Towards a Collaborative Omnidirectional Mobile Robot in a Smart Cyber-Physical Environment

Staal, Alexander Schøn; Salvatierra, Carolina Gomez; Albertsen, Ditte Damgaard; Mahendran, Mathiebhan; Ravichandran, Rahul; Thomsen, Rasmus Finderup; Hansen, Emil Blixt; Bøgh, Simon

Published in:
Procedia Manufacturing

DOI (link to publication from Publisher):
10.1016/j.promfg.2020.10.028

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 51 (2020) 193–200

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021)
15-18 June 2021, Athens, Greece.

Towards a Collaborative Omnidirectional Mobile Robot in a Smart Cyber-Physical Environment

Alexander S. Staal[a], Carolina G. Salvatierra[a], Ditte D. Albertsen[a], Mathiebhan Mahendran[a], Rahul Ravichandran[a], Rasmus F. Thomsen[a], Emil B. Hansen[b], Simon Bøgh*[b]

[a] *Dept. of Electronic systems, Aalborg University, Aalborg Øst, DK-9220, Denmark*
[b] *Robotics & Automation Group, Dept. of Materials and Production, Aalborg University, Aalborg Øst, DK-9220, Denmark*

Abstract

This paper presents current progress in a three-fold research project, focusing on increasing the movement flexibility in smart cyber-physical environments, which are becoming more prevalent with industry 4.0. The objective is to design and develop a new research platform in the form of a highly flexible omnidirectional mobile manipulator with a full kinematics model. The research platform entails an omnidirectional autonomous mobile platform with an integrated collaborative manipulator, making it a single dynamic system. This task is split into three parts: 1) designing and constructing a flexible low cost omnidirectional mobile platform, 2) the addition of a collaborative robot arm to add task execution capabilities, 3) design the 9 DoF model for the system. This paper presents the first step towards the overall goal of collaborative omnidirectional mobile manipulators. The constructed mobile robot was able to generate and navigate an obstacle free path, using sensor data, from an initial pose to a goal pose.

Keywords: Smart Manufacturing; Omnidirectional Mobile Manipulator; Mobile Manipulator; Little Helper

1. Introduction

With the increased demand for customizable products and shorter production time from the consumer side, a need for dynamic production environments has increased along with it. It is not uncommon to have different products with different tools and moulds produced on the same production line. This could, e.g. be with the use of dynamically programmed robots, with interchangeable tooling facilitating a quick and effective switch between models being produced [1]. However, most industrial manipulators are placed in a cell, which offers the necessary automation but limits the flexibility of the workspace. One approach to solving these problems is to integrate AMRs (Autonomous Mobile Robots). The use of efficient AMRs has been one that industries have been developing and looking into in recent years. Since the first industrial revolution, companies are still trying to make productions and logistics handling efficient, faster and more flexible [1]. A group of autonomous industrial mobile manipulators named Little Helpers (LH) have been developed from Aalborg University, to tackle this task. LH mobile manipulators aim to produce flexible and efficient systems through a profound degree of automation [2]. The focus for the newer models are mainly on implementation in a reconfigurable production environment [1, 3].

The system and framework presented in this paper build upon the forgoing LHs being the eighth generation in the series, appropriately named Little Helper 8 (LH8). The purpose of LH8 is to increase flexibility through the use of an omnidirectional base with an attached collaborative manipulator modeling and creating a single dynamic system. This task is split into three parts; 1) designing and constructing a flexible low-cost omnidirectional mobile platform capable of moving autonomously from an initial position to a goal position, while avoiding obstacles. 2) the addition of a collaborative robot arm for object manipulation, 3) design the 9 DoF model for the system as a whole.

E-mail address: sb@mp.aau.dk (Simon Bøgh*).

2351-9789 © 2020 The Authors. Published by Elsevier Ltd.

10.1016/j.promfg.2020.10.028

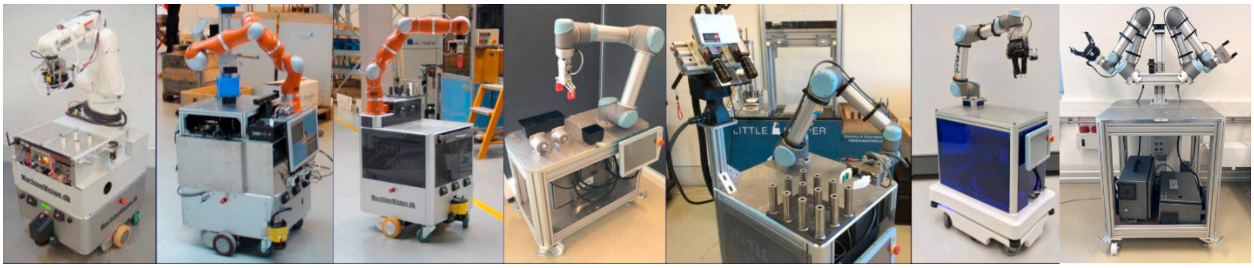

Fig. 1. The previous generations of Little Helpers from generation one through seven (LH1 on the left and LH7 on the right)[4]

2. Background

In this work the construction of an omnidirectional research platform in the context of industrial applications and earlier generations of Little Helpers, is presented. The research project Little Helper has been in the works since 2009 [2], and was designed to be independent, flexible and later integrated into the industry 4.0 concept. As of 2019, there have been seven generations of Little Helpers, with the eighth generation presented in this paper. The different generations have focused on various aspects of the problems encountered in industry. Furthermore, each generation has implemented different manipulators, sensors, mobile platform etc. This part of the project concerns designing and developing an omnidirectional mobile platform, where earlier generations have utilized non-holonomic differential drive mobile platforms such as Neobotix and MiR or a simple pushcart operated by the shopfloor worker. However, the translation of the mobile robot is limited to only two degrees of freedom in the first two earlier examples. At the same time, the downside of the pushcart is the fact that it depends on a human pushing it. Looking at previous little helpers Figure 1,Hi ditte :D purpose of these robots is to:

- LH1 is designed for logistics and multiple part feeding
- LH2 is for gesture based teaching
- LH3 is for assembly and machine tending
- LH4 is for hardware independence
- LH5 is understanding human readable instructions
- LH6 is for robot co-worker in Industry 4.0
- LH7 is for dual-arm robot co-worker

As for all the Little Helpers the common approach is to have a mobile platform and robot arm, with minor wheels and base differences. The sensors are also positioned differently to try and find the best solution for navigation around the environment. However, the main difference between the previous LH's in the implemented software.

In 1990 Carnegie Mellon University developed the omnidirectional mobile Robot Uranus. It was created together with the kinematic models and an algorithm for feedback control. The Uranus robot was able to make use of the third DoF, as it has mecanum wheels. The motion of these wheels created a holonomic translation. The downsides of the Uranus is that it does not have any exterior sensors like LIDARs to sense the environment. However, this limitation will not be present in relation to this platform, since LIDARs will be added. Furthermore, the Uranus does not have suspension, which makes it sensitive to certain floor types. [5]

Like the Uranus, the paper presented by Fuente et al. [6], uses a four wheeled mecanum wheel setup, with a rectangular shape as a basis for their omnidirectional robot, this will also be the setup used for this project. On the other hand, the setup presented by Javadi A and Mojabi [7] shows a Spherical rolling robot with an internal mechanism for forward propulsion for omnidirectional movement. Another alternative omnidirectional setup presented by Sharifi et al. [8], is a four wheeled drive/steer non-holonomic omnidirectional robot. Here omnidirectional movement is produced by having standard wheels connected to the robot frame by a rotational joint.

The paper is divided into the following sections. Section 3 presents industrially relevant use cases for highly flexible mobile manipulators. Section 4 presents the hardware and software framework of the robot. Section 5 covers the kinematics modeling of the system. Section 6 presents the results of the tests conducted, while Section 7 discusses the problems encountered. Section 8 and 9 concludes the paper.

3. Use Cases

A industrial use case would be pick and place tasks. The robot would get a signal that a component needs to go to another location or needs handling at an automated smart production line. The platform will then move there, as the collaborative manipulator on top grasps the object. It would then move to a given pose, where the object is expected. In Figure 2 the use case is illustrated, where the green line indicate the omnidirectional robot's path, while the red line indicates the necessary path of a differential drive robot in that pose. Furthermore, a use case for the mobile base without the manipulator could be transportation of e.g. components. This concept can be seen in Figure 3.

4. System Design

The LH8 mobile platform is designed to be a low-cost mecanum wheel holonomic platform. The total hardware cost amounts to € 3, 345. The most significant difference is the use of mecanum drive, whereas, the many other mobile robots use

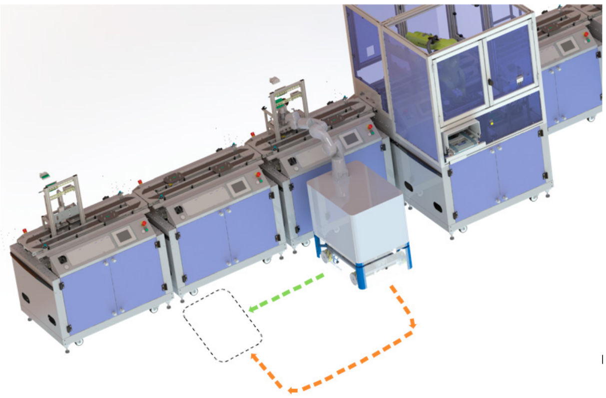

Fig. 2. Pick and place use case, where the path the LH8 would take (green line) versus a differential drive robot (red line). In the configuration shown here, the differential drive would have to take a longer path, because of it's limited amount of DoF.

Fig. 3. The mobile platform has flexibility in both the movement and the use cases as the figure shows the platform is able to handle different tasks as transporting e.g. equipment or components.

a differential drive setup to navigate. Differential drive enables non-holonomic platforms to control only two degrees of freedom, i.e. to rotate and move forward, whereas the mecanum drive, can also move laterally in addition to the other two degrees of freedom.

The CAD design for Little Helper 8 can be seen in Figure 4. The plan is to have an onboard computer, but for this prototype, a laptop is used.

4.1. Hardware Design

The final version ended up with dimensions of 0.793 x 0.568 x 0.297 m and a weight of 30 kg, which is a bit smaller than other mobile manipulator platforms. The skeleton of the robot is made using 0.02 x 0.02 m aluminum frames, which gives enough strength to lift a 200 kg payload.

On Figure 4 the LH8 can be seen. The motors, torque and RPM were selected based on the load requirements and speed

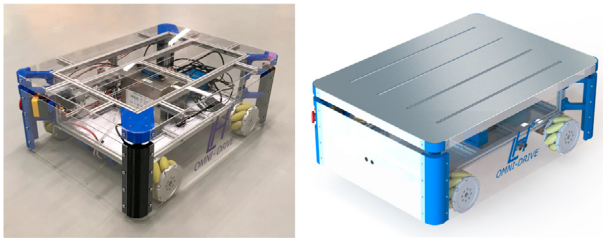

Fig. 4. On the left, the result of the LH8 can be seen whereas in the right side the final CAD model can be seen. The CAD model has an aluminum top plate which is why it is not transparent as the real one, which is currently being made and the front of the CAD model is not transparent due to the light.

required. The calculations showed a need for 300 RPM and torque of 1.58 Nm (around 50 W), but the actual power of the brushed motor used is 100 W. Two Roboclaw 15 A motor controllers are used in this project, where each can control two motors simultaneously. The four mecanum wheels are of width 0.127 m and have a combined payload capacity of 200 kg. The Roboclaw 15 A has its own internal PID, which will be used as the controller. The sensors used are the SLAMTECH RPlidar-A2 and an IMU. Both of these will be fused and used to acquire information about the environment. The LIDARs are used for the robot's perception to move around a particular area.

The next thing in the design is the electrical diagram and plan, which were done in SolidWorks Electrical. The main goal of this part of the design is to limit physically the amount of current that goes through the robot, in order to prevent short circuits and damage in electronic components. The second main goal is to set up a baseline to select and chose the components. The baseline establishes the compatibility between the components, such as the topology and operative voltage level. One example application of this would be the choice of 5 V distribution as the logical voltage level, so all the sensors that are going to be used must indicate in their specifications that they work in this level.

The components are divided in three main groups: power supply, controller and peripherals.

- Power supply: group of components for the battery and safety circuit breakers. This part of the hardware ensures that none of the rest of components gets damaged, including a safety function as well to stop the robot in case of external failures.
- Controller: this group is formed by the computer, Arduino board and the USB Link port. They contain the robot operative system and the communication interface. The communication in this project uses the following two protocols:
  – Ethernet-IP: enables the robot to communicate with the environment, allowing the mobile platform to interact with external elements, such as a wireless joystick.
  – Universal Serial Bus (USB): established between the computer and its internal components, such as the Arduino board, sensors and actuators.

- Peripherals: group of sensors and actuators. The Little Helper has two LIDARs, two motor controllers and four motors for the wheels. Apart from that, each motor controller has been assigned to two motors and its encoders.

4.2. Software Design

The navigation package for the LH8, was programmed in Robot Operation System (ROS). ROS operates by managing smaller individual programs (nodes), which perform different sub-tasks. The nodes can communicate with each other by subscribing and publishing to/from other nodes in the package. The navigation package for the robot is based on the ROS navigation stack and it requires the packages to be in a proper structure for it to run [9]. An overview of the navigation stack configuration for this project can be seen in Figure 5. It consists of the *map_server*, which provides the map created by running *gmapping* prior. *Global_costmap* contains the cost of different tiles in the map.

Local_costmap on the other hand is made by the *sensor data*, to provide information about the immediate environment [9]. The *global_planner* is an implemented A* algorithm [10], and is used for long-term planning from an initial pose to a goal pose [9]. The *teb_local_planner* is a plugin for the ROS navigation stack, used instead of the *base_local_planner*. It implements an online local trajectory planner for an omni-drive, where the initial trajectory generated by the *global_planner* is optimized during runtime. This is done with regards to minimizing the trajectory execution time[11]. The local planner also handles obstacle avoidance of dynamic obstacles in the environment. Nevertheless, *amcl* (adaptive Monte Carlo localization) is a probabilistic localization system, which uses a particle filter for localizing the robot in the map. The *sensor transforms* are used to publish information about the relationship between coordinate frames. The *odometry* is a combination of the wheel encoders and IMU, it gives an estimated change in pose over time. The *Navigation_script* is used to send navigation goals to the global planner.

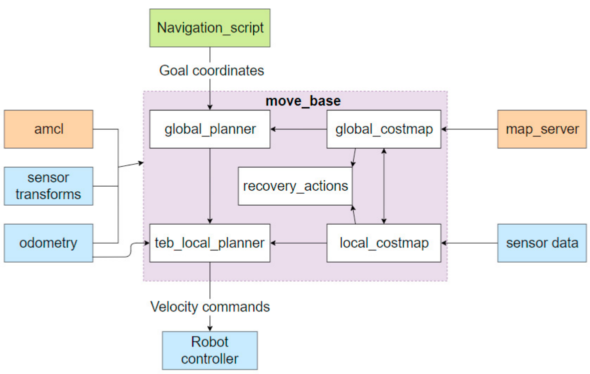

Fig. 5. The configured ROS navigation stack, where: blue boxes are platform specific nodes, orange boxes are optional provided nodes, white boxes are elements of the move_base, and green is a self-made node providing goal coordinates. [9]

4.2.1. Localization & Navigation

As explained in the Section 4.2, the robot localizes itself in the map and finds a path from an initial position to a goal position, using the navigation stack implemented in ROS. While the purpose of the former section was to give an overview, the section here explains the major components of the navigation stack namely, the Global and Local planner for path planning and the AMCL algorithm for localization.

*Global Planner - A**

A* (A star) like Dijkstra's algorithm uses a pre-constructed connectivity graph to perform a graph search aimed at finding the shortest path from a start location to a set goal. However, as an improvement to Dijkstra, A* includes a heuristic cost function that takes the distance from a node N to the goal node into account, when performing a graph search. This way, the time needed to find the shortest path is reduced due to the lower overall cost of following the route leading towards the goal. The total expected cost function $f(N)$ driving this algorithm is defined below: [12]

$$f(N) = g(N) + h(N) \tag{1}$$

$g(N)$ refers to the accumulated cost from the start node to any given node N while $h(N)$ refers to the heuristic cost from a node N to the goal node.

Local Planner - TEB

The TEB (Timed Elastic Band) local planner follows an adjustable section of the global path in successive order by breaking the local path down into a sequence of n intermediate robot poses/configurations [11], this is defined mathematically as:

$$Q = \{x_i\}_{i=0...n} \tag{2}$$

By including the estimated time needed to go from one configuration to the next (denoted by the letter τ) the objective of the TEB local planner can be defined to be the task of optimizing the sequence of configurations and the time required to transform between them. While at the same time taking both static and dynamic obstacles into account along with velocity and acceleration constraints. The TEB along with the process of optimization, is defined as follows:

$$B := (Q, \tau) \tag{3}$$

The definition of the TEB is denoted by the letter B which defines it as a combination of the sequence of robot configurations, along with the time required to transform between them.

$$f(B) = \sum_k \gamma_k f_k(B) \tag{4}$$

Here $f(B)$ denotes the objective function, which according to [11] is defined as the weighted sum of components f_k, taking the path objective (shortest or fastest route) along with velocity and acceleration constraints into account.

$$B^* = \underset{B}{argmin} f(B) \tag{5}$$

With respect to B, B^* refers to the optimized version of the TEB. The TEB parameters used for this robot application were set in according to the TEB local planner tutorials found on the ROS wiki and further tuned to optimise the LH8's behavior [11]. The most essential parameters tuned for the application are listed below:

- **max_vel_y**
  Since the robot build in this project is omnidirectional it allows for a strafing velocity in the y-direction (moving left and right). Therefore max_vel_y must be set to a positive value to allow for omnidirectional movement.
- **acc_lim_y**
  Along with a velocity in the y-direction a acceleration limit must also be set to allow for movement in that particular direction.
- **weight_kinematics_nh**
  Initially the value for this parameter is set to a high value (1000) since it specifies the weight for satisfying the non-holonomic constraints. In order for the robot to consider non-zero y-velocities, the value for this parameter should be set low. A value of 1 is recommended.
- **weight_optimaltime**
  Though tuning it was found that in order for the robot to move smoothly along its path instead of moving raggedly and getting stuck, it was necessary to chose a relatively high value for this weight (10 instead of 1). This way there was put more pressure on choosing a more time optimal path which ended up allowing the robot to get to the goal in seconds instead of minuets.

Localization - AMCL

AMCL is used to make it possible to self-localize in a given environment. It is a probabilistic localization system, which is a merged version of Monte Carlo Localization and KLD-sampling. This enables the robot to localize in a known 2D map with the use of exterior sensors. It uses the sensor data to predict/estimate the current position. [12]

$$P(x|z_1, ..., z_t) = \frac{P(z_t|x, z_1, ..., z_{t-1})P(x|z_1, ..., z_{t-1})}{P(z_t|z_1, ..., z_{t-1})} \tag{6}$$

AMCL used the derived version of Bayer's formula called Recursive Bayesian estimation formula, which can be seen in Equation 6, where x is the estimated position and z is the sensor measurement. By using this formula the AMCL has no need to store received measurement data but can process it sequentially, which makes it ideal for localization.

5. System Modelling

A mecanum wheel is a circular wheel surrounded by free moving rollers fastened to the hub in an angle of 45 degrees as can be seen on Figure 6 mecanum wheel bottom figure. The alignment of the free moving rollers enables a second translation. This translation together with three other wheels creates the omnidirectional motion, making the vehicle able to move in any direction in a 2D plane, while still having the same orientation depending on the different rotations of the wheels.

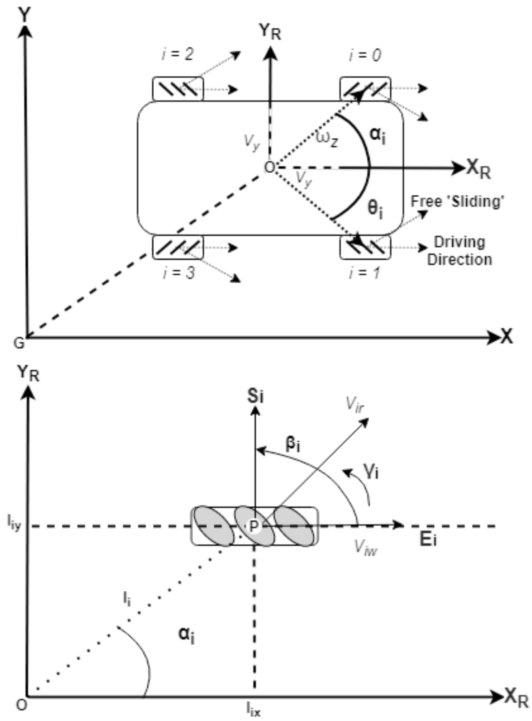

Fig. 6. The coordinate system of the mobile robot (*top*). The top figure shows the mobile robot inertial frame x, y are the coordinates of the reference point O in the inertial basis. The coordinate system for the mecanum wheel (*bottom*). The bottom figure shows the mecanum wheel frame point O center of the mobile robot to point P. Using the coordinate system the angles and sizes can be derived.

In order to develop a kinematic model of the LH8, one has to assign frames/coordinate systems for the LH8 as seen on Figure 6. The coordinate systems are also known as reference frames.

i	Wheels	α_i	β_i	γ_i	l_i	l_{iy}	l_{ix}
0	1sw	45°	90°	-45°	l	l_x	l_y
1	2sw	-45°	-90°	45°	l	l_x	l_y
2	3sw	135°	90°	45°	l	l_x	l_y
3	4sw	-135°	-90°	-45°	l	l_x	l_y

Table 1. LH8 Parameters (Solutions for mecanum 4 wheel configuration which the wheels sizes are the same). Where α_i is the angle between OP and XR, β_i is the angle S_i and X_R, and γ_i is the angle V_{ir} and E_i. l_i is the distance between wheels and the base. l_{ix} is half the distance between front wheels and l_{iy} half of the distance between front wheel and the rear wheels.

The derived kinematics for this omnidirectional mobile robot can be seen on Table 1 LH8 parameters, and can be derived by using the illustrations from Figure 6. The forward kinematics giving the robot velocities and inverse kinematics which gives the angular velocities. [13, 14]

Forward kinematics (FK):

$$\begin{bmatrix} v_x \\ v_y \\ w_z \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ \frac{-1}{l_x+l_y} & \frac{1}{l_x+l_y} & \frac{-1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{bmatrix} \begin{bmatrix} w1 \\ w2 \\ w3 \\ w4 \end{bmatrix} \tag{7}$$

Inverse kinematics (IK):

$$\begin{bmatrix} w1 \\ w2 \\ w3 \\ w4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(l_x + l_y) \\ 1 & 1 & (l_x + l_y) \\ 1 & 1 & -(l_x + l_y) \\ 1 & -1 & (l_x + l_y) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w_z \end{bmatrix} \tag{8}$$

Using the kinematics in the simulation environment (*roboclaw_complete.launch*), each wheel can be controlled independently to accomplish a motion in a specific direction. The possible directions can be seen on Figure 7 with the different wheel configurations.

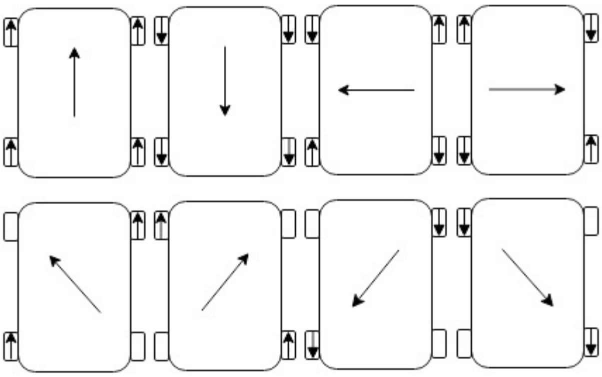

Fig. 7. By turning each mecanum wheel separately in different directions, various motions can be achieved. These motions are whats makes the mecanum wheels special.

6. Experiments and Results

In order to test the system (software and hardware) based on the requirements, different experiments were conducted. The tests can be found on Youtube[1]

- Obstacle avoidance
- Position error in Waypoint Navigation
- Payload capacity
- Maximum acceleration
- LIDAR deviation
- Battery duration

These tests were performed on a flat floor in both a narrow/small and open environment with both static and dynamic obstacles. The obstacle avoidance and the waypoint navigation tests were performed both in simulation and in the real world. The experiments were done using the newest iteration of the LH8, which can be seen in Figure 4.

6.1. Simulations

The simulations were performed in Gazebo, and the path planner used for the holonomic navigation is TEB local planner. The CAD model was imported and adjusted to fit the Gazebo ROS environment, which can be seen in the center of Figure 8. The robot was able to accurately navigate the environment with

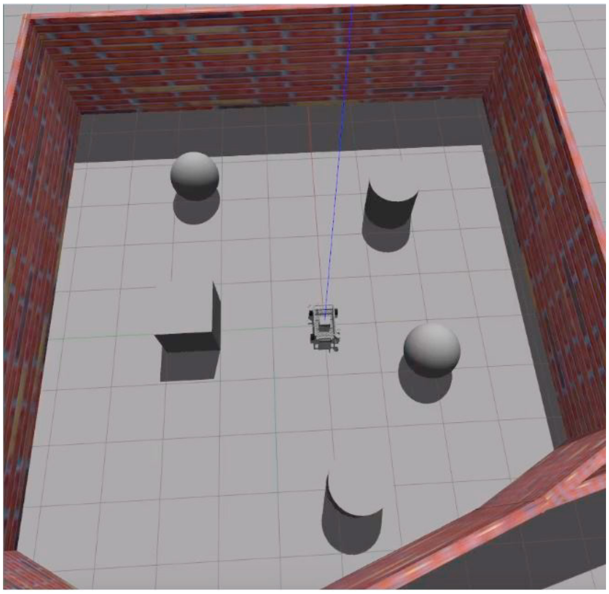

Fig. 8. By default, the mobile platform is located at the center of the bounded simulation environment and can through commands move around in it. Five obstacles, of different shapes, are placed in the scene, for testing of obstacle avoidance algorithms.

obstacles while moving at a max speed of 0.5 m/s. The position error of the point reached was accurate to the tolerance specified in the move base navigation which was 0.2 m, but this is due to the accurate LIDAR and odometry readings. On Figure 9, the LH8 CAD model was simulated on Gazebo/Rviz using the navigation stack. The simulation was successful since the robot follows the navigation points.

[1] https://www.youtube.com/playlist?list=PLSG9gXgVHC2NteGCGuiLw8a74-lBwlP6O

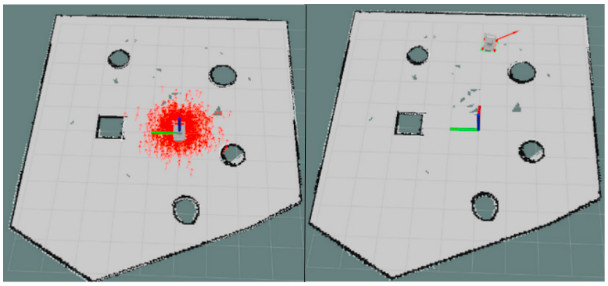

Fig. 9. It can be seen that the predicted position (*red blob of arrows*) in the left figure is very imprecise as this is the initial position. However, as the robot moves around in the environment the LIDAR data and the map are compared to reduce the position uncertainty and the prediction gets more precise (*right*).

6.2. Test Results

In a non-ideal environment, the robot is prone to position errors due to inaccuracies in the sensor measurements. These tests were repeated for a maximum of 20 iterations and then averaged to get the result. The results from testing showed that:

- The robot was able to avoid obstacles. This was tested by introducing an obstacle in the path it previously took and observe how it re-plans its path.
- The robot can navigate from an initial position to a goal position with a max error margin of 0.33 m. It was seen that the error would increase over time, due to drifting in the odometry.
- Can carry a payload of maximum of 170 kg (calculated based on individual wheel payload capacity).
- The maximum acceleration at robots max speed was found to be 19.73 m/s^2 (measured using IMU).
- The deviation tests on the LIDARs showed an error margin of 0.009 m. This was done by checking the error deviation in the distance measured while a particular angle was observed.
- The battery on the robot can operate a normal work shift of eight hours. It was tested by having the robot doing a task until battery charge reached shutdown levels. However, the laptop used for controlling the platform, could only function for approximately one hour.

7. Discussion

As the robot software is running on a laptop, the operating time is decreased to half an hour, to an hour. Thus, the LH8 can perform obstacle avoidance as well as self-localization in a known environment. The use of omnidirectional wheels has the benefit of providing the third DoF, but if the floor is not flat or has low friction, the wheels can either get stuck or create slippage as the velocity increases. A result of this is the 0.33 m max deviation from the set goal location. Doing testing, it was found that the deviation would increase over time as the robot went back and forth between two set locations. However, the error margin did not pass 0.33 m due to the AMCL algorithm slightly correcting the position. Meanwhile, to avoid oscillation around the goal location, an x-y goal-tolerance of 0.2 m was set along with 0.1 m theta goal-tolerance, this further contributes to the resulting error margin.

Some of the similar mobile robots to Little Helper was taken inspiration from are Neobotix MP0-500 and MiR 100 whose payload capacity are 50 kg and 100kg respectively which is less that the LH8's capacity i.e. 170 kg. But the MiR and the MPO-500 robot has better position accuracy i.e. +-50 mm which is much better than the 0.33 m accuracy of the LH8 since the robots use much better sensors like the SICK Lidars.

8. Conclusion

The industrial sector is moving towards smart cyber-physical environments, which requires new solutions for challenges such as transport, material manipulation and human-robot collaboration in smart production.

This paper focused on the first part of building and integrating a new research platform of highly flexible and low-cost design. The flexibility come in the form of holonomic movement, which makes it possible to complete tasks more efficiently compared to e.g. differential drive. The plan is to have the finished platform be implemented in a cyber-physical environment, but that element was not an aspect of this part of the project.

The omnidirectional wheels of the robot allows movement in every direction. Hence, the software provides accurate localization and robust navigation. Overall, the LH8 self-localizes and moves autonomously in a pre-made map of the environment. The combined payload of the wheels amount to 200 kg. This was chosen with the approximate weight of the collaborative manipulator, equipment and desired handling payload in mind. Furthermore, the chassis was designed to withstand the force applied by the combined weight of the finished platform. The hardware cost ended up being approximately € 3,300 in total.

Finally, based on the simulation and real life tests it can be seen that the implemented path planner was to generate an obstacle free path, using sensor data, from an initial pose to a given pose. However, it was also observed that the error in the estimated position in relation to the map of the testing environment, would increase over time. This was mainly due to drift in the sensors. All in all, this project resulted in a step towards a more flexible collaborative mobile robot, where the next step will be to add a collaborative manipulator and improve upon the mobile platform.

9. Future Work

The mobile platform will continue to be improved upon, with suspension for minimizing slippage on uneven ground, light for telling the condition, computer connected to the battery, a charging station etc. This is however just improvements, the next step is to add a collaborative manipulator on top of the mobile platform (see Figure 10) and make it a single dynamic system, where both the mobile platform and the manipulator work as one unit. This will enable the robot to reach goal poses faster and more efficiently, by making use of 3 DoF provided by the mobile platform.

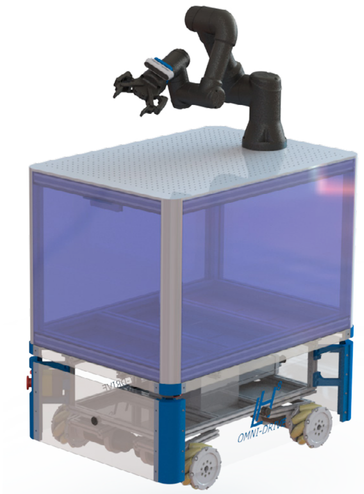

Fig. 10. Illustration showing the future end result of the Little Helper 8 with a manipulator added on top.

References

[1] C. Bartodziej, *The Concept Industry 4.0: An Empirical Analysis of Technologies and Applications in Production Logistics*, ser. BestMasters. Springer Fachmedien Wiesbaden, 2016.

[2] O. Madsen, S. Bøgh, C. Schou, R. Andersen, J. Damgaard, M. Pedersen, and V. Krüger, "Integration of mobile manipulators in an industrial production," *Industrial Robot: An International Journal*, vol. 42, pp. 11–18, 01 2015.

[3] R. Andersen, E. Hansen, D. Cerny, S. Madsen, B. Pulendralingam, S. Bøgh, and D. Chrysostomou, "Integration of a skill-based collaborative mobile robot in a smart cyber-physical environment," vol. 11. Elsevier, 2017, pp. 114–123.

[4] J. Filtenborg Buhl, R. Grønhøj, J. Kjær Jørgensen, G. Mateus, D. Pinto, J. Krunderup Sørensen, S. Bøgh, and D. Chrysostomou, "A dual-arm collaborative robot system for the smart factories of the future," in *29th International Conference on Flexible Automation and Intelligent Manufacturing*, 2019.

[5] M. Blackwell, "The uranus mobile robot," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-91-06, September 1990.

[6] M. L. l. Fuente, D. C. Santos, E. I. D. l. Garza, I. A. T. Saca, D. T. Garza, and F. A. H. Aleman, "Design and control of a four wheeled omnidirectional robot with image processing," in *2019 7th International Engineering, Sciences and Technology Conference (IESTEC)*, Oct 2019, pp. 514–519.

[7] Amir Homayoun Javadi A and P. Mojabi, "Introducing august: a novel strategy for an omnidirectional spherical rolling robot," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 4, May 2002, pp. 3527–3533 vol.4.

[8] M. Sharifi, M. Young, X. Chen, D. Clucas, and C. Pretty, "Mechatronic design and development of a non-holonomic omnidirectional mobile robot for automation of primary production," *Cogent Engineering*, vol. 3, 10 2016.

[9] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010-05, pp. 300,307.

[10] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, no. C, pp. 59,69, 2014.

[11] C. Roesmann, W. Feiten, T. Woesch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *ROBOTIK 2012; 7th German Conference on Robotics*, 2012, pp. 1–6.

[12] I. R. N. Roland Siegwart and D. Scaramuzza, *Introduction to autonomous mobile robots.*, elektronisk udgave. -2nd ed. ed., ser. Intelligent robotics and autonomous agents. Cambridge, Mass: MIT Press, 2011.

[13] H. Taheri, B. Qiao, and N. Ghaeminezhad, "Kinematic model of a four mecanum wheeled mobile robot," *International Journal of Computer Applications*, vol. 113, no. 3, pp. 6–9, 2015-01-01.

[14] P. F. Muir and C. P. Neuman, "Kinematic modeling of wheeled mobile robots," 1986.