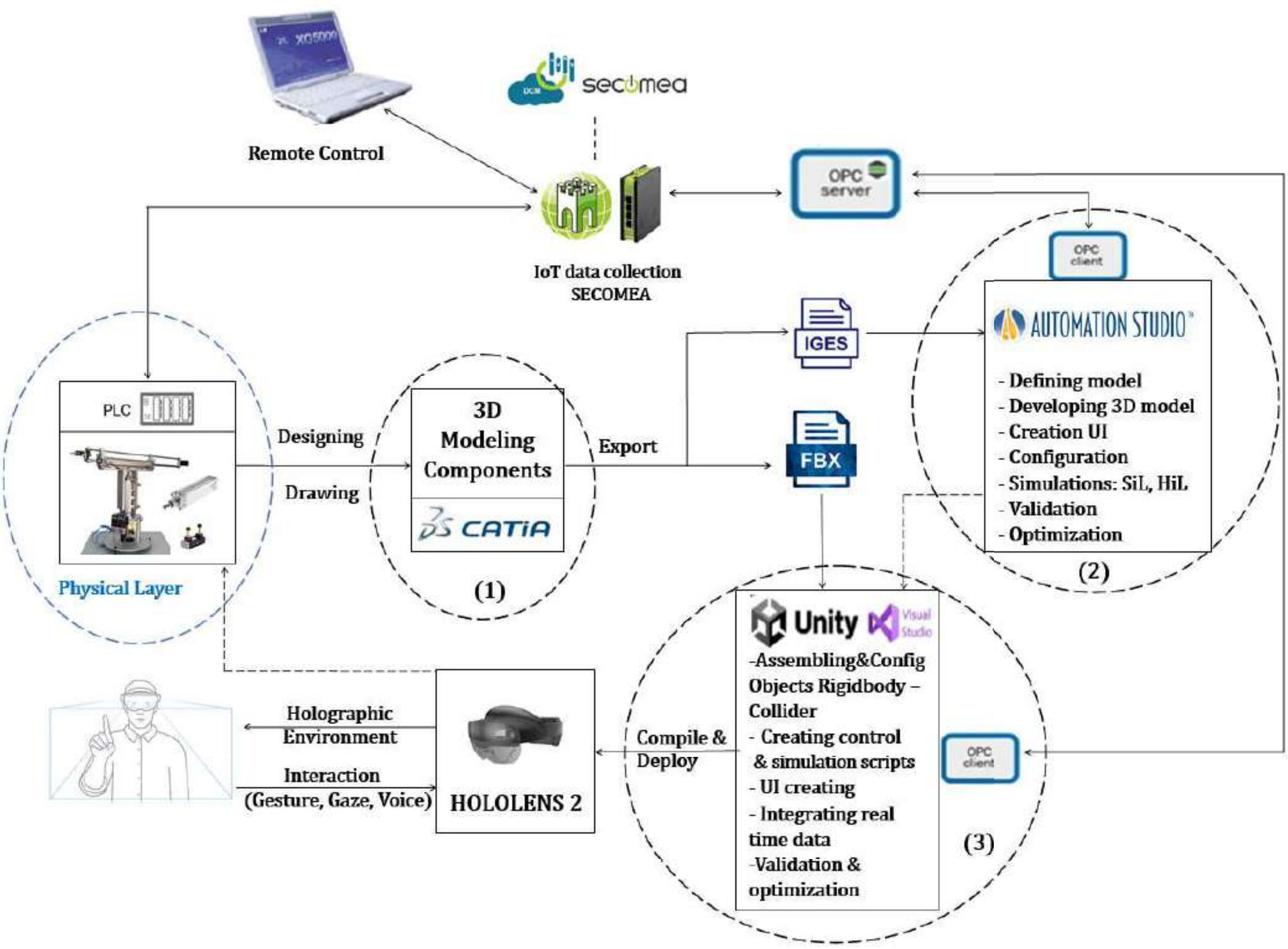
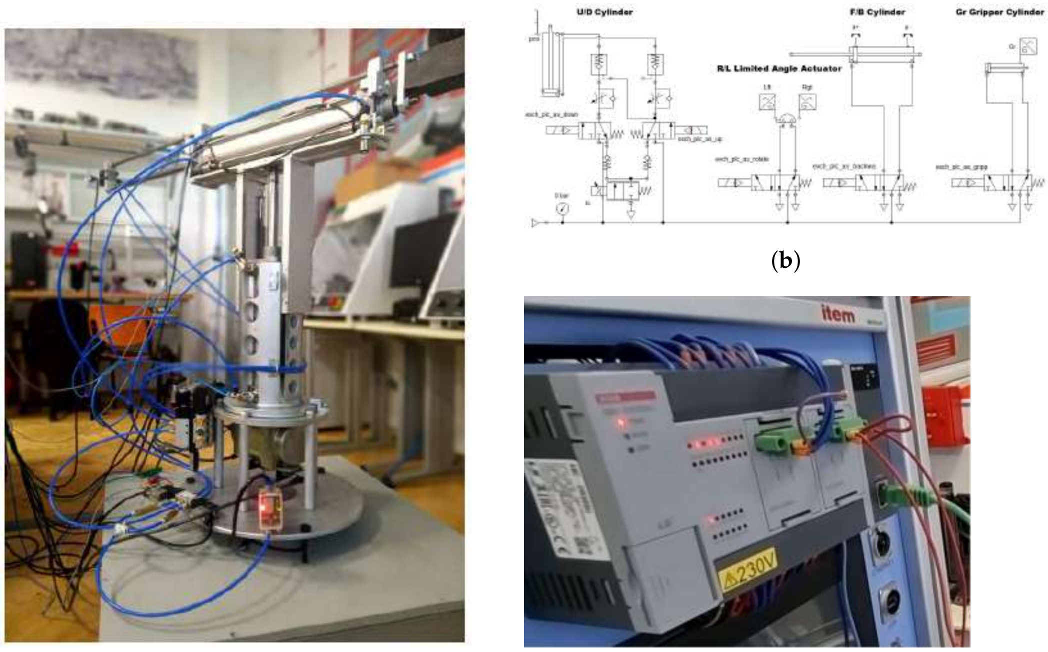


Figure 3. The development workflow of the project.

3DoF Pneumatic Cylindrical Robot

For activities requiring repetitive, accurate motions in a constrained environment, such as pick-and-place operations, assembly tasks, or simple welding tasks, a 3DoF pneumatic cylindrical robot is well suited. The RPP configuration implies that the robot has one revolute joint followed by two prismatic joints. In a pneumatic cylindrical robot, all movements are executed through pneumatic actuators. These actuators convert compressed air into mechanical energy, providing a fast and responsive actuation method. The robot's joints are powered by these actuators, allowing it to perform various tasks within its cylindrical workspace. This robot, as depicted in Figure 4a, is an educational setup from Tech-Con Lab—Department of Automatic Control and Applied Informatics from TUIASI—and used by students to learn about the principles and operation of pneumatic systems, control algorithms, and robotic kinematics.



(a) (b) (c)

Figure 4. The real RPP robot configuration: (**a**) the real 3DoF pneumatic robot; (**b**) the pneumatic circuit diagram of the robot; and (**c**) the PLC of the robot.

The following pneumatic actuators power the robot's joints:

1. Revolute joint (R): a pneumatic rotary actuator—called L/R—(model ARP 055) is used to execute the rotation of the base joint. It allows the robot arm to rotate horizontally at a maximum of 90°, panning within the cylindrical workspace;
2. Prismatic joint 1 (P1): a pneumatic linear actuator used to control the vertical motion of the robot arm—is called U/D. This actuator extends or retracts to raise or lower the arm within the workspace. The model is 41M2P080A0200, a magnetic double action cylinder with Ø80 mm and 200 mm strokes.
3. Prismatic joint 2 (P2): another pneumatic cylinder used to control the robot arm's radial motion is F/B. This actuator extends or retracts to adjust the robot's reach within the cylindrical space. The model is 60M6L063A040, a magnetic bidirectional cylinder with Ø63 mm and 400 mm stroke. It is a bidirectional cylinder because one end contains EOAT—180° angular gripper model CGSN-25 and the other has a counterweight to ensure smooth movement.

The control device for the robot is implemented with a PLC model XBC-DR30SU, which is a compact PLC offering a variety of input and output (I/O) options and communication capabilities. The PLC receives input signals from various sensors (inductive limit switches for L/R pneumatic rotary actuator and gripper, ultrasonic sensors for the U/D cylinder and magnetic sensors for the F/B cylinder). The PLC sends output signals to the solenoid valves with electric commands and the external servo pilot, causing the robot's joints to move and perform the desired tasks. The rotary actuator uses a bistable valve 5/3 with a central position closed model 338D-015-02 with coils voltage of 24 V DC—which receives the command directly from the PLC output. For the F/B cylinder and gripper valves, monostable solenoid valves 5/2 model 358-015 are used. Control in position for the U/D cylinder is necessary to use a valve 2X3/2 NC model 338D-015-02 and a proportional solenoid valve 2/2 way NC model 21A2KCV15-10 connected to a controller +Smart.

4. Implementation and Validation DT in AS Ecosystem

One of the challenges in using modelling and simulation software, particularly when employing high-level modelling approaches with predefined elements from internal libraries, is the limitation imposed on customisation and flexibility. It is the case with industry applications for building DT models. This becomes evident when a low-level modelling approach is required, which involves building new components or mechatronic systems using CAD data and focusing on assembly and functional interactions between various elements. This is the case for the project from this paper developed in AS (a physics-based modelling and simulation tool with workshops such as pneumatic, hydraulic, electrical, and control), where the analytical and dynamic behaviour of the entire model can provide significant added value for a pneumatic mechatronic system. The process follows a V-diagram, as shown in Figure 5, starting from requirements powerfully linked with the tasks of the pneumatic robot from physical space up to the validation and optimisation of DT through the VC approach.

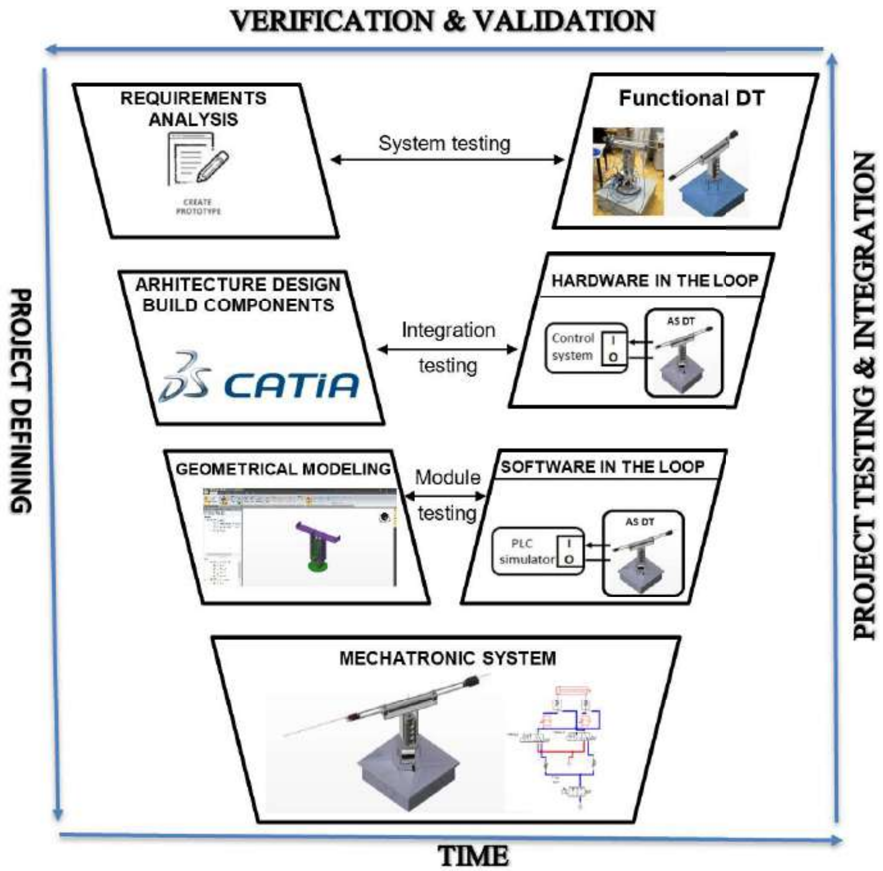
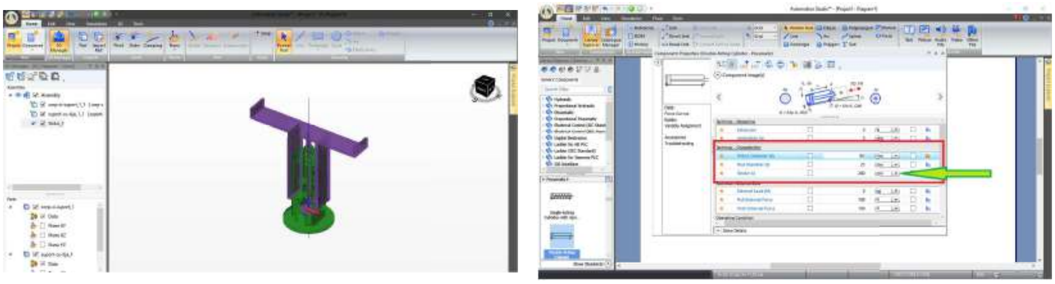


Figure 5. V-model of implementation in AS—reinterpreted from [1].

The 3D model is changed into a simulation model as the first step towards obtaining a digital twin. To accomplish this, 3D models (with an .igs extension) must be imported into the AS—3D workshop. After, it is necessary to define the required rigid and collision bodies and create the type of displacement for each joint—Figure 6a.



(**a**) (**b**)

Figure 6. The implementation of first prismatic cylinder (U/D) in AS for two approaches: (**a**) the screenshot from the 3D editor workshop where the constraints and the displacement were defined; and (**b**) the screenshot from the pneumatic workshop to customise parameters similar to a real cylinder.

Virtual commissioning can be performed using software-in-the-loop simulation (SIL) or hardware-in-the-loop simulation (HIL). The role of a PLC in the digital twin model is two-fold: it provides real-time data to the digital model and allows for the virtual testing and optimisation of control system logic, notably streamlining the process of virtual commissioning. The PLC simulator from AS is used in the SIL simulation to control the virtual robot. The PLC program is checked and runs in a real PLC linked to the virtual robot developed in AS, which runs on the PC during the HIL simulation. Because the model of PLC is not included in the PLC library from AS, it is necessary to work with a generic PLC that closely matches the specifications of a real PLC from an I/O point of view. The communication configuration between the real PLC and the AS environment uses an OPC server to establish the connection, as presented in Figure 3. Setting up the OPC server software on the computer required configuring communication settings (PLC IP address, communication protocol, and port number) and adding the desired PLC tags/variables to the OPC server.

The realistic behaviour of a DT developed in AS can be attributed to the comprehensive pneumatic workshop offered within the software. This workshop facilitates the creation and simulation of complex pneumatic systems by providing an extensive library of pre-built components, accurate physics modelling, parameter customisation, and the ability to integrate with other domains such as electrical, hydraulic, and control systems.

Detailed models of pneumatic systems can be created using the pneumatic workshop to accurately represent the real-world behaviour. The customisation of various parameters ensures that the digital twin closely matches the characteristics of the physical components. At the same time, powerful simulation and analysis tools enable the visualisation, issue detection, and optimisation of the system's performance.

To have increased this project's complexity in the implementation of the virtual commissioning, the cylinder that lifts the robot chassis (U/D cylinder) was controlled in position because a pneumatic cylinder in a specific position is less usual than the simple extension and retraction control. It is assumed that the pressure drops across the piston are small compared with the supply pressure and may be neglected. Thus, the input for the block diagram of the cylinder is the airflow, and the output is the cylinder stroke, Figure 7. Equation (1) presents the relation between the volumetric flow rate ($q(t)$), piston surface (A), and the rod's velocity $\dot{x}(t)$:

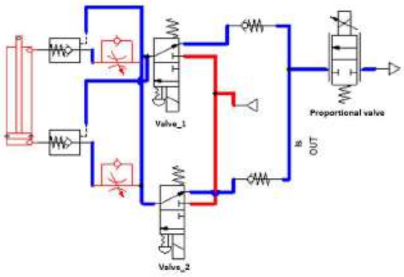
$$q(t) = A\dot{x}(t), \tag{1}$$

from where the cylinder position results from Equation (2):

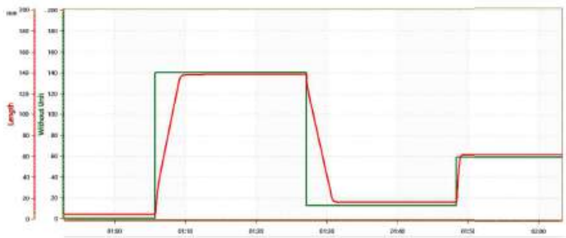
$$x(t) = \frac{1}{A}\int q(t)\,dt, \tag{2}$$

In AS, a servo pneumatic system was implemented for the physical behaviour of the U/D cylinder, with the main goal of accurately controlling the position of the cylinder's rod. The proportional valve is responsible for managing the airflow exhausted from the cylinder. It allows one to make precise adjustments to the cylinder's motion by varying the airflow according to the control signal from the feedback controller. The process begins when the controller receives a desired setpoint for the position of the cylinder's rod. The ultrasonic position sensor continuously measures the actual position of the rod and sends this information back to the controller.

The controller calculates the error between the desired setpoint and the actual position, using this information to generate a control signal to adjust the proportional valve. As the proportional valve adjusts the airflow exhausted from the cylinder based on the control signal, the cylinder's rod moves towards the desired position. The position sensor keeps providing feedback to the controller, ensuring that the system maintains a closed-loop control configuration. This enables the servo pneumatic system to combine the benefits of a pneumatic, such as high force and robust components, with the precision and accuracy of a closed-loop control system.



(**a**)



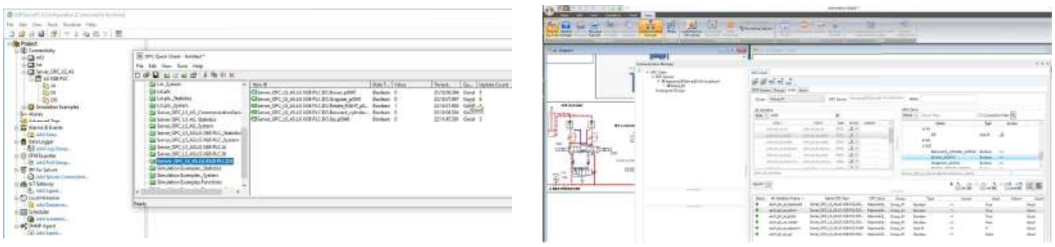
(**b**)

Figure 7. The cylinder (U/D) for two approaches: (**a**) the pneumatic diagram for positioning cylinder in AS; and (**b**) displacement of cylinder for different position references.

The hardware system utilised for conducting the HiL simulation consists of several components, presented in Figure 3. A host computer is included, which runs the DT of the 3DoF robot in an AS environment which the real PLC conducts. The mechanism behind the communication between the mechatronic digital twin and the real PLC is an OPC server. OPC is a popular industrial communication protocol that permits data transmission and compatibility between automation devices and software programs. In this setup, the OPC server (KEPServerEX) links the digital twin and the real PLC in this configuration. Regardless of the implementation specifics or communication protocols the two components use, it offers a common interface that enables seamless communication. KEPServerEX facilitates M2M communications by enabling connections between data values from PLC and AS [39]. The choice of communication protocol is decisive to the successful operation of a DT. The chosen protocol must be well suited to the system's specific requirements, including the network infrastructure, PLC compatibility, real-time performance requirements, etc. Given its robustness and real-time performance, the OPC is an excellent choice for many DT applications, especially mechatronic applications.

Employing an OPC server establishes a vital communication bridge between the digital twin of the mechatronic system and the real-world PLC. Firstly, the OPC server builds a connection with the PLC, enabling a two-way communication pathway. This pathway facilitates the transmission of various data types, encompassing status indicators, sensor-derived measurements, and control commands. The OPC server adeptly converts data originating from the PLC into a format compatible with and readily interpretable by the digital twin. This conversion mechanism accommodates the diversity of data formats and protocols potentially employed by different PLCs, thereby establishing a universal communication language. Subsequently, the OPC server orchestrates the data transfer to the digital twin. Through this exchange, the digital twin becomes a mirror, accurately reflecting the state of the real-world mechatronic system. Furthermore, the OPC server also relays control directives from the digital twin to the PLC, effectively empowering the digital twin to guide the operational state of the real system. The OPC server diligently ensures rapid data delivery to the digital twin, thereby maintaining congruence between the digital twin and the real system. Hence, the OPC server's role is pivotal in simplifying and streamlining the communication and data translation processes between the digital twin and the PLC. By adhering to the OPC standard, the server guarantees seamless interfacing capabilities with any compliant system. This results in the more efficient, flexible, and robust development and utilisation of digital twins across diverse mechatronic systems.

In the project, communication with the PLC model XBC-DR30SU occurs via an Ethernet communication module, specifically the EMTA configured as a Modbus TCP server. The process involves creating a generic channel (because the model of PLC is not in the standard library of KEPServerEX and adding the device connected through its IP address. Initially, all inputs and outputs are defined in the server (Figure 8a, and subsequently, they are set up in the client within AS. AS has the capability to connect to an OPC Server, which simplifies the process of pairing OPC tags with sensors and solenoid valves from the virtual model (Figure 8b). The DT can effectively communicate with the real-world PLC by establishing this connection. This simulation makes it easy to understand the impact of different parameters (flows, pressures from pneumatic circuit) in the behaviour robot to make the finest adjustments to synchronise the robot 1:1 with its DT. In a pneumatic system, the flow rate significantly affects the speed at which the pneumatic cylinder operates. A higher flow rate means more air can fill the cylinder per unit of time, resulting in faster cylinder actuation and vice versa.



(**a**) (**b**)

Figure 8. Configuration of the OPC mechanism: (**a**) the screenshot from the KEPServerEX—defining the OPC server; and (**b**) the screenshot from the AS—the OPC client linked with the variable from DT.

To ensure that the DT of the pneumatic robot accurately replicates the real-world system, its behaviour must be validated. The validation process entails a series of steps that compare the DT's performance with that of the actual system, ensuring that the DT effectively represents the real-world system's behaviour. In the industrial domain, the accuracy of a DT is of paramount importance, often even more so than in academia. This is because industries rely on DTs to make informed decisions regarding optimisation, control strategies, predictive maintenance, training and safety, and last but not least, customisation and personalisation. All these things have significant financial, operational, and safety implications.

A Systematic Approach Validation Process for DT

A systematic approach validation process for DT involves several interconnected stages that work together to improve the DT's accuracy and reliability:

- **Establish performance metrics**—it is important to establish key performance indicators (KPIs) for the modelled system and prioritise them based on their significance for the specific application. This prioritisation helps focus the validation process on the most critical aspects of the system's performance.
- **Set-up working scenarios—**this step involves defining the working scenarios. A bottom–up approach can be employed, starting with the individual components

of the robot and progressing towards the entire system. This method ensures a comprehensive understanding of the system at each level of complexity.

- **Running scenarios for the pair: the robot and its DT—**the defined scenarios are run for both the robot and its DT, and the performance metrics are recorded. This step provides valuable data about the behaviour of the DT under different conditions, which can be compared with the real-world system performance.
- **Comparative analysis between the real robot's behaviour and that of the DT**—by examining the differences in performance metrics, areas of discrepancy and improvement can be identified. The DT can be considered validated if the discrepancies are within acceptable tolerances. Otherwise, it is necessary to take the next step.
- **Calibration and adjustments of the DT**—adjust the parameters and settings of the DT to match the real-world system. This may include tuning the pneumatic actuator's characteristics, control logic, and environmental conditions. By performing multiple iterations, the parameters are adjusted until the DT accurately represents the real system's behaviour.
- **Resumption of tests from the third step**—once the DT has been recalibrated, the working scenarios are run again with the updated values. This step validates the changes made during recalibration and ensures that the DT more closely represents the real system.
- **Iterate and improve**—in this last stage, the validation of the DT is understood, as is continuously updating the DT with new data from the real system and adjusting its parameters accordingly, ensuring that the DT remains a reliable and effective representation of the real system. This iterative process is vital for the DT to remain a valuable tool for the design, simulation, optimisation, and control strategy testing tasks.

Employing this systematic approach streamlines the validation process for the DT pneumatic robot, ensuring its accurate representation of the real-world system and enhancing its efficacy in design, simulation, and optimisation tasks. Although the validation model has been specifically applied to a pneumatic robot, it can be generalised to accommodate other mechatronic systems with the DTs created in diverse software platforms that encompass physical properties while allowing access to the analytical and dynamic aspects. This versatility expands the scope of the validation process, making it applicable to a diverse array of mechatronic systems featuring DTs.

The sequence corresponding to the graph depicted in Figure 9 was carried out at the macro level for both the robot and its DT.

The robot returns to its homing point when the user presses the "Start" button. In the home pose configuration, the robot has the gripper closed, rotated to the left, completely backward, and has the cylinder responsible for lifting the robot chassis positioned at 10 mm. In the following state of the graph, the robot transitions from its home pose to a new state in which the cylinder responsible for lifting the robot chassis (U/D cylinder) raises it to its maximum height of 200 mm. This movement represents a change in the robot's position and a step towards executing the defined sequence. After reaching the maximum height, the next step in the sequence involves the bilateral F/B cylinder moving forwards. This action represents a further change in the robot's position and continues the execution of the defined sequence.

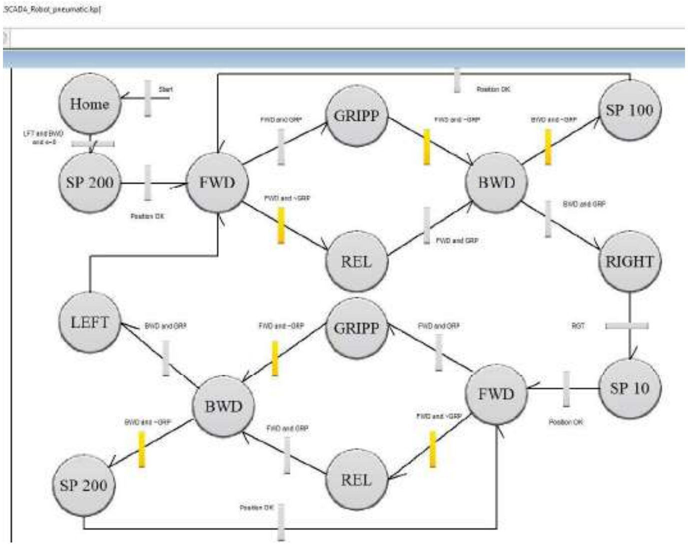
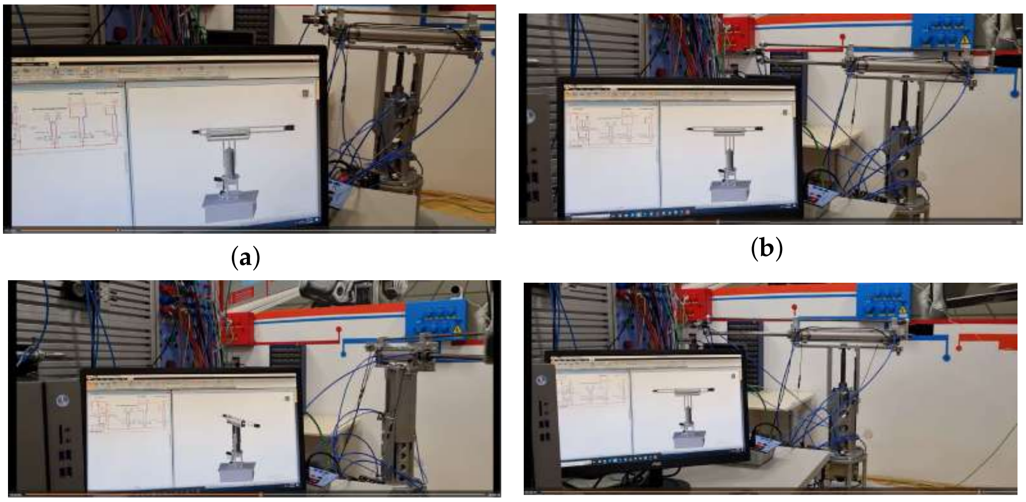


Figure 9. Working scenario implemented for testing.

Figure 10 captures snapshots at various moments during the execution of the working scenario, with the real robot in the background and its digital twin displayed on the monitor in the foreground. These snapshots provide a visual comparison of the robot's behaviour and the digital twin's performance throughout the scenario, further highlighting the accuracy and effectiveness of the digital twin in replicating the real system.



(a) (b)

(c) (d)

Figure 10. Snapshots at various moments during the execution of the working scenario. (**a**) t = 12 s; (**b**) t = 23 s; (**c**) t = 32 s; and (**d**) t = 57 s.

A rough analysis can be conducted by observing the movements of the real robot and its digital twin in parallel. This qualitative comparison identifies any major discrepancies in motion and behaviour, providing an initial assessment of the DT's accuracy in replicating the real system. This approach is particularly feasible for cylinders with a full stroke, as their motion is more easily observed and compared between the real robot and the DT. In the first three captures from Figure 10, good alignment and synchronisation between the real robot and its digital twin can be observed. However, in the last capture at t = 57 s, it can be noticed that the cylinder in the simulation is lagging behind the real one. This discrepancy indicates that there may be some differences in the parameters or timing between the digital twin and the real system that need to be addressed in order to improve the accuracy of the DT's representation. The first action to be taken is to modify the air flow rate entering the cylinder, as it is known that the cylinder's speed primarily depends on the airflow rate. By adjusting the airflow rate in the DT's parameters, the simulation can more accurately represent the real system's behaviour, leading to better synchronisation and alignment between the real robot and its DT. This adjustment should be made based on the observed discrepancy and the known relationship between the air flow rate and cylinder speed to minimise the motion difference between the real system and its digital twin. AS provides access to the internal parameters of the flow control valves, as shown in Figure 11. This feature allows users to easily adjust the airflow rate and the other characteristics of the components within the pneumatic system, ensuring that the digital twin's behaviour more closely matches that of the real robot. By fine-tuning these parameters, the user can adjust the speed cylinder.

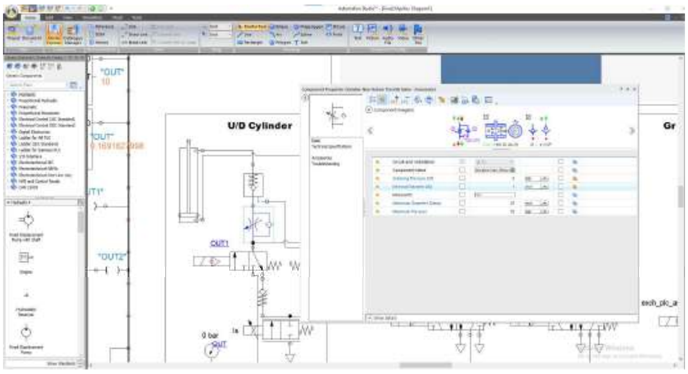


Figure 11. The screenshot from AS—unidirectional flow regulator properties.

Considering the proposed KPIs, a more detailed and quantitative analysis follows to ensure the accurate alignment of the DT with real-world robot performance. During the scenario execution, data are extracted into a graphical format to analyse the real robot's behaviour and DT. This visualisation allows for a more comprehensive understanding of their performance, allowing the comparison and identification of any discrepancies between the two systems.

In Figure 12, the behaviour of the U/D cylinder is represented for both the actual robot (derived from monitoring the PLC programming software for the values received from the ultrasonic distance sensor) and its DT (taken from the position graph specific to the cylinder in AS) for various reference values imposed in the working scenario. This comparison directly assesses the DT's accuracy in replicating the robot's behaviour. It helps identify potential areas for improvement and calibration to ensure a more faithful representation of the natural system.

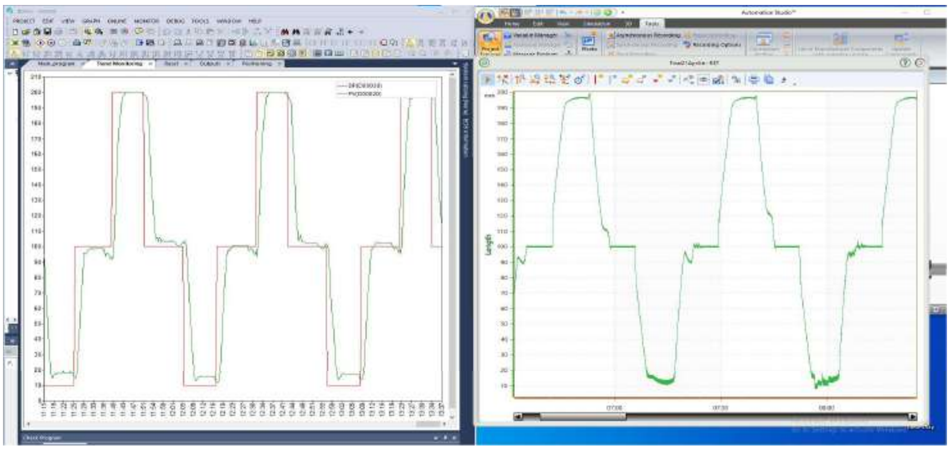


Figure 12. The real and mirrored behaviour of the U/D cylinder.

The refinement and analysis of the two graphs can be conducted within a dedicated computational environment tailored for data processing and visualisation. Widely utilised tools for this purpose include MATLAB or Python and spreadsheet applications such as Microsoft Excel or Google Sheets. Importing the data from both graphs into the selected software allows a comprehensive examination, comparison, and visualisation of the robot's behaviour and DT.

Communication delay, also known as latency, between the DT and the real system is an important aspect affecting the system's performance. Low latency can be a key issue for achieving real-time performance in a HiL simulation. High latency, on the other hand, may lead to a time lag between the execution of control actions on the DT and their implementation on the existing system, potentially affecting the overall system performance and accuracy. The performance of the host computer where the DT from AS is implemented can significantly influence the communication delay or latency. A computer with high performance will be able to run the DT more smoothly, leading to less delay and better real-time performance.

Upon completing the entire process, an operational DT is obtained. To preserve its effectiveness and reliability, continuous updates are imperative. By systematically updating the DT with new data from the real system and adjusting its parameters as needed, the DT remains a precise and trustworthy representation of the real system. This ongoing process of iteration and improvement is crucial for the DT to serve as a valuable instrument while adapting to alterations and evolving demands in the real-world system.

5. Implementation and Interaction with DT in Unity Ecosystem

By leveraging the advantages of the DT implementation using the features provided by AS, one can enhance their understanding and expertise in the system's operation. Transitioning to the Unity platform enables an MR experience to seamlessly integrate the real robot and its digital counterpart. It offers an immersive environment that further strengthens the comprehension of the system's behaviour and functionality. The immersive environment allows the development of unique and personalised experiences with the DT, which involve and engage the user at a deeper level than in the case of traditional methods.

Unity is a powerful game engine and development platform that is primarily used for creating high-quality 3D games and interactive simulations. While Unity is not designed for creating DTs, its 3D modelling and interactivity capabilities can certainly be leveraged for that purpose. DTs are virtual replicas of physical systems. Unity can provide a powerful tool for creating and exploring those replicas in a highly immersive and interactive way, where the external environment can influence DT behaviour. By creating a DT influenced by external factors, it is possible to understand better how the system operates under different conditions and make more informed decisions about its design and operation.

The DT of the pneumatic robot was constructed by importing components from a CAD model with the $.FBX$ format and then configuring its rotational or translational movements within the Unity engine. In this process, the CAD model was likely converted into a series of meshes and materials that were imported as `GameObjects` in Unity. Such `GameObjects` were then assembled and configured to create the DT of the robot, including its movements and interactions with the environment. The game engine comprises numerous working contexts, or "scenes"—as shown in Figure 13—where the DT is positioned and configured according to the intended simulation-related parameters. The `GameObjects`, which has been put on the robot, can later change a wide range of properties, including the location, orientation, size, weight, material, and animations. The `GameObjects` is then animated using *C#* scripting. *C#* provides a clear, powerful, and robust environment to develop the required codes. Nonetheless, it also has the advantage that most VR, AR, and MR hardware-related packages are already available in this programming language.

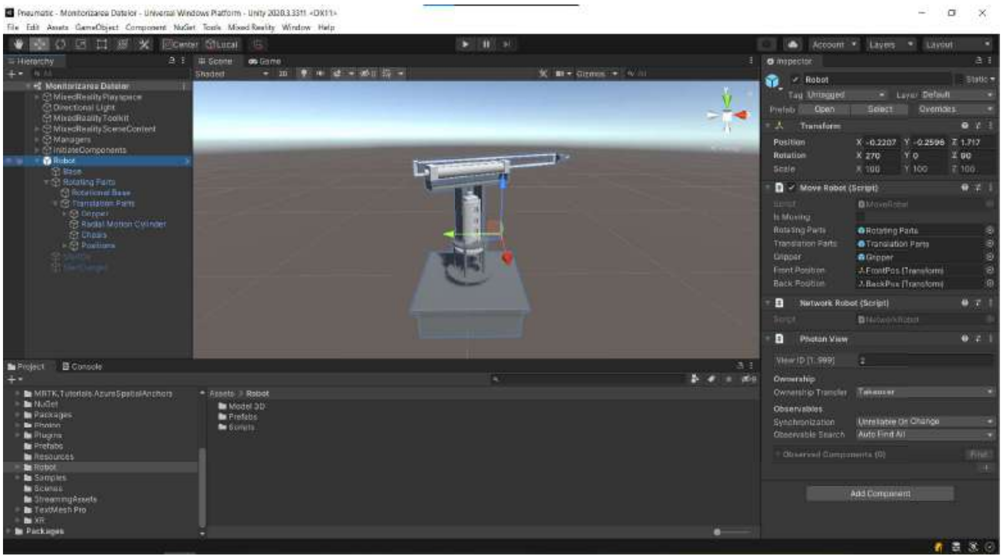
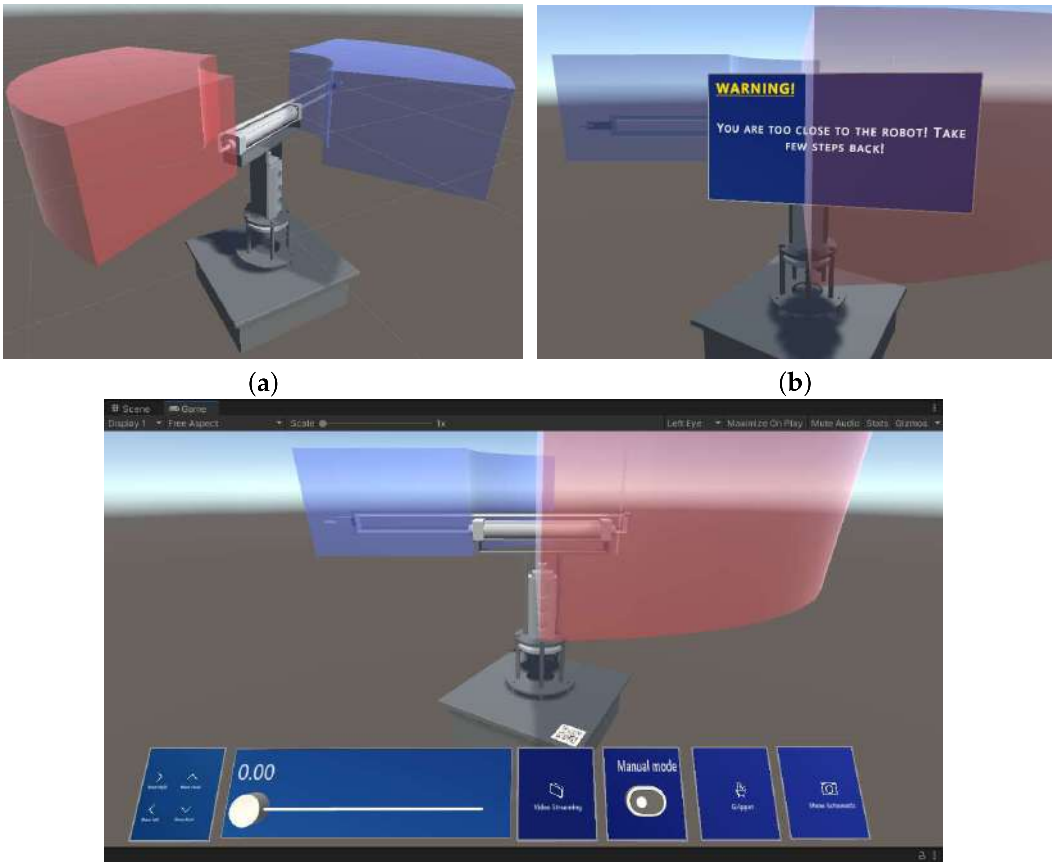


Figure 13. The screenshot from the scene in Unity 3D and the proprieties of the robot's DT.

In mixed-reality experiences involving real-world robots, safety is a critical consideration, especially in the case of non-collaborative robots. The DT representation of the robot and its environment can be used to create virtual boundaries and limits corresponding to the robot's physical workspace and safety zone, Figure 14a. These virtual boundaries and limits were programmed based on the movement of the real-world robot, ensuring that they accurately represent the current workspace and safety zone. The working space for the cylindrical robot is a hollow quarter-cylinder with the height determined by the stroke of the U/D cylinder. The difference between radii equals the stroke of the second prismatic cylinder (F/B). The user is alerted when they are about to enter a potentially dangerous zone. This is performed by displaying a warning message through haptic feedback, Figure 14b. At the same time, when the user enters these dangerous areas, the real-world robot is programmed to stop its operation until the user automatically exits the area. Figure 14c depicts the menu that accompanies the robot in MR.

5.1. Defining Moving

The robot's DT can perform various movements such as the real robot using the `RoboMove` *C#* script. This script defines the behaviour of a robotic arm with rotating and translating parts in Unity. It includes several methods for initiating movements in different directions, moving the arm's gripper to specific positions, and updating the arm's position in the scene. The script includes an enumeration called `Position`, which represents the two possible states for the arm's rotating part: `Right` and `Home`. The class `MoveRobot` has several public methods such as `StartMovementFront()`, `StartMovement-Back()`, `StartMovementRight()`, `StartMovementLeft()`, `TranslateUpDown()`, `MoveFront()`, `MoveBack()`, `MoveRight()`, and `MoveLeft()`. The class also has a public variable, *isMoving*, a flag that prevents the robot from starting another movement before the current movement is completed. The script defines the actuator ranges, including the maximum and minimum radii, angle, and height, which control the arm's movements. All these values are in link with the values of strokes of pneumatic actuators. The `Start()` method initialises variables and sets the arm's initial position to "Home"—similarly to the position defined for the robot in the scenario from the HiL simulation from AS. Because of the rotation and translation moving out of the *F*/*B* cylinder, it is necessary to define the equivalent end-of-stroke sensors (for this, the *Position* flag is necessarily near *Home*). The `Update()` method checks for movement initiation flags and calls the appropriate movement method based on which flag is set.



(**a**) (**b**)

(**c**)

Figure 14. The screenshot of the MR application in HoloLens 2: (**a**) the working space—a hollow quarter-cylinder (blue colour space)—and the predefined safety zone linked to the F/B cylinder which has a bilateral construction (red colour space); (**b**) intrusion incident into the safety area; and (**c**) holographic menu that accompanies the robot in the scene: push buttons—MoveRight/MoveLeft/MoveFront/MoveBack/VideoStreaming/Manualmode/Gripper/ShowSchematic and a slider for lifting the robot chassis.

In Listing 1, the constant *desiredDuration* corresponds to the time required by the real robot for the L/R actuator to complete its entire travel. This is closely related to the flow rate and pressure values at the air inlet port of the real actuator. The mathematical function *Mathf.SmoothStep* models cylinder cushioning, which is needed to lower the speed of the cylinder before it reaches the end cap. The function in Unity is used to interpolate between two values smoothly, creating a natural transition ideal for animations or object movements. Based on a cubic formula, this function generates a smoother and more natural result than simple linear interpolation.

5.2. Pose of the DT in Virtual Space

In the context of real-world interaction in the mixed-reality application for the non-collaborative robot, the dimensions of the DT must match the dimensions of the real-world robot so that the safety zones and warnings are consistent for users. Therefore, the DT must be accurately configured with the exact dimensions and precisely anchored to the position and orientation of the real robot.

If the CAD files are incomplete or inaccurate, the DT may not be a faithful replica of the real-world object. The first option is to modify the file in CAD software and reimport it into Unity. However, Unity provides tools for accurately measuring distances, angles, and other properties of `GameObjects` that can help adjust the DT. If it is accurately configured with the same dimensions, the next step is to anchor the hologram to the space so that there is a perfect overlap between the robot and the hologram. Since HoloLens 2 comes with the native ability to read QR codes, the positioning and orientation of the hologram will use the position of the QR code on the real robot, which is fixed to the robot pedestal. Using the add-on *Adjust Pivot* in Unity positions the reference point of the robot model at the position of the QR code, and this allows the position of the QR code to be used as a reference point for the position and orientation of the robot in the Unity scene. Doing so ensures the accurate overlapping of virtual objects relative to real objects in the environment.

Listing 1. MoveRight method from RobotMove class.

```
public void MoveRight()
//MoveRight method
{
    if (currentPositon != Position.Right)
    {
        isMoving = true;
        elapsedTime += Time.deltaTime;
        var percentageComplete = elapsedTime / desiredDuration;

        rotatingParts.transform.localRotation = Quaternion.Lerp(Quaternion.
            Euler(rotatingParts.transform.localEulerAngles.x, rotatingParts.
            transform.localEulerAngles.y, rotatingParts.transform.
            localEulerAngles.z),
            target, Mathf.SmoothStep(0, 1, percentageComplete));
        if (rotatingParts.transform.localRotation.eulerAngles == target.
            eulerAngles)
        {
            elapsedTime = 0;
            currentPositon = Position.Right;
            startRight = false;
            isMoving = false;
            Debug.Log(currentPositon);
        }
    }
    else
    {
        startRight = false;
    }
}
```

The implementation of this functionality in the actual project, which starts from scanning the QR code, identifying the robot's DT and positioning the robot's hologram was achieved through several scripts grouped in the QRTracking namespace. The `QRCodesManager.cs` script initialises and manages the QR code tracking system, which uses the Microsoft.MixedReality.QR library to detect and track the QR codes in the scene. The `QRCodesVisualizer.cs` script visualises the detected QR codes in the scene, using a specified prefab object as a template for each QR code object. It subscribes to various events raised by the `QRCodesManager` script, such as `QRCodeAdded`, `QRCodeUpdated`, and `QRCodeRemoved`, which allow it to add, update, and remove the QR code objects in response to changes in the tracking data. The `QRCode.cs` script is attached to each QR code object and is responsible for displaying the relevant information about the QR code, such as its ID, text, physical size, and timestamp. It also handles user input, such as clicking on the QR code object, and launches the associated URI if it is valid. The script also contains a code for positioning a virtual pneumatic object (the DT of the robot) in the scene, which represents the position and orientation of the QR code relative to the real-world robot—as shown in Figure 15.

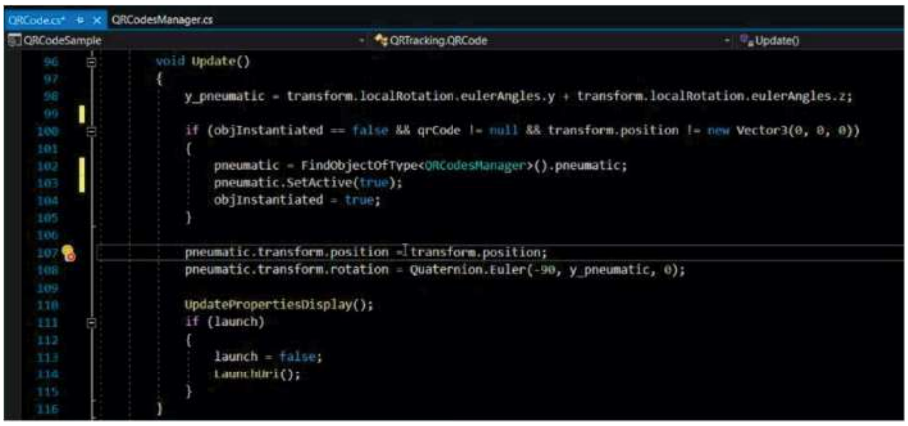


Figure 15. The `Update()` method is responsible for matching the position and orientation of the DT robot with the QR code object.

5.3. A Virtual Space for Multiple Users to Interact with Robot

Enabling multiple users to simultaneously view and interact with a robot in a shared virtual environment can effectively showcase its capabilities and facilitate collaboration, especially in training or educational contexts where the use of a DT is required. This was made possible using Photon Unity Networking (PUN), a Unity plugin that enables network synchronisation and communication for multiplayer games and applications [40]. By broadcasting data from the PUN server, the robot's and users' DT are synchronised across all connected devices. Any user input or interaction with the robot triggers an RPC event callback on all associated devices via the PUN server. This ensures that the robot's state is synchronised across all participants in the shared environment.

The process of integrating the Photon SDK for Unity in a DT project to establish a shared virtual environment is depicted in Figure 16. The procedure involves importing the SDK, configuring the server settings, creating a Singleton pattern-based NetworkManager to facilitate connections to the Photon server, adding a lobby to manage room creation and joining, instantiating a `Robot` prefab equipped with pertinent components to enable user interactions, devising a `Robot` prefab script to manage object parenting and ownership transfer.

The implementation of the right side of the diagram in Figure 16 is shown in Figure 17.

The `MoveRobot` script discussed in Section 5.1 allows for the precise and controlled movement of a robotic arm in a Unity scene, with the ability to synchronise movement across multiple instances of the game object using Photon networking.

5.4. The System for Remote Support

By incorporating shared DT technologies, teams can collaborate remotely to solve challenging problems in real-time without being nearby. This is feasible in both industry and academia. Leveraging the features of shared DT technology, a video-streaming concept incorporating distinct roles for technical and non-technical individuals was designed to further improve collaboration and learning experiences. In situations where non-technical individuals need to execute intricate tasks, real-time expert guidance via videostreaming can be invaluable. The expert (professor) can observe the non-technical (student) individual's actions and provide immediate feedback, ensuring that the task is performed correctly and safely.

Figure 18a presents the user interface menu, where users are prompted to select their role as either "technician" or "non-technician." This simple, intuitive menu is designed to quickly identify and categorise users based on their technical expertise, enabling a more streamlined and targeted communication experience within the system. By making this selection, users are guided to the appropriate functionality tailored to their role, ensuring that the subsequent interactions are suitable for their level of knowledge and experience.

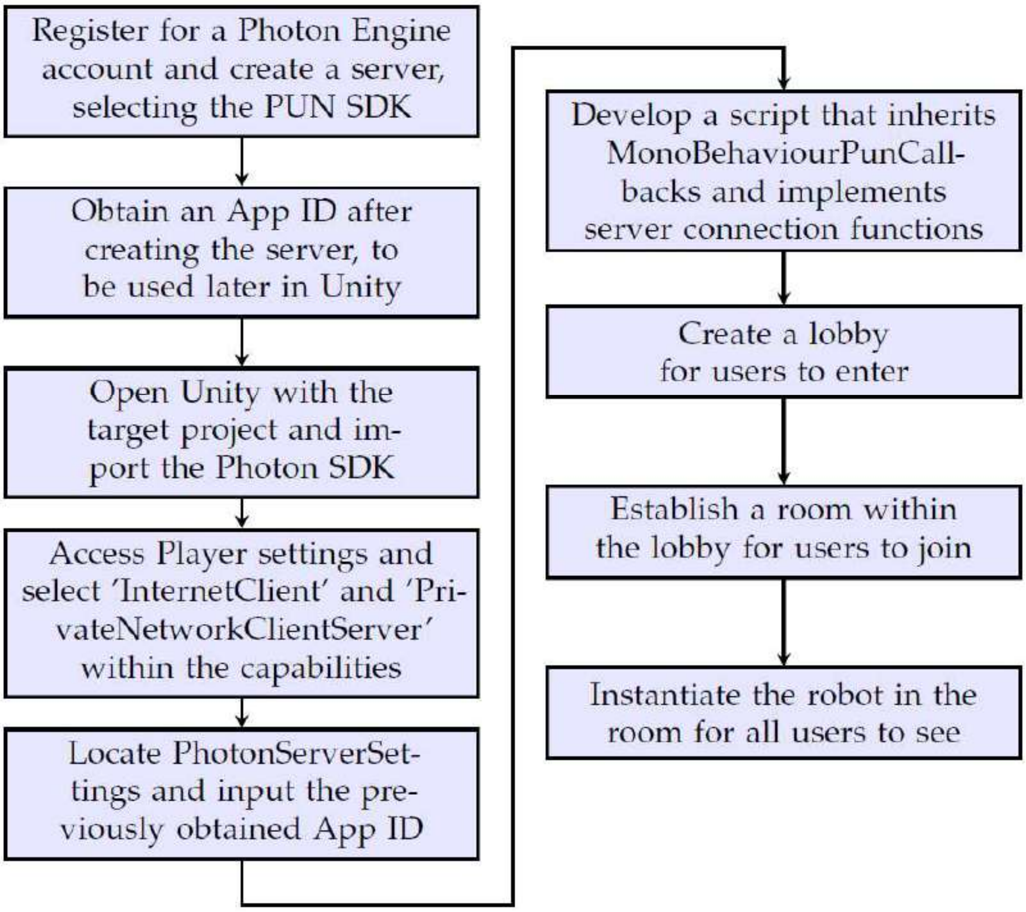
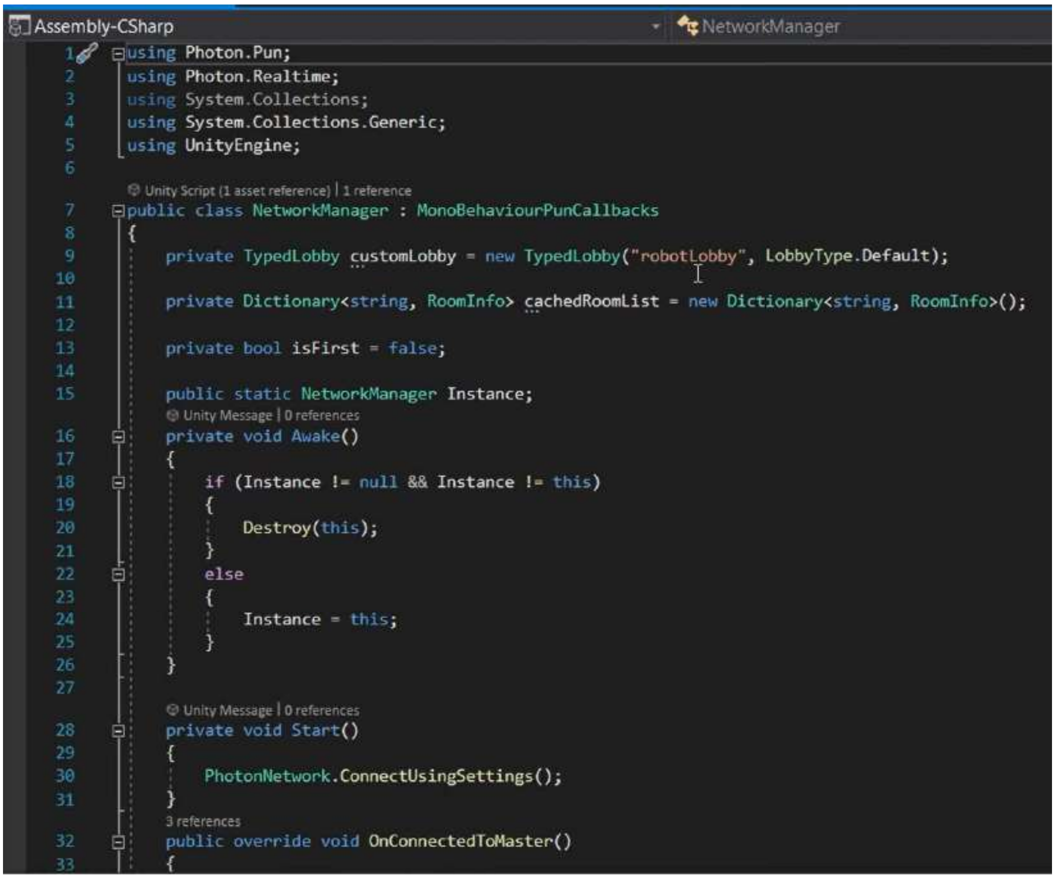


Figure 16. Procedure for developing an application with PUN.

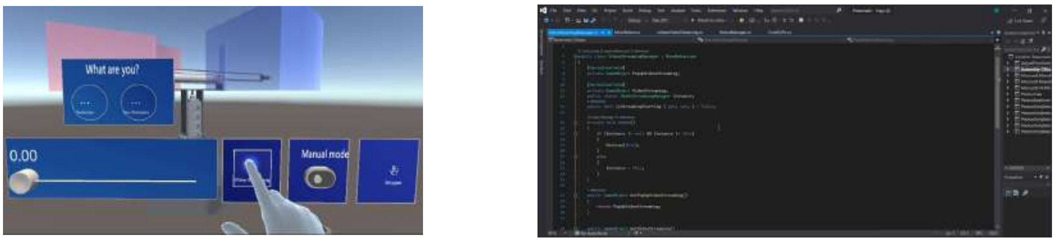
`MixedReality-WebRTC` was used for the implementation with a DSS server made in NodeJS. The server allows the communication of packets between glasses. Two instances with different IDs must be created for the network to be able to detect them. Communication can only be achieved if the two instances are connected to the server. Creating a manager with a script (in Figure 18b) to return the pop-up with the choice of role and the window where the received image will be. This manager will have a single instance; thus, it can access the objects returning in other scripts. The system establishes an audio and video connection between technicians and non-technicians, allowing for the transmission of video and sound feeds between both parties. In the implemented set-up, the technician can view the non-technician's perspective via video, while the non-technician can hear the technician's instructions through audio. This one-way video and two-way audio setup ensure privacy for the technician (which may be the case when the technician uses a laptop, not HMD) while still enabling them to offer valuable assistance.

Figure 19 shows sequences from an experiment in which two operators interact with the robot's hologram synchronised with the real robot. The film is integrated into the PIP technique so that it is possible to see two perspectives: the real one and the perspective seen by the operator who chose their role as a technician. The projection capture on the technician's HoloLens uses the facilities offered by mixed-reality Capture—Microsoft.



```csharp
using Photon.Pun;
using Photon.Realtime;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class NetworkManager : MonoBehaviourPunCallbacks
{
    private TypedLobby customLobby = new TypedLobby("robotLobby", LobbyType.Default);

    private Dictionary<string, RoomInfo> cachedRoomList = new Dictionary<string, RoomInfo>();

    private bool isFirst = false;

    public static NetworkManager Instance;
    private void Awake()
    {
        if (Instance != null && Instance != this)
        {
            Destroy(this);
        }
        else
        {
            Instance = this;
        }
    }

    private void Start()
    {
        PhotonNetwork.ConnectUsingSettings();
    }
    public override void OnConnectedToMaster()
    {
```

Figure 17. C# script `NetworkManager`—implementing a lobby defining a dictionary for all rooms from the lobby and connection of users through AppID.



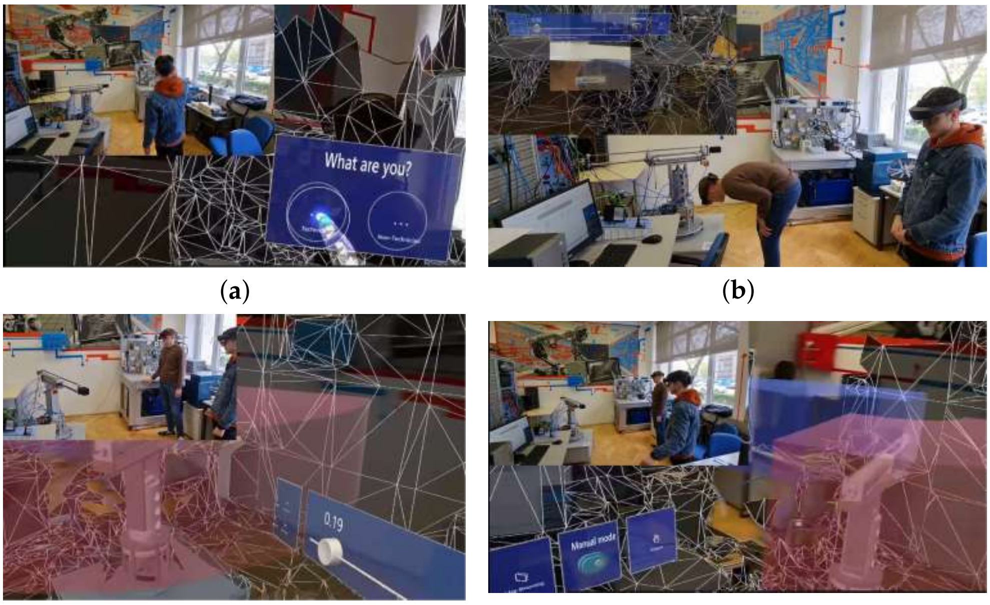
(**a**) (**b**)

Figure 18. Remote assistance platform that enables the technician and non-technician to communicate effectively: (**a**) the choosing of the role; and (**b**) the implementation by the manager.

5.5. PLC Integration in Unity Ecosystem

In order to connect the DT of the robot to the real PLC, a remote connection was created using the `GateManager` facility from Secomea to link the laptop with the PLC. The laptop was configured to redirect any requests received on a port 502 to the PLC. This enabled the direct transmission of information from the HMD to the PLC registers through the laptop. `NModBus` was used for communication with the PLC.

To establish a connection with the PLC, the method `StartConnection()` in the `ConnToPlc` script was marked with [PunRPC], indicating that it is an RPC method that can be called by all users. Similarly, all methods marked with [PunRPC] can be forced to be called by all users. The function `CloseConnection()` in the `ConnToPlc` script was used to close the connection.



(**a**) (**b**)

(**c**) (**d**)

Figure 19. Snapshots at various moments during the execution of the working scenario from MR application in HoloLens2 and real actions: (**a**) choosing the technician role; (**b**) scanning the QR code by a non-technician under the careful observation of the technician through the video streaming facility; (**c**) the positioning of the cylinder that raises the robot chassis to 19% of the stroke using the slider; and (**d**) the technician takes control. It can be seen that both safety zones on the hologram are superimposed in the space above the real robot.

When the manual button is pressed, the `Toggle()` function in the `ConnToPlc` script is called. If the connection is successful, the button turns blue, but there is a chance it will turn blue even if the connection is not successful, and the button will need to be pressed again. The function SetToggleToOthers visually sets the button as pressed for the other participants.

After the connection has been established, the position of the real robot is queried from the PLC so that the hologram of the robot can be positioned accordingly. The `Update()` function in the `RobotManager` script is called at each frame to check whether the robot is in the scene. If the robot appears, it is assigned to a variable, and if the manual button has been pressed and activated, the real robot's position is queried and synchronised with the hologram, and the synchronisation flag is set to true. All commands are sent to all other participants in the room, so their hologram robot should move accordingly.

To move the robot in manual mode, when it pushes the `MoveRight` button, the photon calls the `OnMethodCalledRight()` method, which calls another method `MoveRight()` from `ConnToPlc`. Through the `PhotonNetwork` mechanism, the `OnMethodCalledRight()` method sets the parameter to "true" for all participants, which checks whether the action comes in manual mode and if the movement action comes from pressing the button. If these conditions are met, the PLC receives a signal, and the real robot executes the movement to the right as shown in Figure 20.

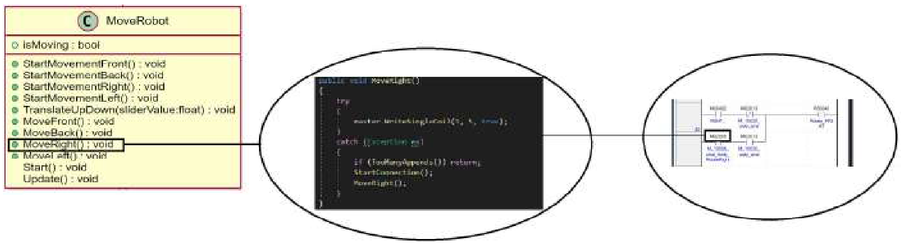


Figure 20. The mechanism for the right robot movement began with Unity—with the textttMoveRobot class, which includes the `MoveRight()` method, and from here, set the M2005 Modbus address as responsible for activating the output of the PLC that energises the solenoid valve coil via the *ModbusBitWrite* bridge.

5.6. The PlantUML Diagram

Figure 21 visualises the system's structure and behaviour, enabling a deeper understanding of the DT's robot project in the Unity development environment. The relationship between *MonoBehaviour*, *MonoBehaviourPun*, and the other classes in your *PlantUML* code is based on inheritance. Custom classes inherit from *MonoBehaviour* or *MonoBehaviourPun* to access the respective features provided by the Unity engine and Photon Unity Networking library, enabling them to interact with the Unity scene and synchronise networked *GameObjects* in a multiplayer setting.

While only some classes have been examined in detail in the article thus far, this overview provides a general understanding of each class's roles in the development and interaction with the robot's DT from this implementation. The diagram effectively illustrates how the classes interact with one another, either through inheritance or other relationships, and the methods and properties associated with each class.

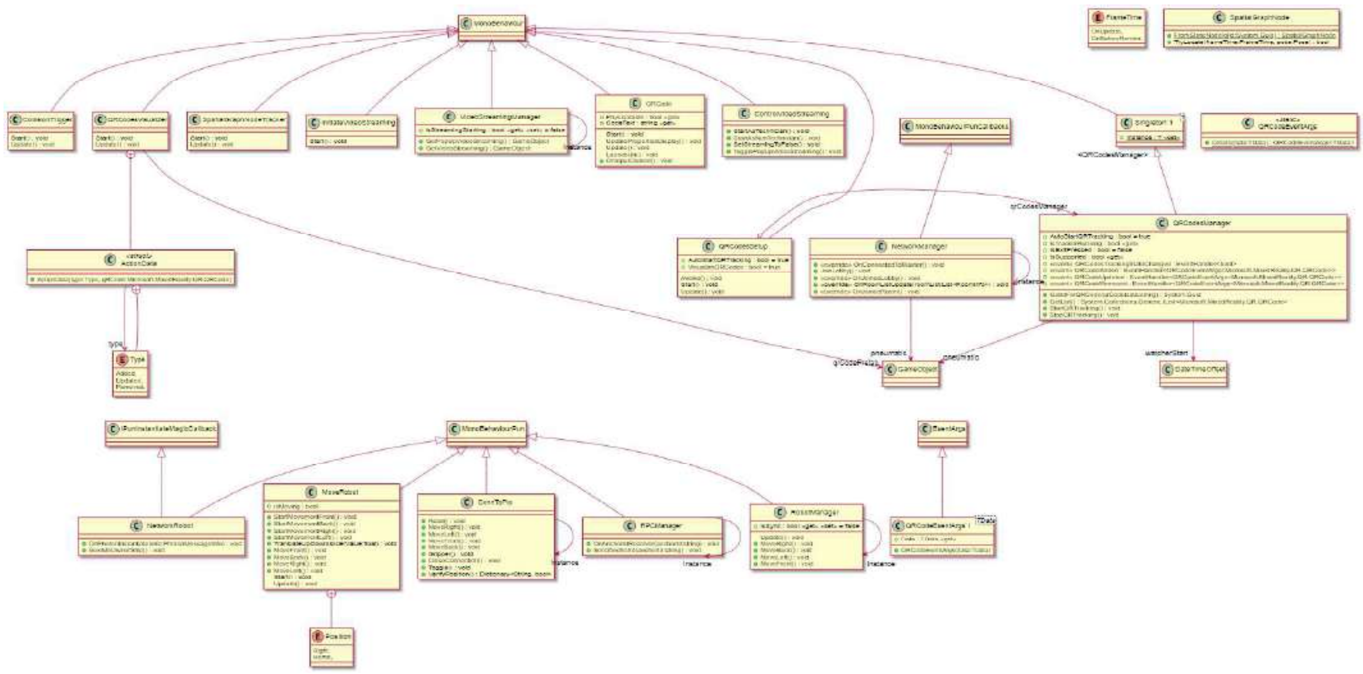


Figure 21. The PlantUML diagram of the pneumatic robot project in the Unity development environment.

5.7. The Challenges of Using Unity in DT

While the Unity Engine boasts an excellent platform for visualising and interacting with digital twins (DTs) using a variety of HMDs such as Microsoft's HoloLens 2, it presents some challenges. Primarily designed as a game engine, Unity might not natively support complex engineering simulations, necessitating the integration of external libraries or tools. This increases the complexity of the DT development process within Unity.

Moreover, Unity's steep learning curve may pose a barrier, especially for those without a computer programming background, as scripting in languages such as C# are often a requirement for Unity applications.

Unity can also encounter performance limitations when dealing with highly detailed DTs or data-heavy real-time systems. The task of rendering sophisticated 3D models or conducting extensive real-time simulations might strain Unity's capabilities, particularly when hardware resources are limited. Additionally, Unity may face compatibility issues when interfacing with other industry-standard hardware or software tools. For instance, establishing real-time communication between Unity and specific industrial controllers, such as PLCs, could require additional middleware or customised development.

Furthermore, managing large volumes of real-time data, a cornerstone in DT operations, may necessitate a sophisticated data processing system to guarantee the prompt and accurate updating of the DT.

Nevertheless, despite these challenges, the Unity community continuously evolves and develops solutions. This continuous advancement helps to uphold Unity's position as a valuable tool in the ever-growing field of DTs development.

6. Discussion

The development of DTs for mechatronic systems can be approached differently depending on the software used. AS and Unity each have unique strengths and challenges, which this article aims to compare in creating DTs for the same mechatronic system.

In AS, the operating characteristics of pneumatic equipment, such as actuators, valves, and sensors, are mainly predefined. These predefined elements make it easier for students, engineers and developers to create realistic simulations of pneumatic systems, as they can directly use the built-in components and characteristics. AS is specifically designed for simulating, designing, and validating complex automation systems, streamlining the modelling process in a particular 2D environment.

On the other hand, Unity is a general-purpose game engine and development platform, which means that it is not tailored for simulating pneumatic systems out of the box. In order to create a DT in Unity, developers need to implement various mathematical functions, physics simulations, and other techniques to accurately model the behaviour of pneumatic components. This inconvenience can be more challenging and time-consuming, as developers have to create these models from scratch. Despite the challenges, Unity's flexibility and extensibility can be advantageous in some situations. For instance, developers can create custom visualisations, integrate the DT with other systems, or even deploy the simulation to various platforms, including virtual and augmented reality environments. This level of versatility allows for more creative and innovative applications, although it may come at the cost of a steeper learning curve and increased development time.

One potential approach to overcome these challenges is to leverage the knowledge gained from working with AS. By initially using AS, the persons involved can understand the underlying phenomena and the behaviour of pneumatic components. This knowledge can be transferred to the development process in Unity. Once the developers in Unity have a clear grasp of the requirements and the desired dynamic behaviour of the system, they can more effectively search for and implement various techniques and tricks to achieve realistic simulations.

This sequential approach can help bridge the gap between the specialised knowledge of pneumatic systems and the flexible development environment offered by Unity. By leveraging the expertise from both domains, creating high realistic and accurate DTs is possible while benefiting from Unity's versatility and creative potential.

The DT model's complexity affects the latency, so more intricate and detailed models necessitate more computational resources and processing time, increasing the latency. On the other hand, simpler models, although they might reduce the latency, may also compromise the accuracy of the DT's representation of the physical system. Experiments from this project with the implementation of DT in AS and Unity showed differences in latency. Specifically, the DT implemented in AS exhibited more delay than Unity. This observation can be attributed to the different computational demands of these two environments, emphasising the simulation environment's role in influencing latency. A relevant trade-off arises in this context: the balance between the fidelity of the model representation and real-time performance. Reducing the complexity of the model can decrease the latency, improving the real-time representation. However, this simplification can also reduce the accuracy of the digital twin. Therefore, it is important to consider this trade-off depending on the specific requirements and goals of the project. the impact of latency on the DT's performance reinforces the need for a balanced approach in DT development. Decisions regarding the model complexity, choice of simulation environment, and computational resources should consider both the desired accuracy and real-time performance of the DT.

In summary, the choice between AS and Unity for developing the DTs of pneumatic systems depends on the specific requirements and goals of the project. The AS platform offers a more streamlined and specialised approach, while Unity provides greater flexibility and versatility, albeit with a potentially higher degree of complexity. The comparison is a valuable resource for engineers, developers, and researchers looking to create DTs for mechatronic systems, helping them make informed decisions about the best tools and techniques for their projects.

The advantage of using a hologram projected by a head-mounted display (HMD) makes implementing a DT in Unity well worth it, especially when the DT has a solid connection to the actual dynamic behaviour of the process. By employing HMDs such as the HoloLens 2, students and professionals can interact with the DT in a highly immersive and intuitive manner, providing them with a unique learning experience. This immersive experience can enhance the understanding and retention of complex concepts related to mechatronic systems, as users can directly visualise and manipulate the DT in a three-dimensional space. Furthermore, holograms can foster collaboration among students and professionals, as they can share their virtual environment and work together to understand and improve the DT's behaviour.

7. Conclusions

The article highlights the differences and potential advantages of developing DTs in AS and Unity, showcasing the challenges and opportunities inherent in each development environment. The approach is multidisciplinary (automation and computer science, mechanical engineering, electrical engineering, and human–computer interaction), combining knowledge and methods from various fields to develop the efficient DT of the 3DoF pneumatic robot. By integrating expertise from both AS and Unity, this paper showcases the potential of interdisciplinary collaboration in addressing the complex challenges and achieving innovative solutions in DT development. The paper effectively amalgamates the best of both worlds by creating an initial, accurate DT in AS and then expanding and enhancing it in Unity. AS was used for accuracy, clarity, and reliability in simulating pneumatic systems and was exploited by Unity for its versatility, creativity, and capacity to deliver an immersive, interactive experience. This combined approach results in a DT that accurately represents its real-world counterpart, offers improved user engagement, and has broader applicability.

One future research direction is to investigate the possibility of creating hybrid frameworks that combine the strengths of both AS and Unity. Such frameworks allow for the streamlined development of DT with a focus on pneumatic/hydraulic systems while also providing the flexibility and versatility of Unity. This approach could lead to more efficient development processes and novel applications.

The experience gained from developing a digital twin in both Automation Studio and Unity can be highly valuable for students and professionals alike. By working with these two platforms, learners can comprehensively understand the modelling and simulation process and the challenges and solutions unique to each environment. This immersive experience can enhance the understanding and retention of complex concepts related to mechatronic systems, as users (especially students who belong to the category of digital natives) can directly visualise and manipulate the digital twin in a three-dimensional space.

An accurate DT enables industries to minimise costly downtime, reduce waste, improve efficiency, and ensure product quality. Moreover, it allows for the early identification and mitigation of potential issues, resulting in increased reliability and reduced maintenance costs. In safety-critical applications, an accurate DT can be vital in preventing accidents and ensuring the well-being of workers and the environment.

Funding: The research leading to these results has received funding from the European Union's HORIZON-CL4-2022-TWIN-TRANSITION-01-01 research and innovation actions under grant agreement R3GROUP Project GA no. 101091869, and by a research grant of the TUIASI, project number GI-TD-DigitAll-7/2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TUIASI	Technical University "Gheorghe Asachi" from Iasi
DT	Digital Twin
CPS	Cyber-Physical Systems
I4.0	Fourth Industrial Revolution, Industry 4.0
IT	Information Technology
IoT	Internet of Things
IIoT	Industrial Internet of Things
AI	Artificial Intelligence
DevOps	Development and Operational Cycles
XR	Extended Reality
MR	Mixed Reality
VC	Virtual Commissioning
PLC	Programmable Logic Controller
RPP	Revolute Prismatic Prismatic
OPC	Open Protocol Communication
DoF	Degree of Freedom
AS	Automation Studio conceived by Famic Technologies Inc.
EOAT	End of Arm Tooling
M2M	Machine-to-Machine
SiL	Software in the Loop Simulation
HiL	Hardware in the Loop Simulation
PUN	Photon Unity Networking
RPC	Remote Procedure Call
HMD	Head-Mounted Display
HL	HoloLens
URI	Uniform Resource Identifier

References

1. Dosoftei, C.C.; Lupu, A.; Pascal, C.M. A new approach to create a realistic virtual model of a cylindrical robot using Automation Studio. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *591*, 012078. [CrossRef]
2. Liagkou, V.; Stylios, C.; Pappa, L.; Petunin, A. Challenges and Opportunities in Industry 4.0 for Mechatronics, Artificial Intelligence and Cybernetics. *Electronics* **2021**, *10*, 2001. [CrossRef]
3. Gerber, A. What Does Industry 4.0 Mean? The Definition of Digitization. Available online: https://www.wfb-bremen.de/en/page/bremen-invest/what-does-industry-40-mean-short-definition (accessed on 10 December 2022).

4. Grieves, M.; Vickers, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*; Springer International Publishing: Cham, Swizerland, 2017; pp. 85–113. [CrossRef]
5. Park, J.S.; Lee, D.G.; Jimenez, J.A.; Lee, S.J.; Kim, J.W. Human-Focused Digital Twin Applications for Occupational Safety and Health in Workplaces: A Brief Survey and Research Directions. *Appl. Sci.* **2023**, *13*, 4598. [CrossRef]
6. Madni, A.M.; Madni, C.C.; Lucero, S.D. Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems* **2019**, *7*, 7. [CrossRef]
7. Steindl, G.; Stagl, M.; Kasper, L.; Kastner, W.; Hofmann, R. Generic Digital Twin Architecture for Industrial Energy Systems. *Appl. Sci.* **2020**, *10*, 8903. [CrossRef]
8. Tao, F.; Qi, Q.; Wang, L.; Nee, A. Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering* **2019**, *5*, 653–661. [CrossRef]
9. Cortés, D.; Ramírez, J.; Villagómez, L.; Batres, R.; Vasquez-Lopez, V.; Molina, A. Digital Pyramid: An approach to relate industrial automation and digital twin concepts. In Proceedings of the International Conference on Engineering, Technology and Innovation (ICE/ITMC), Cardiff, UK, 15–17 June 2020; pp. 1–7. [CrossRef]
10. Attaran, M.; Celik, B.G. Digital Twin: Benefits, use cases, challenges, and opportunities. *Decis. Anal. J.* **2023**, *6*, 100165. [CrossRef]
11. Wang, Z. Digital Twin Technology. In *Industry 4.0-Impact on Intelligent Logistics and Manufacturing*; IntechOpen: London, UK, 2020; Chapter 7. [CrossRef]
12. Lee, J.; Azamfar, M.; Singh, J.; Siahpour, S.; Siahpour, S. Integration of digital twin and deep learning in cyber-physical systems: Towards smart manufacturing. *Int. J. Precis. Eng. Manuf. Smart Technol.* **2020**, *2*, 34–36. [CrossRef]
13. Hu, W.; Hu, W.; Zhang, T.; Deng, X.; Liu, Z.; Liu, Z.; Tan, J.; Tan, J. Digital twin: A state-of-the-art review of its enabling technologies, applications and challenges. *J. Intell. Manuf. Spec. Equip.* **2021**, *2*, 1–34. [CrossRef]
14. Gamble, J. Digital Twins: Bridging the Physical and Virtual Worlds. Available online: https://www.ericsson.com/en/about-us/new-world-of-possibilities/imagine-possible-perspectives/digital-twins/ (accessed on 8 February 2023).
15. Saracco, R. Digital Twins' Future. Available online: https://cmte.ieee.org/futuredirections/2021/01/26/digital-twins-future/ (accessed on 2 April 2023).
16. Saravanan, S.K.; Muthusenthil, B.; Gurusubramani, S. A Review of Digital Twin Leveraging Technology, Concepts, Tools and Industrial Applications. In Proceedings of the 1st International Conference on Computational Science and Technology (ICCST), Chennai, India, 9–10 November 2022; pp. 742–748. [CrossRef]
17. Constantinescu, C.; Popescu, D.; Todorovic, O.; Virlan, O.; Tinca, V. Methodology of Realising the Digital Twins of Exoskeleton-Centred Workplaces. 2018. Available online: https://atna-mam.utcluj.ro/index.php/Acta/article/view/1026/955 (accessed on 10 February 2023)
18. Mincă, E.; Filipescu, A.; Cernega, D.; Șolea, R.; Filipescu, A.; Ionescu, D.; Simion, G. Digital Twin for a Multifunctional Technology of Flexible Assembly on a Mechatronics Line with Integrated Robotic Systems and Mobile Visual Sensor—Challenges towards Industry 5.0. *Sensors* **2022**, *22*, 8153. [CrossRef] [PubMed]
19. Moiceanu, G.; Paraschiv, G. Digital Twin and Smart Manufacturing in Industries: A Bibliometric Analysis with a Focus on Industry 4.0. *Sensors* **2022**, *22*, 1388. [CrossRef] [PubMed]
20. Dosoftei, C.C.; Lupu, A.; Mastacan, L. Real-Time Communication between Automation Studio and PLC Based on OPC Technology for Control 3-DoF Robot. In Proceedings of the 24th International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 1493–1496. [CrossRef]
21. Innovation through Digital Transformation. Available online: https://digital-innovation.zone/en/home/ (accessed on 12 February 2023).
22. Romania-Digital Innovation Hub. Available online: https://projects2014-2020.interregeurope.eu/ruralsmes/news/news-article/12598/romania-digital-innovation-hub/ (accessed on 4 February 2023).
23. Ugarte Querejeta, M.; Illarramendi Rezabal, M.; Unamuno, G.; Bellanco, J.L.; Ugalde, E.; Valor Valor, A. Implementation of a holistic digital twin solution for design prototyping and virtual commissioning. *IET Collab. Intell. Manuf.* **2022**, *4*, 326–335. [CrossRef]
24. Süß, S.; Strahilov, A.; Diedrich, C. Behaviour simulation for virtual commissioning using co-simulation. In Proceedings of the 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg, 8–11 September 2015; pp. 1–8. [CrossRef]
25. Ugarte, M.; Etxeberria, L.; Unamuno, G.; Bellanco, J.L.; Ugalde, E. Implementation of Digital Twin-based Virtual Commissioning in Machine Tool Manufacturing. *Procedia Comput. Sci.* **2022**, *200*, 527–536. [CrossRef]
26. Kucak, D.; Juricic, V.; Juričić, V.; Dambic, G. Machine Learning in Education—A Survey of Current Research Trends. In Proceedings of the 29th Daaam International Symposium on Intelligent Manufacturing and Automation, Zadar, Croatia, 24–27 October 2018; pp. 406–410. [CrossRef]
27. Beilby, A. Top 4 Applications for Digital Twins. Available online: https://virtualcommissioning.com/top-4-applications-for-digital-twins/ (accessed on 15 April 2023).
28. Ryalat, M.; ElMoaqet, H.; AlFaouri, M. Design of a Smart Factory Based on Cyber-Physical Systems and Internet of Things towards Industry 4.0. *Appl. Sci.* **2023**, *13*, 2156. [CrossRef]

29. Apv, R. Virtual commissioning and digital twin of yaskawa motoman robotic cells based on the industry 4.0 context. *Int. Robot. Autom. J.* **2021**, *7*, 35–38. [CrossRef]
30. Dumitrascu, A.; Nae, L.; Predincea, N. Virtual Commissioning as a Final Step in Digital Validation of the Robotic Manufacturing Systems. *Proc. Manuf. Syst.* **2014**, *9*, 215–220.
31. Liu, Y.K.; Ong, S.K.; Nee, A.Y.C. State-of-the-art survey on digital twin implementations. *Adv. Manuf.* **2022**, *10*, 1–23. [CrossRef]
32. Zamfirescu, I.; Pascal, C. Modelling and simulation of an omnidirectional mobile platform with robotic arm in CoppeliaSim. In Proceedings of the 24th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 8–10 October 2020; pp. 667–672. [CrossRef]
33. Mixed Reality Documentation. Available online: https://learn.microsoft.com/en-us/windows/mixed-reality/ (accessed on 10 February 2023).
34. Künz, A.; Rosmann, S.; Loria, E.; Pirker, J. The Potential of Augmented Reality for Digital Twins: A Literature Review. In Proceedings of the Conference on Virtual Reality and 3D User Interfaces (VR), Christchurch, New Zealand, 12–16 March 2022; pp. 389–398. [CrossRef]
35. Kim, H.I.; Kim, T.; Song, E.; Oh, S.Y.; Kim, D.; Woo, W. Multi-scale Mixed Reality Collaboration for Digital Twin. In Proceedings of the International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Bari, Italy, 4–8 October 2021; pp. 435–436. [CrossRef]
36. Orsolits, H.; Rauh, S.F.; Estrada, J.G. Using mixed reality based digital twins for robotics education. In Proceedings of the International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Singapore, 17–21 October 2022; pp. 56–59. [CrossRef]
37. Valles, J.; Zhang, T.; McIntosh, P.; Pacilli, M.; Nataraja, R. Assessment of Core Surgical Skills Using a Mixed Reality Headset—The MoTOR Study. *J. Med. Syst.* **2022**, *46*, 102. [CrossRef] [PubMed]
38. Yang, C.; Tu, X.; Autiosalo, J.; Ala-Laurinaho, R.; Mattila, J.; Salminen, P.; Tammi, K. Extended Reality Application Framework for a Digital-Twin-Based Smart Crane. *Appl. Sci.* **2022**, *12*, 6030. [CrossRef]
39. KEPServerEX: One Data Source for your Industrial Automations. Available online: https://www.ptc.com/en/products/kepware (accessed on 19 February 2023).
40. The Ease-of-Use of Unity's Networking with the Performance & Reliability of Photon Realtime. Available online: https://www.photonengine.com/pun/ (accessed on 8 March 2023).

The Immersive Mixed Reality: A new opportunity for experimental labs in engineering education using HoloLens 2

Constantin-Cătălin Dosoftei

"Gheorghe Asachi" Technical University of Iasi
Department of Automatic Control and Applied Informatics
D. Mangeron 27, Iasi, 700050, Romania

constantin-catalin.dosoftei@academic.tuiasi.ro

Abstract. The paper addresses a new direction for experimental labs in engineering education, which will facilitate a systemic approach to the educational process within the application hours of many technical disciplines and this will create new perspectives for in-depth understanding and implementation of the Industry4.0 paradigm for engineers in the control area but not only. This paper aims to present the flow for translating two pneumatic drive workstations into mixed reality on the HoloLens2 device.

Keywords: Mixed Reality, experimental lab, pneumatic drives, education, Industry 4.0

1 Introduction

Engineering is closely related to science, technology, and practice - it is the generator of products and processes that contribute to the progress of humanity to the increase the quality of life.

The complex challenges of this complicated period, hopefully post-pandemic, require new creative, innovative, and holistic solutions within the training/education process within the technical universities. Because engineering is a practising profession, laboratory experiments are considered an imperative component of engineering programs by both parties (student-teacher) involved in the education process. In the learning process within technical universities, two main directions must be combined harmoniously: the theory behind the specific phenomena and the practical experience. However, there is a paradox in engineering that is strongly impacted by rapid technological changes. This is mainly because, in this area, it becomes relatively difficult to adopt a relevant infrastructure of educational equipment for training facilities at levels similar to those of current industries. The existing infrastructure is at a different level than every student in a study group could have access during the laboratory class to its experiment.

adfa, p. 1, 2011.

Using of mixed reality systems, those that overlay virtual reality on top of physical reality, represents a possibility to reduce this gap by creating much more intuitive, engaging experiences with a much higher impact for the generation of digital natives, a generation with increased skills in the use of digital technology from an early age.

As a result of mixed reality, students can touch and manipulate objects, generating a better understanding. Students can also interact with datasets, complex formulas, and abstract concepts in an animated form that is sometimes harder to understand through the teacher's classic instructions.

The training of future engineers will have to be subjected to ensuring consistent analytical skills based on the fundamental principles of science, systemic thinking for understanding complexity, practical ingenuity, and creativity.

The paper is structured as follows. Section 2 presents the current works using HoloLens 2 in mixed reality. Section 3 describes in the first part the hardware components used in the realisation of the two applications of this paper and the pneumatic workstations. The second part of this presents the software tools used. Section 4 discusses application deployment, from 3D modelling of pneumatic components and panels to create interactive animations and buttons in Unity. There is also a QR-code link to a video with the proof of the applications. Finally, Section 5 concludes the established purpose of the paper and the next steps in improving the two mixed reality applications.

2 Related work

Immersive learning has become an integral part of quality education in many areas. Academic institutions and industry leaders are looking to cutting-edge research support technologies to prepare learners for the success of digital literacy of the future. Universities are increasingly looking to innovative emerging technologies, such as augmented reality (AR), virtual reality (VR), and mixed reality (MR). The main reason for this is to increase the efficiency of the educational process, improve communication and drive collaboration between actors involved to deliver great student experiences during lab hours and not only.

The computer game industry and the digitisation process specific to the Industry 4.0 paradigm have influenced the development of extended reality systems. The terminology in link with this eXtended Reality (XR) can sometimes be confusing, meaning the entire way from real life to the virtual environment [1]. The aim of Table 1 is to clarify these tech terminologies.

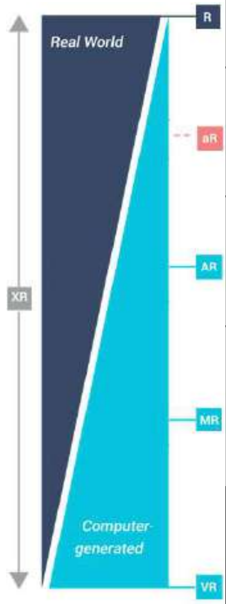
These emerging technologies are different, and the instruments differ.

The foundations of these concepts appeared two decades ago when the way from the real world to the computer-generated environment was in an early stage of development [2], and the existing tools were with incomparable performance.

Nowadays, the types of equipment used in XR have become more and more varied [3,4], the spearhead for the mixed reality area being the Ho-loLens2 system, a head-mounted display (HMD) produced by Microsoft.

The fields strongly impacted by mixed reality technology both from an academic and an applied perspective, are healthcare and medical assistance, architecture and civil engineering, defence, life science, tourism, marketing, manufacturing and automotive, and last but not least, automation control. Considering the applications made for HoloLens 2, the field of medicine keeps the headline. The spectrum contains training in anatomy, examinations of patients, and the collaboration between doctors worldwide in times of surgery. [5-8].

Table 1. The layers of eXtended Reality

Tech terminology	Representation	Description layer
Reality (R)		Physical reality
Assisted Reality (aR)		It used, in special, to add information into the operator's peripheral vision, through a different type of screens
Augmented Reality (AR)		Anchoring new digital objects on the real world in which the user can see and feel the new updated environment
Mixed Reality (MR)		One step further comparatively with AR, more powerfully, an environment where the physic objects and digital objects co-exist and more important interact in real-time in a hybrid environment
Virtual Reality (VR)		Completely artificial three-dimensional environment, a computer-generated simulation where the users is transformed into an avatar in this virtual world

In the current field of interest of the actual paper-control engineering, the applications made so far for mixed reality using HoloLens2 are from the perspective of the impact that information and communication technology (ICT) must increase the productivity and flexibility of production in the context of the new industrial revolution - I4.0. The connection between HoloLens2 glasses and robotic systems is a relatively developed topic [9, 10] in link with conveying the motion intent from robot to user or vice versa. Another class of applications with a high impact is using technology for remote assistance, visual inspection and maintenance of complex machines during the production process [11].

3 System architecture

The proposed goal from this study refers to the direct interactions and visualisation of the operation of the two pneumatic workstations by the students on the HoloLens2 device, connected wirelessly to a laptop.

This section depicts the tools this research requires to create the pneumatic circuits, obtain the 3D model components, write the code, and finally implement and display them on the HoloLens 2 device.

3.1 Hardware components

In the Pneumatic and Hydraulic Control Systems Lab from the Department of Automatic Control and Applied Informatics from TUIASI, there are a few workstations (with Camozzi pneumatic components) controlled by a PLC and HMI panel [12]. Future engineers must acquire technical skills and be flexible in adapting and learning to operate pneumatic equipment in different circuits. The way to touch these aims is to instruct them in the "hands-on" concept using these educational workstations. The configurations of the two circuits implemented on workstations, implemented in section 4 from the actual paper in mixed reality applications, are in Fig. 4. and Fig. 5.

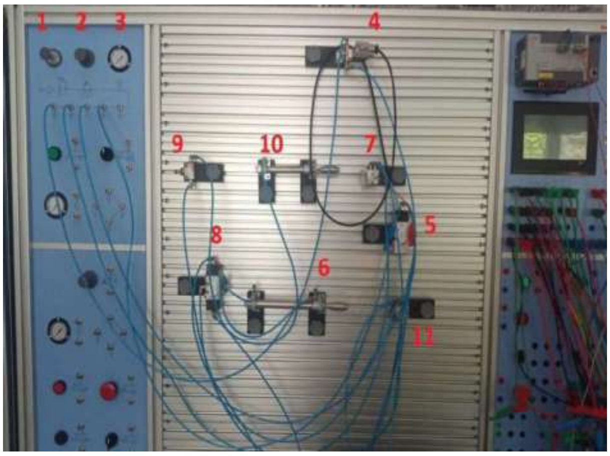


Fig. 4. Implementation of the first pneumatic circuit on real workstation

The configuration of the pneumatic circuit from Fig. 4. contains on the panel follow equipment: 1 - Lockable isolation valve 3/2-way; 2 - Pressure regulator; 3 - Manometer; 4 - Solenoid valve 3/2-way; 5,8 - Pneumatically operated valves 5/2-way; 6 - Double-acting cylinder with end-of-travel braking; 7,11 - Roller operated valves (spring return) 3/2-way; 9 - Variable throttle valve; 10 - Magnetic single-acting cylinder.

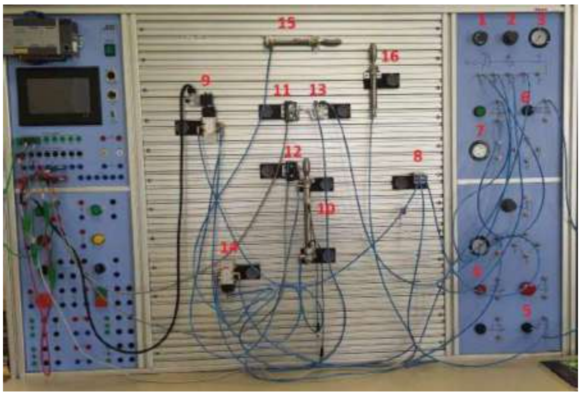


Fig. 5. Implementation of the second pneumatic circuit on real workstation

Also, the components that are found on the second pneumatic system from Fig. 5. are the following: 1 - Lockable isolation valve 3/2-way; 2 - Pressure regulator; 3,7 - Manometers; 4,5,6 - Manually operated valves with different types of action; 8 - Logic element OR; 9,14 - Solenoid valve 5/2-way; 10 - Magnetic double-acting cylinder; 11,12 - Roller Limit Switch the cylinder stroke end; 13 - Mechanically operated valve; 15,16 - Magnetic single-acting cylinder.

Next, the head-mounted display HoloLens2 device will briefly describe its technical characteristics. HoloLens2 is defined by Microsoft, [13], as an “untethered holographic computer” which use the Windows Holographic operation system. The device is a mixed reality headset that can be declared as a hands-free controllable computer with wireless connectivity controlled by voice or gestures.

The basic specs of the HoloLens 2 are: resolution- 2K 3:2 light engines in each eye, holographic density: >2.5K light points per radian, processor: Qualcomm Snapdragon 850, holographic unit: 2nd-generation, wireless connectivity: IEEE 802.11ac, Bluetooth 5.0, camera: 8MP stills, 1080p video, mics: 5-channel, speakers: built-in, spatial audio, eye and head tracking, rechargeable battery 16.5Wh, 6-degrees-of-freedom (6DoF) tracking through an inertial measurement unit. The main components are depicted in Fig. 6.

The HoloLens2 hardware contains a few cameras: four grey-scale environment tracking cameras and a depth camera to sense its environment and capture the gestures of the human user. The depth camera uses active IR illumination to obtain depth through time-of-flight.

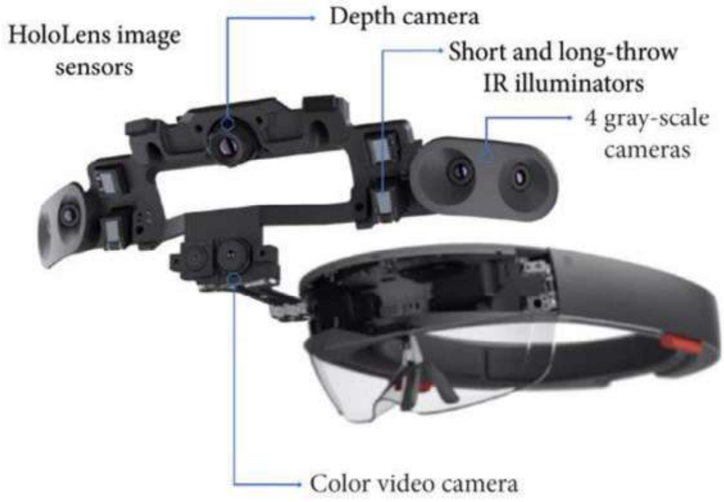


Fig. 6. Main components of Microsoft HoloLens 2, [6]

3.2 Software

To create MR applications was necessary to use notions from various fields. It started from the pneumatics working principle of each piece of equipment from the workstation, 3D modelling equipment using SolidWorks and Blender software, programming notions in using functions in Visual Studio later assigned to Unity objects, and the implementation of holograms in Unity.

After identifying the components on each workstation, the pneumatic circuits are created in the Automation Studio software, which allows the design of non-standard pneumatic equipment based on adjustments of predefined components [14].

The body of the panels and pneumatic components identified in Section 3.1 was 3D modelled using the SolidWorks 2016 program and then saved with the .stl extension. As there are also dynamic elements whose movements want to be highlighted in applications, they were modelled on parts. The parts modelled that way were: single and double-action cylinders, manometers, solenoid valves and roller-operated valves. When each pneumatic part was designed, the dimensions specified in the Camozzi catalogue were considered.

The next step was to add as realistic textures as possible to each component. To add the real textures specific to virtual workstations is necessary to import the assembled panels with the .stl extension in Blender software. After creating, texture was exported in .fbx format, which Unity recognises.

The most complex step was to design the virtual scenarios and add interaction functions to the pneumatic components imported into Unity. The steps necessary to create the two MR applications are shown in the next section.

Also, the Microsoft Visual Studio software was used, in which all the encodings necessary for the Unity objects were made. The programming language in which most HoloLens functions are written in C#. After completing the applications in Uni-ty, they are compiled from Visual Studio and uploaded to the HoloLens 2 device via USB or wireless. Fig. 7. depicts the workflow to develop the two holographic applications for HoloLens 2.

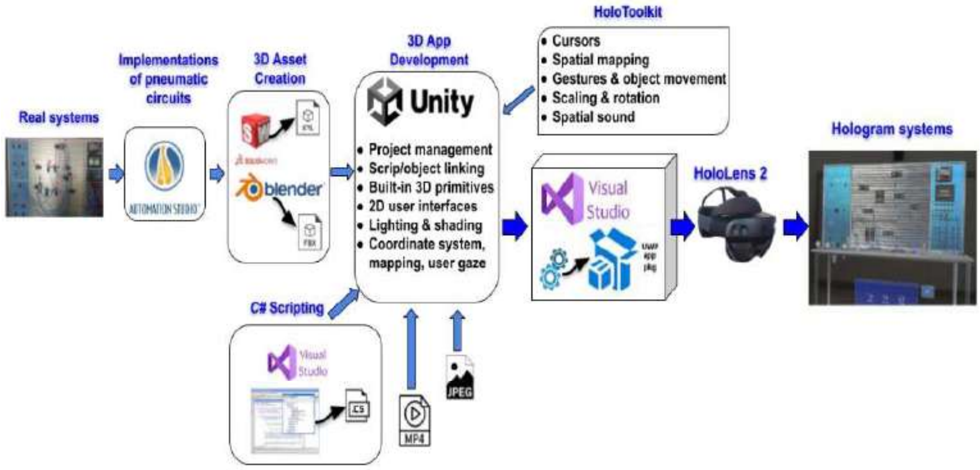


Fig. 7. Development workflow

4 Implementation of pneumatic circuits in Mixed Reality

The two applications created in this paper differ from each other. In the first application, the user can interact with the hologram to assemble the circuits on the two pneumatic stands, starting from scratch. In the second project, the pneumatic components are assembled by default; the student can visualise the operation of the panels in parallel at the same time.

As specified in Section 3.2, the project's most important and complex part was the creation of hologram-type pneumatic panels in Unity using Mixed Reality. To use Windows Mixed Reality, the newly created project must be set to be exported as a Universal Windows Platform application. This platform automatically targets any device, including HoloLens 2.

Then, the packages needed for the Unity project found in the MixedRealityFeatureTool executable were added to allow users to update and accelerate the development of Mixed Reality applications.

The next step was when the pneumatic workstations in FBX format were imported into the Unity virtual scene. Because the textures added to Blender could not be retained, they were applied again.

Fig. 8. Pneumatic workstations with added textures

To interact directly with the components on the pneumatic panels, it will be necessary to add to each HandInteraction action, such as Object Manipulator (which includes Constraint Manager), Box Collider and NearInteractionGrabbable. In this way, the user can move, scale, or rotate objects by hand in Unity and on the HoloLens 2 device. Also, the Rigidbody component has been added, which applies the gravitational force to each object. Therefore, when a piece is taken by hand from the table of the panel, and then it is released, the pneumatic component will have a free fall.

To help the user become more familiar with the circuits, labels containing the name of the pneumatic parts have been created. Animation gif files have been added to illustrate the operation principle of the components in the longitudinal section.

An interesting object of this paper is the tool for displaying the catalogue with pneumatic components that can also be found on panels. It will only be displayed when the wrist is turned and the user's gaze turns to the catalogue while wearing the HoloLens 2 device, as depicted in Fig. 9.

Fig. 9. Immersion of specialized literature (catalogue) in virtual environment

The next step was to create an interactive menu that could indicate the locations of each pneumatic part in the circuit or manipulate the components placed on the panel. The following will explain the buttons on the menu: Hints - indicates the place corresponding to each pneumatic part in the assembly, Explode - will remove the fixed components at a set offset, Reset - will place the pneumatic piece in the initial place where they were taken. The menu is represented in Fig. 10.

Fig. 10. The interactive menu

For extra creativity, animations such as translation and rotation movements were made on the cylinders, manometers, roller-operated valves and the flash action of the solenoid valve led. The bottleneck of the paper was the creation of the rotational movement around a fixed point. Because it could not be done directly by changing the rotation angle on the z-axis of the roller, it was necessary to create a special script called "Gizmo", which attaches a transparent sphere to the roller around which it can rotate.

A plus of the project's originality is the one in which it was wanted to activate the animations on manometers, cylinders, valves and solenoid valves when attaching the air tubes to them. Thus, each air tube that connects to these parts will have a "Drag&Drop" script that specifies the location of the tube that will start the animation of the pneumatic part while fixing it in that place.

Finally, the Pause and Resume interactive buttons were created to which I added the Pause/Resume component. Therefore, the created animations will be stopped and resumed by pressing the buttons - Fig. 11.

Fig. 11. Interactive buttons

To be able to display the applications on HoloLens 2, it was necessary to build projects in Unity. A .sln file is generated and opened with Visual Studio following this process. The application was loaded on the device using the computer's IP address connected to the HoloLens2 device. Fig. 12. shows students working with Mixed Reality applications near the pneumatic workstations. While testing the created scenarios, everything the students viewed on HoloLens 2 was displayed on the laptop in the Windows Device Portal by accessing the Live Preview option. A video with a demonstration of the holographic pneumatic workstations is illustrated in a video (https://youtu.be/im3G4kHGgFc or scan QR-code below).

Fig. 12. Testing applications by students and QR code to access the video experiment

5 Conclusions and future work

This paper aims to contribute to facilitating the learning process of future engineers by creating the virtual scenes described and, at the same time, familiarising, capturing their interest and training them in MR tech that will storm the entire production of companies shortly. Integrating digital twin capabilities with MR tech started being used across some industries. The capability to visualise the digital model in the proximity of a real process is an incredible opportunity given by Industry 4.0.©

Educational MR tools in a "hands-on" lab can improve student performance by a significant percentage, giving an opportunity of increasing the experiment's difficulties without compromising the understanding level.

The collaborative mixed reality can be used to transmit procedural knowledge and could eventually replace, in some scenarios, face-to-face training.

The next step is introducing voice commands for ease and more user control over the components. It is also desired to change the structure of the 3D air tubes to give them malleability similar to those in the real environment.

Acknowledgement

The authors thank Tech-Con Industry for their continuous support in developing the Hydraulic and Pneumatic Control Systems Lab from the Department of Automatic Control and Applied Informatics from TUIASI.

References

1. https://www.amaxperteye.com/, accessed May, 29th, 2022
2. Milgram P., Takemura H., Utsumi A., Kishino F., "Augmented Reality: A class of displays on the reality-virtuality continuum", Proc. SPIE vol. 2351, Telemanipulator and Telepresence Technologies, pp. 2351–34, 1994.
3. Parida, K.; Bark, H.; Lee, P.S. "Emerging thermal technology enabled augmented reality", Adv. Funct. Mater. 2021, 31, 2007952
4. Liu Y., Dong H., Zhang L., Saddik A.E., "Technical Evaluation of HoloLens for Multimedia: A First Look," in IEEE MultiMedia, vol. 25, no. 4, pp. 8-18,2018
5. Park, S.; Bokijonov, S.; Choi, Y. "Review of Microsoft HoloLens Applications over the Past Five Years. Appl. Sci. 2021", 11, 7259. https://doi.org/10.3390/app11167259
6. Galati R., Simone M., Barile G., De Luca R., Cartanese C., Grassi G., "Experimental Setup Employed in the Operating Room Based on Virtual and Mixed Reality: Analysis of Pros and Cons in Open Abdomen Surgery", Journal of Healthcare Engineering, vol. 2020, Article ID 8851964, 11 pages, 2020
7. https://www.gigxr.com/blog/roi-in-xr-a-look-at-standardized-patients
8. Hanna, M.G.; Ahmed, I.; Nine, J.; Prajapati, S.; Pantanowitz, L. Augmented reality technology using Microsoft HoloLens in anatomic pathology. Arch. Pathol. Lab. Med. 2018, pp 638–644
9. Hietanen, A.; Pieters, R.; Lanz, M.; Latokartano, J.; Kämäräinen, J.K. "AR-based interaction for human-robot collaborative manufacturing". Robot. Comput.-Integr. Manuf. 2020
10. Gruenefeld, U., Prädel, L., Illing, J., Stratmann, T., Drolshagen, S., Pfingsthorn, M., "Mind the ARm: Realtime visualization of robot motion intent in head-mounted augmented reality", In Proceedings of the Conference on Mensch und Computer, Germany, 2020; pp. 259–266
11. Vorraber, W., Gasser, J., Webb, H., Neubacher, D., Url, P., "Assessing augmented reality in production: Remote-assisted maintenance with HoloLens", Procedia CIRP 2020, pp 139–144
12. Dosoftei C.C., Cojocaru A.E., "Implementation of a Virtual Control Lab to Support Teaching in Engineering Control," 2020 International Conference and Exposition on Electrical and Power Engineering, Romania, 2020, pp 699-703
13. https://docs.microsoft.com/en-us/HoloLens/HoloLens2-hardware, accessed April 9th, 2022
14. https://www.famictech.com/portals/0/PDF/brochure/automation-studio-educational, accessed July 1st, 2022

Digital Twin and Virtual Reality: A Co-simulation Environment for an Educational Hydraulic Workstation

2024 International Conference on Development and Application Systems (DAS) | 979-8-3503-4929-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/DAS61944.2024.10541200

1st Fănică-Valeriu Hrib
Department of Machine Manufacturing Technology
"Gheorghe Asachi" Technical University of Iasi
Iasi, Romania
fanica-valeriu.hrib@student.tuiasi.ro

2nd Andrei-Iulian Iancu
Department of Automatic Control and Applied Informatics
"Gheorghe Asachi" Technical University of Iasi
Iasi, Romania
andrei-iulian.iancu@academic.tuiasi.ro

3rd Constantin-Cătălin Dosoftei
Department of Automatic Control and Applied Informatics
"Gheorghe Asachi" Technical University of Iasi
Iasi, Romania
constantin-catalin.dosoftei@academic.tuiasi.ro

***Abstract*—Hydraulic system comprehension forms the pillar ofnumerous engineering fields; however, its academic assimilation remains intricate. Addressing this, our research introduces a digital twin-enabled virtual hydraulic workstation simulator developed in Automation Studio™. This simulator mirrorsa tangible hydraulic workstation, including its multifaceted components, to enhance the pedagogical paradigm of hydraulic education. We ensure the symbiosis of theoretical understanding and practical application by meticulously modeling 3D hydraulic components and executing analogous simulations. Our key findings spotlight the utility of time-dependent system behavior graphs in bolstering educational insights. The paper underscores the significance of digital twin solutions in university programs, mainly when physical resources are scarce, reinforcing educational technology's potential to facilitate studentengagement in technically dense arenas.**

***Index Terms*—Digital twin, hydraulic equipment, modeling andsimulation, experimental laboratory**

I. INTRODUCTION

Hydraulic systems, underpinned by fluid power mechanisms primarily driven by oil hydraulics, form the linchpin of numerous pivotal applications across various engineering sectors. Their remarkable significance manifests in civil engineering through water resource management, aerospace with precise aircraft actuation, marine systems for adept steering, and robotics for exact actuation. Further, it extends to agricultural machinery, mining ventures, biomedical apparatuses, environmental modeling, and state-of-the-art automotive systems. These systems have not only revolutionized equipment handling through precision control and torque multiplication but have also consistently showcased performance excellence in a multitude of industries. Their widespread adoption, from everyday machinery to the intricacies of aerospace operations, isa testament to their versatility and impeccable efficiency, cementing their foundational stature in driving the essenceof modern engineering innovations. However, despite theirunparalleled capability to generate substantial torques in compact spaces, they currently grapple with an impendingchallenge. In the burgeoning era of Industry 4.0, where digital technologies are swiftly advancing, the inherent worth of hydraulic systems is at peril, particularly when viewedthrough the prism of next-gen engineers and technologists.

Enter the Digital Twin paradigm: a transformative approach that stands poised to rejuvenate the realm of hydraulics. An intricate digital facsimile of a hydraulic system can be constructed using tools like Automation Studio™. This simulates physical operations in real-time and enhances the educational dimension by providing an interactive bridge between abstract hydraulic principles and their practical manifestations.

In an educational context, especially in universities where resource constraints might limit the availability of extensive hydraulic setups, the Digital Twin methodology presents aparadigm shift. Through this lens, students can dive deep into the mechanics of hydraulics without being physically present, thereby circumventing logistical and cost barriers. Such an approach can crucially retain the interest of a digitally-native student body, ensuring that the allure of hydraulics is maintained in a digital age.

Simultaneously, as the wave of Industry 4.0 swells, the convergence of the digital and physical spheres has become pivotal. The Digital Twin epitomizes this convergence. Withinthe academic domain, it has the potential to redefine traditional experimental laboratories, blending the advantages of virtual labs – scalability, safety, and universal accessibility – with the tangible, hands-on experience of remote labs. This synergistic fusion allows an experiential learning experience that is both immersive and secure.

As the boundaries of traditional education expand in response to evolving digital paradigms, the role of the Digital Twin becomes even more central. It symbolizes the future of versatile, dynamic, and inclusive academic methodologies. This transformative journey ensures that the essence of laboratory practices is preserved whilst making them more adaptable and resonant with the digital age.

979-8-3503-4929-0/24/$31.00 ©2024 IEEE

The manuscript is structured as follows: In Section 2, ”Related work”, a review of relevant literature is conducted, highlighting seminal contributions in the field. Section 3, ”Materials and methods”, comprehensively examines the hardware components, specifically the didactic hydraulic workstation, and introduces the computational framework, Automation Studio™. The ensuing ”Results and Discussion” section elucidates the intricacies of the digital twin for the hydraulic workstation, emphasizing distinct modeling characteristics and application nuances. The ”Conclusions” section encapsulates the principal outcomes and implications.

II. RELATED WORK

Digital Twin (DT) technology has been extensively reviewed, showcasing its transformative potential across diverse sectors, including engineering. [1] This technology, serving as a virtual mirror of physical entities, facilitates real-time data collection, validation, and simulation of its real-world counterpart. The review paper [2] highlights DT’s pivotal role in data-driven decision-making, system monitoring, and product lifecycle management while addressing the challenges inherent in its broad-scale adoption in engineering contexts.

Amidst the challenges posed by the COVID-19 pandemic to traditional experimental laboratories, especially with the shift to online learning modalities, another research advocated deploying Digital Twin technology as a viable recourse [3]. They emphasized that a DT-based laboratory proffers a digital or virtual counterpart of a physical system interconnected via the internet. This symbiotic linkage between the physical system and its virtual doppelganger retains the tangible interaction, concurrently amplifying the laboratory’s flexibility[4], [5].

In [6], challenges in fluid power education, such as outdated trainers and the increasing demand for online learning, have been addressed by developing a modern hydraulic trainer paired with its digital twin. This trainer integrates modernelectro-hydraulic components and sensors and is designed to offer a range of lab experiences, from basic actuator control to complex multi-actuator setups. Complementing this, a virtual trainer, developed using Unity 3D, mirrors the physical one, replicating the tangible lab experience, including common student errors and operational sounds. This virtual approach has been introduced as a promising solution for hands-on experiences in distance learning contexts.

In the extant literature, approaches towards the representation of digital twins in the context of hydraulic applications still need to be explored. While prominent hydraulic equipment manufacturers possess high-level modeling software to showcase applications from a marketing standpoint, these examples are conspicuously absent due to inherent modeling complexities in low-level modeling software, such as Automation Studio™ by Famic Technology. The approach delineated in our study is distinctively novel, offering a systemic perspective on developing a hydraulic workstation. Utilizing the 3D editor workshop within Automation Studio™, this workstation can be configured to illustrate various applications, thereby highlighting diverse operational scenarios. This methodology underscores the potential and versatility of integrating digital twins within hydraulic applications, bridging a notable gap in the current body of research.

Fig. 1: The didactic hydraulic workstation

III. MATERIALS AND METHODS

A. *Hardware – didactic hydraulic workstation*

The Tech-Con Hydraulic workstation, depicted in Figure 1, epitomizes a robust synergy between academic and industrial expertise. Through a dedicated sponsorship tailored for the Laboratory of Hydraulic and Pneumatic Drives, the university engaged in a fruitful collaboration with an eminent figure in the automation equipment arena. Even though the didacticworkstation of Eaton Hydraulics influenced its preliminary design, it underwent refined alterations to dovetail with the specificities of our laboratory environment. Designed withthe pedagogical environment in mind, it comfortably allows two students to work together, fostering a collaborative and immersive learning experience.

At its core, the workstation’s Fluid Direction Control is unparalleled. It boasts a drawer distributor encompassing four paths and three positions. Further enhancing its direction control capabilities, the workstation features a bi-positional solenoid valve that normally remains open and a non-return valve. Control over fluid flow rate is bestowed through a dedicated flow control valve.

Ensuring impeccable Pressure Regulation, the Tech-Con Hydraulic workstation is equipped with a sophisticated suite of valves. This includes a pressure-reducing valve for precise pressure adjustments. A sequence pressure and counterbalance valve supplement this, ensuring optimal pressure regulation across diverse hydraulic applications.

The workstation's Actuation Mechanism is its pride. Two double-acting hydraulic cylinders play a pivotal role. Cylinder2 is distinctively designed, housing an adjustable proximity sensor. In contrast, Cylinder 1 stands out with its unique springload, simulating a realistic piston advancement under load. Beyond these, the workstation accommodates a bidirectional hydraulic motor, emphasizing its diverse actuation capabilities.For real-time Pressure Monitoring, the workstation is adorned with two pressure indicators. While one remains affixed to the system's gallery, the other showcases remarkable flexibility, allowing users to mount it at any chosen point in the circuit. This ensures comprehensive pressure monitoring across the hydraulic setup.

The workstation's Electrical Control is streamlined through an ON/OFF switch, which could control the solenoid valve. The centrifugal pump, characterized by its potential to amplifypressure to 150 bar, is judiciously regulated. Given the academic context of its operation, a pressure valve is employedto cap the working pressure at a safer threshold of 50 bar.This meticulous approach prioritizes safety and aligns withthe typical requirements of a university laboratory setup. The power unit's electric motor, synchronized with a tri-phasic frequency converter, offers the added nuance of adjusting the hydraulic pump's rotational speed.

The components chosen reflect real-world engineering challenges and resonate with those predominantly encountered in modern hydraulic landscapes. Connected through flexible hoses and foolproof couplers, they pave the way for constructing an extensive range of hydraulic circuit configurations, mirroring practical applications.

B. *Software Component: Automation Studio™*

In the development presented in this paper, the Automation Studio 6.3 Educational Edition was instrumental in designing, simulating, and analyzing the hydraulic systems under study. Boasting an expansive Hydraulic Library, this software ensurescompliance with ISO 1219-1 and 1219-2 standards, offering users an exhaustive collection of fluid power components, fromthe basic such as pumps and motors to intricate measurement instruments and valves. Designed to cater to both the novice and the expert, the software comes equipped with built-in configurators, prominently featuring the Fluid Power Valve Spool Designer, allowing for the creation of virtually any valvespool position. Upon schematic completion, the true prowess of Automation Studio™ reveals itself, enabling real-time simulation. Animated schematics allow users to measure, in real-time, various parameters, enriching the analysis. This dynamic platform is enhanced by innovative troubleshooting tools, allowing users to simulate potential hydraulic failures and their subsequent effects. Automation Studio™ collaborateswith leading fluid power component manufacturers to further bolster its capability. This partnership ensures that users can integrate manufacturer-preconfigured components, bolstering the authenticity of their simulations. The software bridgesto Electrotechnical and PLC domains, not limiting itself to hydraulic systems, fostering a comprehensive environment for system analysis [7].

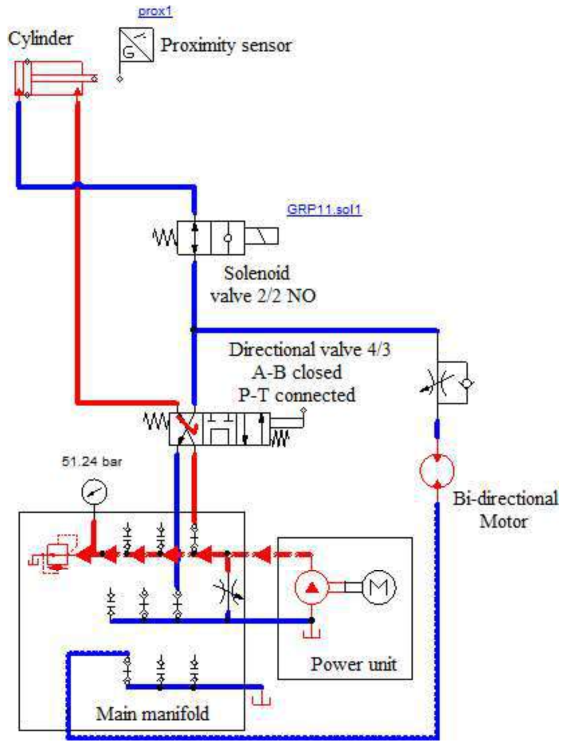


Fig. 2: Electro-Hydraulic Control Circuit Implementation

The hydraulic circuit design was executed in Automation Studio™, leveraging the extensive components available in the EATON library. EATON, a renowned hydraulic component manufacturer, provides a wide array of preconfigured elements within the software. These components cater to standard design necessities; however, specific and unique requirements were identified during the design phase. Automation Studio™, with its inherent flexibility, facilitates the creation of non-standard hydraulic equipment by adapting predefined components. This adaptability is essential for modifying components to meet specific design prerequisites.

Utilizing components from the EATON library streamlined the design and enhanced the simulation process. With the intrinsic simulation parameters of these components, thedesign's efficiency and accuracy were markedly improved.Consequently, the final design closely mirrored the anticipatedreal-world performance and behavior of the hydraulic system.In this research, the embodiment of the hydraulic system under investigation is vividly presented in Figure 2. Thisschematic offers a snapshot of an electro-hydraulic controlsystem. The schematic precisely captures the position control of the cylinder, facilitated by the proximity switch. This control element ensures that the cylinder halts movement onceit reaches a specific point. Simultaneously, the flow control valve depicted in the system plays a pivotal role in regulatingthe motor's speed, with the orifice size being the determining factor for this speed modulation.

This study aims to transition the hydraulic schematic in Figure 2 into a dynamic digital twin, realized in a three-dimensional model using the facilities offered byAutomation Studio's 3D workshop. This digital twin isnot merely envisioned as a representation of the hydraulicstand. Instead, it serves as a bridge, seamlessly connectingthe theoretical understanding of hydraulics with its tangible real-world application. Through this approach, a detailed and immersive simulation is facilitated. The shift from a static, two-dimensional representation to a vividly animated digital counterpart encapsulates this research's central ethos and aspiration. In adopting this approach, the constraints imposed by the physical hydraulic stand, which can accommodate a maximum of two students simultaneously, are effectively circumvented. Integrating this into a virtual application allows students to engage with the digital twin on their computer independently. This democratizes access and enables students to prepare and gain preliminary experience ina virtual environment. Such preparation is paramount as, while digital tools offer invaluable insights and practice, the hands-on, tactile experience at the actual hydraulic stand truly consolidates a student's understanding and skills in the realm of hydraulics.

In the field of hydraulic system simulations, Automation Studio's 3D editor elevates the concept of a virtual system to an advanced level, enhancing it by providing more seamless access to analysis based on the construction and simulation of the virtual environment. This editor facilitates the effortless creation of 3D parts and enables the import of 3D file formats, such as .STEP and .stl from other applications.

Capitalizing on this sophisticated feature of Automation Studio™, the digital twin's construction embarked on a sequential process, and the results are presented in Figure 3: The AL frame of the stand was crafted using fundamental geometric shapes, augmented with depth and realism through specific 3D effects found within the Visibility submenu of the Layout menu - Figure 3(a).

Subsequently, the Reservoir-Pump-Motor assembly was constructed. This was initiated with the tank's development, followed by articulation of the three-phase motor, positioned to the right, and the centrifugal pump on the left, as delineated in section Figure 3(b).

The following phase included two bidirectional hydraulic cylinders, with one endowed with a spring mechanism to emulate the effects of an external load - Figure 3(c).

Diverse valves were then integrated, encapsulating a normally open bi-positional solenoid, a directional check valve, a flow control, a pressure reduction valve, a sequential pressure valve, and a counterbalance valve - Figure 3(d). A spool distributor featuring a four-way, three-position design was also embedded, supplemented with an operative handle sourced from the HMI and Control Panels library. Concurrently, the main manifold for the hydraulic system's gallery was established, as discerned in - Figure 3(e).

Culminating the design, a bidirectional hydraulic motor wasplaced predominantly in the top - Figure 3(f), flanked by the ON/OFF switch, a pressure gauge, a proximity sensor, and theprimary command panel.

After the design phase for individual equipment pieces, emphasis was laid on the assembly of these components. Sucha meticulous step was indispensable to ensure the digital twin'salignment with the actual equipment housed in the laboratory. Central to this endeavor was creating connectors between the various circuit elements. These connectors manifest as hoses, each equipped with quick-coupling valves in alignment with the hydraulic blueprint presented in Figure 2. A pivotal phase involved connecting the integrated motor-pump-reservoirassembly to the system manifold, then to the pressure-reducingvalve, the spool distributor, the cylinder ports, the directional valve, and the bi-directional motor.

While the representation of the stand encompasses all real-world components, only a subset of these componentsis integral to the circuit under investigation in this paper.Components not implicated in the active circuitry have been rendered in shades of grey to enhance clarity and direct focus. This visual distinction serves a dual purpose: it accentuates theoperational components, which retain their foundational color while offering a holistic overview of the stand's entirety. To achieve a cohesive and realistic representation of the hydraulic system in action, 3D animations were judiciously deployed. These animations were vital in simulating both the extension and retraction of the cylinders, as well as the activation ofthe proximity sensor. Moreover, animations were integrated within the connectors, visually illustrating the flow directionof the hydraulic fluid. Components transitioning from dormantto operational were also subject to animation to provide a more immersive understanding.

The synchronization between these animations and the actual circuit was accomplished using internal variables embedded within each component's menu. Complementing this, two electrical control circuits were designed for the control panel and the spool distributor's flow direction toggle, respectively, along with a control circuit for the solenoid valve. Notably significant is the connection between the HMI panel and the motor within the circuit, governed bythe start/stop circuit, responsible for initiating or halting the stand's operation – Figure 4.

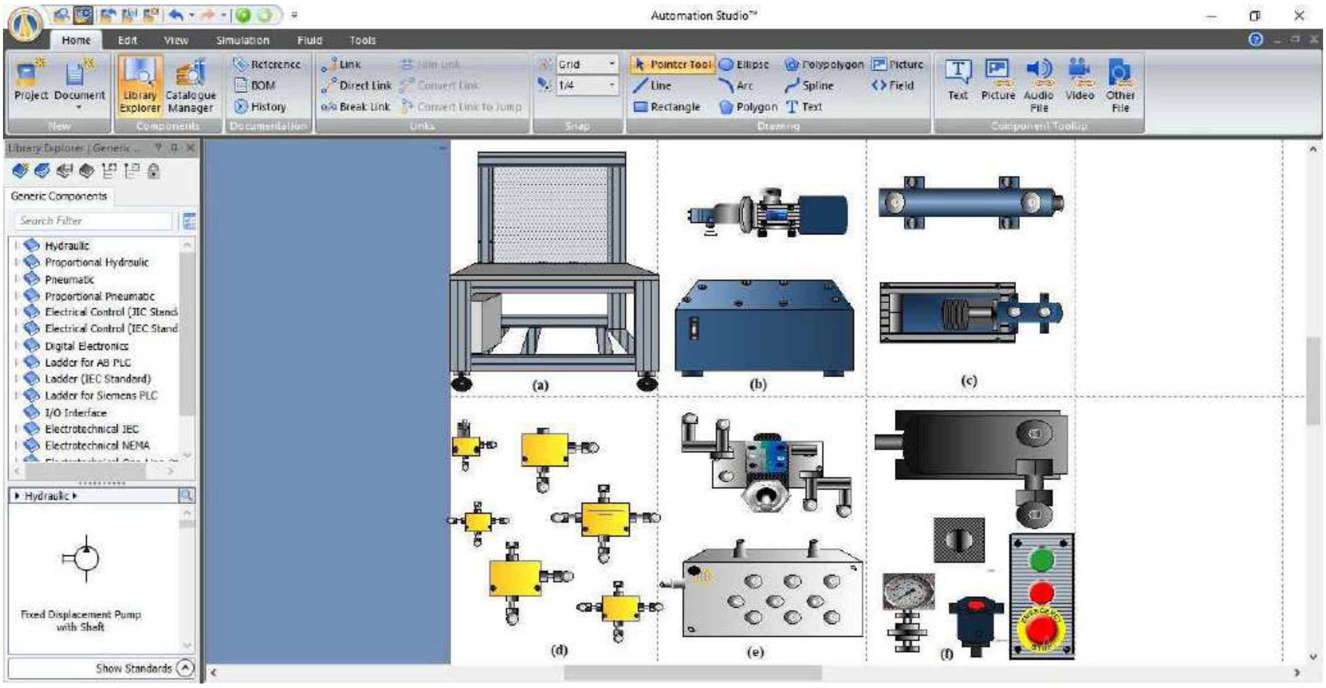


Fig. 3: Process of digital twin construction for the equipment from the hydraulic workstation in Automation Studio™

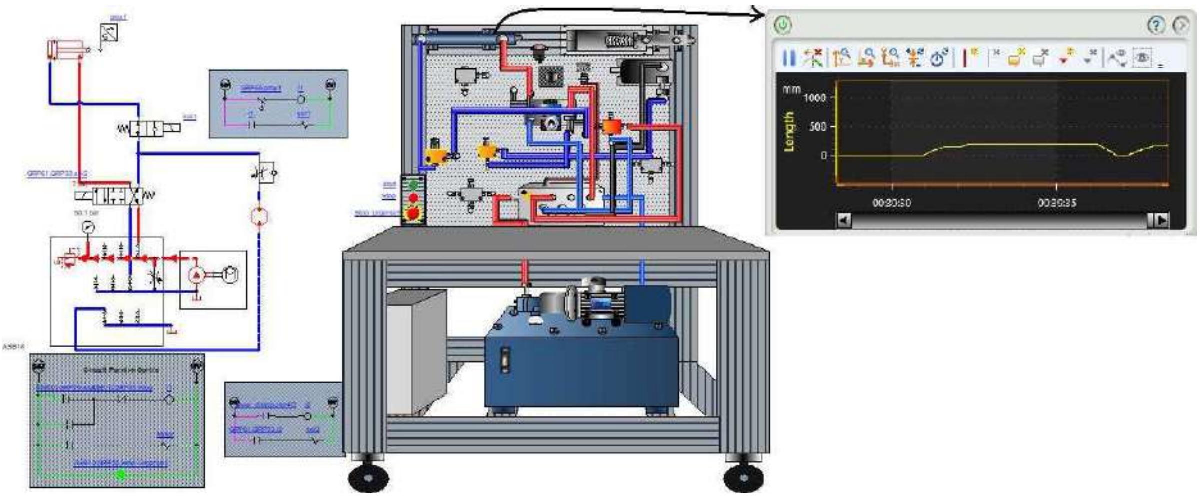


Fig. 4: Hydraulic workstation's digital twin in operation

IV. Results and discussion

The 'Simulation' menu facilitates the designed circuits' simulation. This menu grants access to various options, most notably, the 'Step by Step' progression and the 'Slow Motion' feature, optimized for enhanced visualization of the application in motion. The 'Measuring' section monitors and analyzes individual components' behavior during simulation closely. Here, one can graphically plot and observe the changes transpiring during the simulation process. Another vital subsection of this menu is the 'Troubleshooting', which offers a comprehensive set of tools such as an Oscilloscope, Multimeter, Hydraulic Tester, Pressure Gauge, Thermometer, and editing tools to ensure precision in measurements.

The simulation focused on a circuit whose primary actuationmechanism was a single bi-directional hydraulic cylinder, and its secondary component was a bi-directional hydraulic motor – depicted in Figure 4. The simulation sequence was inaugurated by activating the three valves within the system manifold, mirroring the real-world action of rotating the main valve on the system gallery casing. The subsequent step involved assessing the functionality of the command panel. This panel was designed employing elements from the HMI library. These were connected to the hydraulic circuit's motor via an electrical schematic responsible for receiving anddispatching electrical signals through a solenoid and associatedcoils. The functionality of the 'Start' button was tested, which,

upon activation, instantaneously actuates the motor. The latter subsequently drives the pump, instigating the flow of hydraulicfluid in the circuit. Post this, the 'Stop' button was assessed, leading to the immediate cessation of the tri-phase motor.

The 'Emergency Stop' is a paramount feature in any system due to its role in abrupt equipment halting during technical anomalies. Effective reactivation can only be accomplished after the identification and rectification of the underlying defect, coupled with the release of this button. Importantly, irrespective of the 'Start' button being pressed, the electric circuit design inhibits the motor's activation as long as the emergency button remains engaged.

To glean a spectrum of measurements, spanning from pressure, positioning to acceleration and velocity, the specialised tools offered by Automation Studio can be employed. Figure 4 exemplifies this utility. Specifically, this figure showcases the plotting of the piston's position with its displacement time, subsequently facilitating the derivation of its evolution graph. This graph delineates the trajectory of the piston over time, highlighting its dynamic behavior during the simulation.

Utilizing the advanced capabilities of Automation Studio augments the precision of observational data, forging a robust correlation between tangible operations and their digital counterparts. The data extracted from these tools is quintessential, ensuring that real-time actions in the physical realm seamlessly mirror the digital twin, eliminating operational discrepancies.

One salient instance is the hydraulic cylinder's behavior, where the hydraulic oil's flow rate is paramount. Discrepanciesbetween the digital model and the real-world apparatus hint at potential deviations in the parameters set or the operational timeline. Addressing such discrepancies is pivotal to enhancing the fidelity of the digital twin's portrayal.

A suitable remedy to reconcile these deviations is the recalibration of the hydraulic oil flow rate, considering the integral relationship between this metric and the cylinder's actuation speed. A closer emulation of the real-world system'skinetics becomes attainable by judiciously adjusting this parameter within the digital twin's parameters. This iterative process ensures enhanced synchronization, harmonizing the digital representation with its tangible counterpart.

Furthermore, a standout feature intrinsic to the real-world hydraulic system is the capability to fine-tune the motor's rotational speed using a static frequency inverter. This adaptability facilitates tailored control of the motor's dynamics, aligning them with specific operational needs. This real-world sophistication is mirrored with precision within the digital twin. Embedded in its model is a feature that allows forthe specification of the rotational speed (in rpm) of the drive motor for the centrifugal pump. Such integration emphasizes the digital twin's comprehensive nature, capturing even the most intricate operational nuances.

V. Conclusions

The significance of a digital twin in educational settings becomes increasingly apparent. The accuracy with which the digital twin replicates the real-world hydraulic workstation determines its precision and educational allure. By closely mirroring real-world functionalities, the digital twin offers students a robust, immersive learning platform that connects theoretical knowledge with practical applications.

The modular nature of this co-simulation environment showcases its adaptability and scalability. This allows for the easy integration and activation of additional hydraulic and electrical components, enhancing its comprehensive educational potential. Such flexibility ensures the digital twin can be effortlessly expanded as the hydraulic domain evolves and grows. It is a future-proof and invaluable tool for in-depth understanding and exploration of complex hydraulic systems.

Acknowledgment

The authors thank Tech-Con Industry for their continuous support in developing the Hydraulic and Pneumatic Control Systems Lab from TUIASI.

References

[1] Y. Zhu, H. Su, S. Tang, S. Zhang, T. Zhou, and J. Wang, "A novel fault diagnosis method based on swt and vgg-lstm model for hydraulic axial piston pump," *Journal of Marine Science and Engineering*, vol. 11, no. 3,p. 594, 2023.

[2] D. M. Bot´ın-Sanabria, A.-S. Mihaita, R. E. Peimbert-Garc´ıa, M. A. Ram´ırez-Moreno, R. A. Ram´ırez-Mendoza, and J. d. J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review," *Remote Sensing*, vol. 14, no. 6, p. 1335, 2022.

[3] D. Pang, S. Cui, and G. Yang, "Remote laboratory as an educational tool in robotics experimental course.," *International Journal of EmergingTechnologies in Learning*, vol. 17, no. 21, 2022.

[4] H. Tjahyadi, K. Prasetya, and I. M. Murwantara, "Digital twin based laboratory for control engineering education," *International Journal of Information and Education Technology*, vol. 13, no. 4, 2023.

[5] C.-C. Dosoftei and A.-E. Cojocaru, "Implementation of a virtual control lab to support teaching in engineering control," in *2020 International Conference and Exposition on Electrical And Power Engineering (EPE)*, pp. 699–703, IEEE, 2020.

[6] H. Assaf and A. Vacca, "Hydraulic trainer for hands-on and virtual labs for fluid power curriculum," in *Scandinavian International Conferenceon Fluid Power*, pp. 8–25, 2021.

[7] Famic Technologies Inc., Automation Studio™ Documentation. https://www.famictech.com/portals/0/PDF/brochure/automation-studio-educational/automation-studio-E6-brochure-english-high.pdf, 2023. (accessed: 17.03.2023).

Simplified Mecanum Wheel Modelling using a Reduced Omni Wheel Model for Dynamic Simulation of an Omnidirectional Mobile Robot

Cătălin Dosoftei[1], Vasile Horga[2], Ioan Doroftei[3], Tudor Popovici[1]
[1]Dept. of Automatic Control and Applied Informatics,
[2]Dept. of Energy Utilisation, Electrical Drives and Industrial Automation,
[3]Dept. of Mechanical Engineering, Mechatronics and Robotics
"Gheorghe Asachi" Technical University of Iasi, Romania
{constantin-catalin.dosoftei, vasile.horga, ioan.doroftei, alexandru-tudor.popovici}@academic.tuiasi.ro

Ştefan Custura[4]
[4]Dept. of Mechanical Engineering Design
SELF TRUST Company from Iasi, Romania
stefan.custura@selftrust.ro

***Abstract*—Industrial and technical applications of omnidirectional mobile robots, with Mecanum wheels, are continuously gaining in importance because their excellent mobility, especially in areas with static or dynamic obstacles. The perspective of Industry 4.0 imposes an effective and efficient approach for production in general and then, specific to this application, to implement the control algorithm able to deliver to robot the deliberatively and reactively behaviour. An important tool to achieve these goals is virtualization of the prototype and his simulation in a 3D virtual environment.**

This paper presents a simplified method of modelling and kinematics simulation in Simscape Multibody™, taking an omnidirectional mobile robot with advantages related to simulation time. The necessary steps to develop inverse kinematics in robots and global reference frame, for specific omnidirectional mobile robot, are depicted in the paper using bottom-up approach with matrix transformation. The 3D modelling process starts from the CAD design, which is then transferred into a Simscape Multibody™ model in order to study the kinematic and dynamic characteristics. Four typical motion models including moving forward, moving laterally, moving in the diagonal direction and rotation have been simulated in Simscape Multibody™.

Keywords— omnidirectional mobile robot; kinematics; virtual prototype; Mecanum wheels; simulation; Simscape-Multibody.

I. Introduction

In the last decade, autonomous systems have become quite a popular topic for collaboration between several research academic groups and industry. Autonomous systems are complex structures being a multidisciplinary approach of various engineering and science disciplines, from mechanical, electrical and electronics engineering to computer, cognitive and social sciences. An important direction of this domain is represented by autonomous mobile robots (AMRs), defined in [1] as a system which operates with a various degree of autonomy in order to perform a goal-oriented. AMRs are highly versatile systems with the capability to take decisions in order to solve the complex tasks through two types of approaches: deliberately and reactively. Acting deliberately means the AMR requires relatively complete knowledge of the operating space and uses it to plan its subsequent actions. On the other hand, reactive actuation is based on techniques for tightly coupling information from the environment and the robot's drive system so that for a displacement in a dynamic or unstructured operating environment it can have a real-time response [2].

Based on diversity of operational environments that they have to operate, the AMRs can be classified into unmanned ground vehicles (UGVs), unmanned water vehicles (UWVs), autonomous underwater vehicles (AUVs), and unmanned aerial vehicles (UAVs). The research presented in this paper is related to a subclass of UGVs, which is using Mecanum wheels. The main strength of the use of this omnidirectional locomotion in UGV applications is due to the ability of these wheels to move instantaneously in any direction, gaining advantages in terms of their manoeuvrability and productivity in narrow spaces and/or crowded environments [3]. This is possible because translational and rotational motions can be decoupled for vehicle displacement. Typical applications for omnidirectional mobile robots (OMRs) are from different fields, as they are presented in [4], such as military applications, plant work-shops, logistics spaces, offices and care facilities like hospitals.

The paper is organized as follows. Section II explains the kinematic model of an OMR with 4 Mecanum wheels. Section III presents the 3D CAD model of the robot which is available at the Department of Automatic Control and Applied Informatics – TUIASI. In section IV the translation of obtained OMR's model in Simscape Multibody™ is presented. The main contribution of this paper, presented in section V, is a proposal of a simplified equivalent model for the 4 Mecanum wheeled platform, based on the behaviour of the 4 Omni wheeled platform. The proposed simplified model is less time consuming and it can guarantee acceptable kinematic and dynamic accuracy. The model behaviour and comparative analyses are presented in section VI. Section VII presents the conclusions.

II. Kinematic Model

In this section, the kinematics of an omnidirectional mobile platform with 4 Mecanum wheels is depicted. Kinematics study offers the possibility to analyse the geometry of motion

978-1-7281-8126-4/20/$31.00 ©2020 European Union

for OMR. On the other hand, in order to control its manoeuvrability one has to use associated inverse kinematics.

A. *Mecanum Wheels*

Mecanum wheel is based on the principle of a central hub with several rollers placed at an angle around the periphery of the hub. The angle between rollers axes and central hub axis could have any value, but in the case of conventional Mecanum wheel it is 45°. The OMR kinematic model is presented in the technical literature [3],[5]-[7] like a typical bottom-up approach: the first step starts with constraints on the motion of individual wheel, and then, by combining the motion of used wheels the behaviour of the designed robot as a whole can be obtained [1]. In order to simplify the model, there are defined some ideal conditions of operating Mecanum wheel in a 2D plane [8] and construction conditions of wheel/chassis. Two common methods to develop the kinematic constraints of a Mecanum wheel are used: one based on vector method and the other characterised by using the matrix transformation. Our approach is the second one, and for this purpose we will define the following Cartesian coordinates systems and configuration parameters:

- $O_{wi}X_{wi}Y_{wi}Z_{wi}$ - the system attached to wheel w_i;
- $o_ix_iy_iz_i$ - the system attached to the roller in contact to the ground;
- OXYZ - the local systems attached to robot base chassis;
- $O^GX^GY^GZ^G$ - the global world reference frame;
- R – wheel radius;
- (l_i, α_i) – defines the position of wheel w_i towards the kinematic centre of chassis O, $|l_i| =$ dist.$(O_{wi}\ O)$;
- β_i, γ_i - the angle between direction of vector l_i and X_{wi} axis of the wheel, the tilt angle between axes x_i and X_{wi} projection respectively;
- θ - rotational angle of the robot around Z_R axis;
- ω_i, Ω - angular velocity of the wheel w_i, angular velocity of the robot, respectively ($\Omega = \dot\theta$)
- v_x, v_y - the instantaneous translation velocities of the robot in its coordinated axes;
- v_x^G, v_y^G, Ω^G - the instantaneous velocities of the robot in global reference frame;
- v_{gi} - the velocity of roller in contact with the ground from wheel w_i;
- l_x, l_y - half of the distance between the front wheels contact points, half of the distance between the front and the rear wheels contact points;

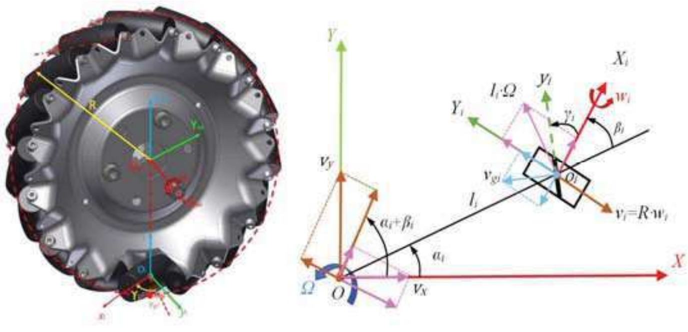


Fig. 1. Coordinate systems attached to Mecanum wheel w_i and kinematic constraint diagram of its using a matrix transformation method.

All conventions in Fig.1 respect right-handed Cartesians coordinate system. In representation of the wheel in OXY plane everything should be judged in terms of wheel tracks left on ground.

The matrix transformation method is the most used method for kinematics constraints analysing of a wheeled mobile system. The calculation is based on the study of rolling and sliding constraints of the Mecanum wheel which requires some subtlety: in order to obtain movement, the contact point between wheel and ground is a pure rolling condition without slipping [9].

B. *Inverse Kinematics*

Performing translation and rotation of velocity of the coordinate frame of the wheel w_i to instantaneous velocities of the robot, the following relations result [6], [10]:

$$v_x \sin(\alpha_i+\beta_i) - v_y \cos(\alpha_i+\beta_i) - l_i\Omega\cos\beta_i = R\omega_i - v_{gi}\cos\gamma_i \quad (1)$$

$$v_x \cos(\alpha_i+\beta_i) + v_y \sin(\alpha_i+\beta_i) + l_i\Omega\sin\beta_i = \qquad - v_{gi}\cos\gamma_i \quad (2)$$

Being an uncontrollable variable of the passive roller, the velocity v_{gi} must be eliminated from above equations. In this way, from eqs (1)-(2) it results

$$v_x \cos(\alpha_i+\beta_i+\gamma_i) + v_y \sin(\alpha_i+\beta_i+\gamma_i) + l_i\Omega\sin(\beta_i+\gamma_i) = -R\omega_i\sin\gamma_i \quad (3)$$

By introducing the vector of robot velocities $\dot\zeta = [v_x\ v_y\ \Omega]^T$ in eq. (3), the wheel speed ω_i can be expressed as

$$\omega_i = -\frac{1}{R\sin\gamma_i}[\cos(\alpha_i+\beta_i+\gamma_i)\ \ \sin(\alpha_i+\beta_i+\gamma_i)\ \ l_i\sin(\beta_i+\gamma_i)]\dot\zeta \quad (4)$$

Because it is taken into consideration an OMR with four Mecanum wheels in optimal configuration, the wheels are in a symmetrical arrangement with axes of rollers in "square" layout – Fig. 2, and rollers angle values for each wheel are presented in Table I.

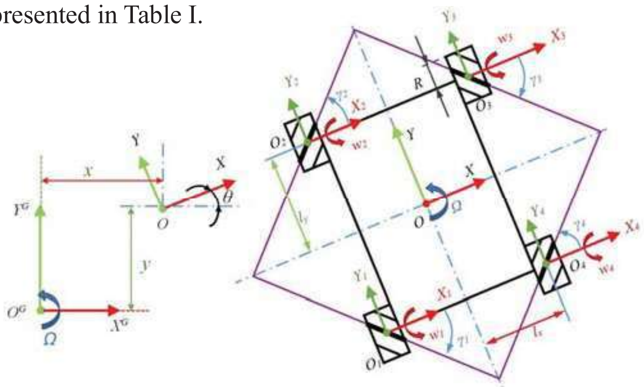


Fig. 2. Coordinates system assignments for the four-Mecanum-wheel robot with arrangement of rollers in "square" layout.

Table I Roller angle values for wheel w_i

Index i	1	2	3	4
γ_i	-45°	45°	-45°	45°

Specific for this kind of wheels distribution is that the sum of angle α_i and angle β_i is null, i.e.

$$\alpha_i = -\beta_i \quad (5)$$

and, for this case, eq. (4) can be rewritten as:

$$\omega_i = -\frac{1}{R}\left[\frac{\cos\gamma_i}{\sin\gamma_i}\ \ 1\ \ l_i\frac{\sin(-\alpha_i+\gamma_i)}{\sin\gamma_i}\right][v_x\ v_y\ \Omega]^T \quad (6)$$

From constructive dimensions of chassis one obtains $l_x = l_i\cos\alpha_i$, $l_y = l_i\sin\alpha_i$ and hence eq. (6) becomes:

$$\omega_i = -\frac{1}{R}\left[\operatorname{ctg}\gamma_i \;\; 1 \;\; l_x - l_y \operatorname{ctg}\gamma_i\right][v_x \;\; v_y \;\; \Omega]^T \qquad (7)$$

Taking into consideration the value of rollers angle for each wheel and the sign of chassis dimensions, by using eq.(7) four times one obtains the inverse kinematics matrix of the OMR, which relates the angular velocities of the Mecanum wheels to the velocities coordinates of robot as follow:

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = -\frac{1}{R}\begin{bmatrix}-1 & 1 & -(l_x+l_y)\\ 1 & 1 & -(l_x+l_y)\\ -1 & 1 & (l_x+l_y)\\ 1 & 1 & (l_x+l_y)\end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ \Omega\end{bmatrix} \qquad (8)$$

The angular velocities of the Mecanum wheels can be related to the velocities in global coordinates by using the *orthogonal rotation matrix, denoted by*:

$$R_{rot}(\theta) = \begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \qquad (9)$$

Considering that the vector of robot velocities can be expressed as

$$\dot{\zeta} = R_{rot}(\theta)\,\dot{\zeta}^G \qquad (10)$$

then the inverse kinematic equation (8) in global reference frame $O^G X^G Y^G Z^G$ becomes:

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = -\frac{1}{R}\begin{bmatrix}-1 & 1 & -(l_x+l_y)\\ 1 & 1 & -(l_x+l_y)\\ -1 & 1 & (l_x+l_y)\\ 1 & 1 & (l_x+l_y)\end{bmatrix}\begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}v_x^G\\ v_y^G\\ \Omega^G\end{bmatrix} \qquad (11)$$

Inverse kinematics is essentially in creating a kinematic chain when the mobile robot is modelled with a chassis compound of rigid segments connected with joints. More than this, with inverse kinematics problem, the joint angles for a desired position of the platform may be computed, which is useful in following a robot trajectory when the controller calculates the translational and angular velocities.

The inverse Jacobian matrix of the OMR is

$$J = -\frac{1}{R}\begin{bmatrix}-1 & 1 & -(l_x+l_y)\\ 1 & 1 & -(l_x+l_y)\\ -1 & 1 & (l_x+l_y)\\ 1 & 1 & (l_x+l_y)\end{bmatrix} \qquad (12)$$

Equation (12) shows that for the OMR with more than 3 Mecanum wheels, the kinematics is over-determined i.e. only three rows of the Jacobian matrix are linear independent, the fourth one being a linear combination of the first three.

C. Forward Kinematics

The approach which starts with wheels velocities in order to determinate the platform velocities is known as forward kinematics. In order to obtain the forward kinematic equations starting from Jacobian matrix, which is a 4x3 matrix, it is necessary to introduce a pseudo inverse matrix J^+ such that $J^+ \cdot J = I_3$. It is determined with the formula [7]:

$$J^+ = (J^T \cdot J)^{-1} \cdot J^T \qquad (13)$$

In this way one can obtain the equation for forward kinematics as following:

$$\begin{bmatrix}v_x\\ v_y\\ \Omega\end{bmatrix} = -\frac{R}{4}\begin{bmatrix}-1 & 1 & -1 & 1\\ 1 & 1 & 1 & 1\\ -\frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} & \frac{1}{l_x+l_y}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} \qquad (14)$$

Equation (14) shows the three degrees of freedom (3DoF) of the platform: two translational velocities on X and Y axes and one angular speed around Z axis. The resultant translational velocity is defined by:

$$v = \sqrt{v_x^2 + v_y^2} \qquad (15)$$

and orientation is calculated as:

$$\theta = \tan^{-1}\frac{v_y}{v_x} \qquad (16)$$

From forward kinematics given by (14) the OMR can be driven in every direction according to the direction and angular speed of the wheels:

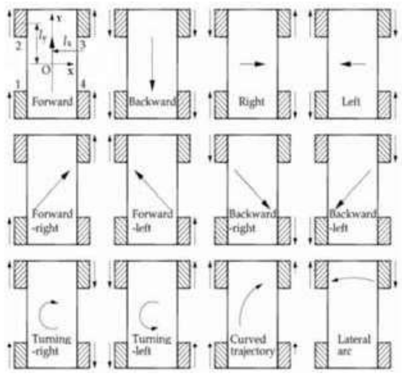


Fig. 3. Locomotion directions according with angular velocity of the wheels [4].

The figured arrows attached to the wheels take into account the (-) sign of Jacobian matrix and that resulted from right-handed Cartesians coordinate system for angular velocities.

III. Designing the Model in 3D CAD

In last decade when the fourth industrial revolution imposed proper paradigms, the virtual prototyping started to have the biggest impact in an integrated concept of preparing physical prototypes. Advantages brought by modelling, testing, measurements in the virtual environment are transposed in real environment by production cost savings, reduced testing and optimization time, tests without destroying prototypes, increase of quality (operational performances) of the products. In [10] virtual prototyping is defined as a computer-aided engineering-based discipline. The starting point for such system is high-performance 3D CAD software.

In order to study the kinematic and dynamic characteristics for the ORM, the 3D CAD model was designed using Autodesk-Inventor software and virtual prototype simulations were carried out in Simscape Multibody™ to simulate and optimize its behaviour. Even though it is possible to develop models directly in Simscape Multibody™, specialized tools offered by high-performance CAD software give the possibility to create more sophisticated 3D geometrical model in order to obtain all the mass and inertia properties.

The real OMR and the model of OMR designed in Autodesk Inventor with all details are presented in Fig. 4.

Fig. 4. The real OMR and its model in Autodesk-Inventor.

Before being exported to Simscape Multibody™, the model should be simplified, to diminish the amount of data processing and to increase the efficiency of simulation time. For this purpose it is necessary to retain only the main components with big impact in simulation: four Mecanum wheels, four motor-gearbox assemblies, eight shock absorbers, eight parallelogram mechanisms and the main body of platform, which freeze all other components with less impact in simulation (battery, control/communications equipment and other mechanical components). Reducing texture of the materials can also affect the performance in simulation.

The simplified model, Fig. 5, contains all details with impact in model dynamics.

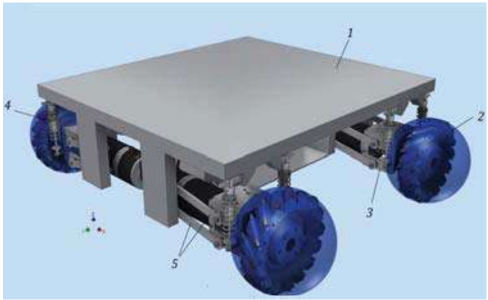


Fig. 5. Reduced 3D model of the OMR in Autodesk-Inventor with 1-rigid body, 2-Mecanum wheel and circumscribed sphere, 3- motor-planetary gearbox assembly, 4-shock absorber, 5-parallelogram mechanisms.

Table II presents some characteristics and dimensions of real OMR embedded in simplified model from Inventor.

Table II Dimensions of real OMR

Parameter (symbol)	Value	Unit
Distance l_x	0.294	[m]
Distance l_y	0.200	[m]
OMR weight	14.5	[kg]
Wheel radius (R)	0.076	[m]
#of rollers/wheel	16	

In order to prevent errors when importing the model it is necessary to make sure that in CAD assembly model the constraint combination between bodies are correctly defined, otherwise unsupported constraint combinations between bodies will be replaced in Simscape Multibody™ by a rigid connection, in one of the forms: direct frame connection line, Rigid Transform block or Weld Joint block. Reference [11] presents all translation processes of a CAD model into a Simscape Multibody™ - Simulink model. This process is done in 2 steps. First, the CAD export procedure creates an Extensible Markup Language (*xml*) multibody description file, and then a set of geometry files with extension *STEP* which provides the 3D surface shapes of the CAD parts.

IV. The OMR Dynamic Simscape Multibody Model

The procedure to obtain a dynamic model for a complex mechatronic system, as it is the OMR, is a confortable way which makes Simscape Multibody™ a powerful and friendly tool for manipulation of a 3D virtual prototyping. The simulation of the model can be done either when the system is going to be built, or if it already exists and it is necessary to develop control system – the case of preparing the prototype in the current research project.

The results derived from importing CAD model are a Simscape Multibody™-Simulink model (extension *slx*) of OMR and an *m* data file. The data from *m* file sets the block parameter values of the imported Simscape-Multibody™ model. The translated model keeps the structural hierarchy from the CAD assembly.

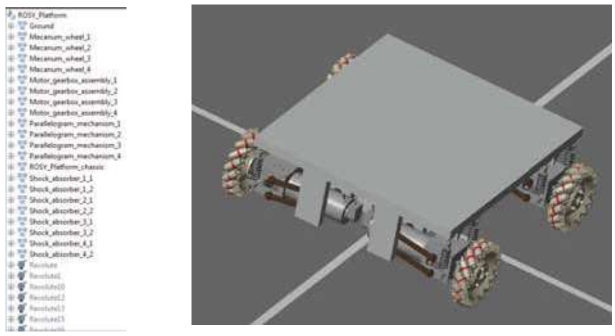


Fig. 6. Structure of the OMR Simscape-Multibody™ model with main components.

The Simscape Multibody™ model is represented as combinations of joints and constraints blocks. After processing and rearranging the blocks, the resulting model is presented in Fig.7, where the main blocks are highlighted in the form of subsystems in which the component elements have been grouped.

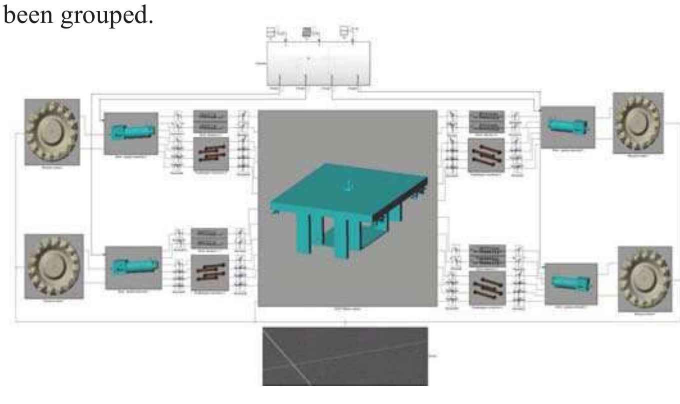


Fig. 7. The OMR Simscape Multibody™ model.

In order to obtain a realistic model for the Mecanum wheels that could also be simulated using Simscape Multibody™, the 16 rollers were approximated using 16 spheres having the same radii as the rollers. Each sphere is attached to the circumference of the wheel using a separate non-actuated revolute joint to provide the needed rotational DOFs, similar to the work presented in [12]. Furthermore, the interaction of each simplified roller with the ground surface is modelled with instances of the standard sphere to plane contact model from Contact Forces Library (CFL) extension of Simscape Multibody™. The modelled contact spheres are partially visible in Fig. 6 represented as red rings on the centre of each roller.

Obviously, the approximation of the original rollers with spheres causes a non-uniform transition between two consecutive rollers that creates vibrations in the OMR body. While this may seem like a discrepancy compared to the ideal Mecanum wheel, it can also reflect the behaviour of some real wheels which due to their particular manufacture may produce a similar effect although attenuated. Such a transition vibration was observed also for the wheels of the experimental OMR used in the current research.

An important downside of the above model is the slow simulation speed, issue that motivated further research into a simplified equivalent dynamic model.

V. Mecanum Wheel Modelling using Reduced Omni Wheel Model

In order to accelerate the simulation, the Mecanum wheel can be considered as a source of force in the direction of the rotation axis of the roller in contact with the ground surface. Starting from this, it is of interest to develop a simulation model that achieves similar dynamic characteristics to a Mecanum wheel while unburdening the computational complexity.

The simplest apparent solution is to model the Mecanum wheel as a sphere of the same radius, sphere which would be torque actuated on one horizontal axis, optionally immobilized on the vertical rotation axis (since rotation of the sphere around the vertical axis would not produce translation motion, nor the original rollers have this degree of freedom) and also allowed to rotate freely around the last remaining orthogonal horizontal axis, as the rollers do.

Such a model can be implemented in Simulink using two components from Simscape Multibody™: a sphere to plan contact model from CFL and a spherical joint which is closed loop torque actuated on two of the orthogonal rotational axis by sensing the orthogonal angular speeds as illustrated in Fig. 8.

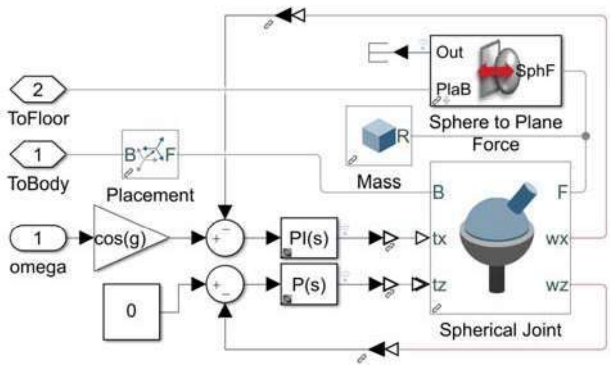


Fig. 8. Simplified model for Mecanum wheel implemented in Simscape Multibody™ by equivalence with a reduced Omni wheel.

One of the torques can be controlled by a PI controller to compensate the angular velocity error between the desired wheel angular velocity and the actual angular velocity, while for immobilizing the rotation around the vertical axis (which can be helpful while inspecting the dynamics of the model) a simple P controller with constant reference speed set to zero would suffice.

The proposed model represents a reduced Omni wheel (an omnidirectional wheel for which $|\gamma_i| = 90°$), Fig.9.

		Mecanum	Omni
Kinematics	forward – v_f	$\omega \cdot R$	$\omega \cdot R\sqrt{2}$
	sideways - v_s	$\omega \cdot R$	$\omega \cdot R\sqrt{2}$
	diagonal - v_d	$\omega \cdot R / \sqrt{2}$	$\omega \cdot R$
Force	forward – F_f	$4\tau / R$	$4\tau /(R\sqrt{2})$
	sideways - F_s	$4\tau / R$	$4\tau /(R\sqrt{2})$
	diagonal - F_d	$2\tau\sqrt{2} / R$	$2\tau / R$

Fig. 9. Comparison based on velocity and pushing force.

The two columns in Fig. 9 summarize the ground velocities and traction forces for vehicles with four wheels of type Mecanum and Omni respectively, considering they have the same diameter. The first three rows are vehicle velocity: forward, sideways and diagonal, for a given wheel speed ω. The second three rows are vehicle total pushing force: forward, sideways and diagonal, for a given wheel torque τ [13].

The model is described as reduced because in simulation a single sphere can be used to produce traction in one direction and allow free rolling on the perpendicular direction, while in practice several rollers along the circumference of the wheel are required. The usage of an Omni wheel model requires two adjustments in order to maintain the equivalence with the Mecanum wheel from the point of view of inverse kinematics matrix [13].

The first is to rotate in place each wheel with γ_i around the vertical axis so that traction force is produced along the free rotation axis of the Mecanum roller in contact with the ground,

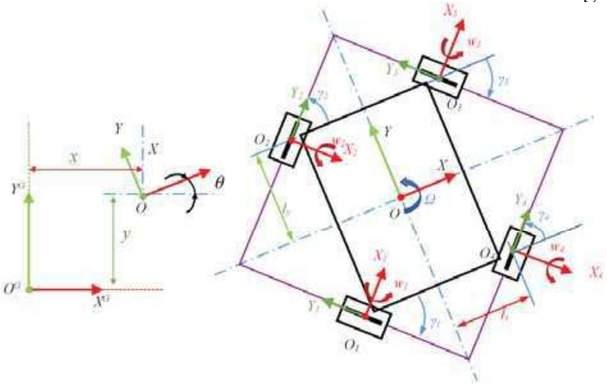


Fig. 10. OMR using the equivalent Omni wheels topology.

The second adjustment is a consequence of the γ_i rotation and requires the scaling of the reference actuation speeds ω_i by a factor of $\cos\gamma_i$ (γ_i=45°). Practically for the Omni wheel model the vehicle's Jacobian is given by (17).

$$J = -\frac{\sqrt{2}}{2R}\begin{bmatrix} -1 & 1 & -(l_x + l_y) \\ 1 & 1 & -(l_x + l_y) \\ -1 & 1 & (l_x + l_y) \\ 1 & 1 & (l_x + l_y) \end{bmatrix} \tag{17}$$

VI. EXPERIMENTAL RESULTS

In order to validate the complete model dynamics and to investigate the response of the two wheel models, a 10s long test case involving four relevant movements depicted in Fig 11. The ground speed references are provided, and each wheel angular velocity is controlled according to the inverse kinematics equations. The open loop ground speeds are then observed and represented against the reference speeds as illustrated in Fig. 12 for the equivalent model using reduced Omni wheels. The test case starts with a 0.2s settling time so that the model suspensions reach equilibrium, then it follows a translation along X axis, a stop, a translation along Y axis, another stop, a translation along the main diagonal towards origin, one more stop and finally a 180° turn and a final stop. The target translation speed on each axis is 1m/s achieved with a reference acceleration of $2.5 m/s^2$. The target angular velocity is 1rad/s and the angular acceleration is limited at 2π rad/s^2.

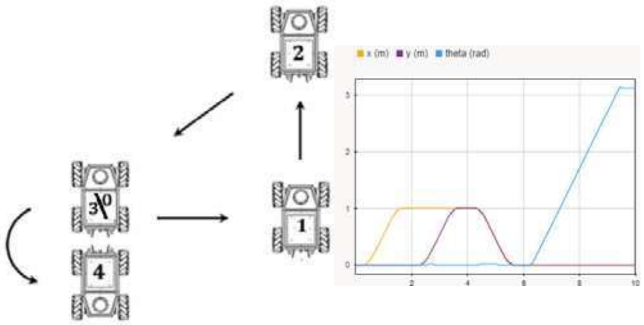


Fig. 11. The schematic map for robot navigation (left) and displacement on global world reference frame (right).

From the point of view of the simulation efficiency, the simplified model using reduced Omni wheels (against Mecanum wheel modelled with 16 spheres) performed about 10 times faster for the test case (330s/3465s) while maintaining all other components and simulation parameters the same. The simulations run on high performance Asus ROG Strix G732LXS graphics system based on Intel i9 10th generation 10980HK 8 core x 2 logical processors CPU @3.1GHz up to 5.1GHz with NVIDIA GeForce RTX 2080 Super 8GB dedicated VRAM.

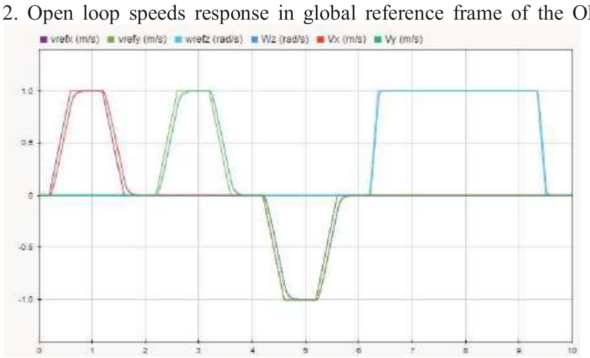


Fig. 12. Open loop speeds response in global reference frame of the OMR using the simplified model based on reduced Omni wheels.

The simulation results validate the proposed simplified model of OMR based on reduced Omni wheels.

VII. CONCLUSION AND FUTURE WORK

The research of the authors is focused on obtaining a simplified Mecanum wheel model and using the bottom-up method to build and simulate the virtual prototype of an omnidirectional mobile robot in Simscape Multibody™ in order to study its kinematic/dynamic behaviour. Analyses in this environment use all facilities given by integration of all Matlab toolboxes. In simulation process, time is a critical resource even on performant systems for complex models, as an OMR, and saving time with this simplified model offers real benefits in entire process of analyses. The modelling process is presented by main milestones among which the representation of contact forces, this plays a key role in analysis of an OMR dynamics.

The simulation results can be used for further dimensioning and optimizations of OMR and not in the least to test different control strategies.

ACKNOWLEDGMENT

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI – UEFISCDI, project no. PN-III-P2-2.1-PTE-2019-0731, contract no. 19/2020, within PNCDI III.

REFERENCES

[1] Berns K. & Von Puttkamer E., “Autonomous Land Vehicles, Steps towards Service Robots”, Wiesbaden, Germany: Vieweg + Teubner, 2009.

[2] Ingrand F., Ghallab, M., “Deliberation for autonomous robots: A survey”, Artificial Intelligence, Elsevier, 2017.

[3] Doroftei I., Grosu V., Spinu V., “Omnidirectional mobile robot-design and implementation. In book: Bioinspiration and Robotics: Walking and Climbing Robots”. Ed. M.K. Habib. Vienna, Austria: I-Tech, 2007; pp. 511-528.

[4] Adascalitei F., Doroftei I., “Practical applications for mobile robots based on Mecanum wheels - a systematic survey”, Romanian Review Precision Mechanics, Optics and Mechatronics, No. 40, 2011, pp. 21-29.

[5] Muir, P. & Neuman, C., “Kinematic Modeling of Wheeled Mobile Robots”, Journal of Robotic Systems, Vol. 4, No. 2, 1987, pp. 281-340

[6] Li, Y.; Dai, S.; Zhao, L.; Yan, X.; Shi, Y. “Topological Design Methods for Mecanum Wheel Configurations of an Omnidirectional Mobile Robot”. Symmetry 2019, 11, pp. 1268.

[7] Tătar M. O., Popovici C., Mândru D., Ardelean I., Pleşa A., "Design and development of an autonomous omni-directional mobile robot with Mecanum wheels" 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, 2014, pp. 1-6.

[8] Gfrerrer, A., “Geometry and kinematics of the Mecanum wheel”, Computer Aided Geometric Design, Volume 25, Issue 9, December 2008, pp. 784-791.

[9] Lin, L.C.; Shih, H.Y., “Modeling and adaptive control of an omni-Mecanum-wheeled robot”, Intell. Control Autom. 2013, pp.166–179.

[10] Alexandru C., “Virtual Prototyping Platform for Designing Mechanical and Mechatronic System”, chapter in book Product Design, IntechOpen 2020, DOI: 10.5772/intechopen.92801.

[11]https://uk.mathworks.com/help/physmod/sm/index.html, accessed 8/15/20.

[12] Li Y., Dai S., Zheng Y., Tian F., Yan X., "Modeling and Kinematics Simulation of a Mecanum Wheel Platform in RecurDyn", Journal of Robotics, vol. 2018, Article ID 9373580, 2018.

[13] Jigar J. Parmar & Chirag V. Savant, "Selection of Wheels in Robotics", International Journal of Scientific & Engineering Research, Volume 5, Issue 10, October-2014, pp. 339-343.