Integration of Navigation, Vision, and Arm Manipulation Towards Elevator Operation for Laboratory Transportation System Using Mobile Robots

Submitted: 2nd August 2017; accepted: 16th January 2018

Ali A. Abdulla, Mohammed M. Ali, Norbert Stoll, Kerstin Thurow

DOI: 10.14313/JAMRIS_4-2017/35

Abstract:
In the automated environments, mobile robots play an important role to perform different tasks such as objects transportation and material handling. In this paper, a new method for a glassy elevator handling system based on H20 mobile robots is presented to connect distributed life science laboratories in multiple floors. Various labware and tube racks have to be transported to different workstations. Locating of elevator door, entry button detection, internal buttons recognition, robot arm manipulation, current floor estimation, and elevator door status checking are the main operations to realize a successful elevator handling system. The H20 mobile robot has dual arms where each arm consists of 6 revolute joints and a gripper. The gripper has two degrees of freedom. Different sensors have been employed with the robot to handle these operations such as Intel RealSense F200 vision sensor for entry and internal buttons detection with position estimation. A pressure sensor is used for current floor estimation inside the elevator. Also, an ultrasonic proximity distance sensor is utilized for checking the elevator door status. Different strategies including HSL color representation, adaptive binary threshold, optical character recognition, and FIR smoothing filter have been employed for the elevator operations. For pressing operation, a hand camera base and a new elevator finger model are designed. The elevator finger is resolved in a way to fit the arm gripper which is used also to manipulate the labware containers. The Kinematic solution is utilized for controlling the arms' joints. A server/client socket architecture with TCP/IP command protocol is used for data exchange between Multi-Floor System and the H20 robot arms. Many experiments were conducted in life science laboratories to validate the developed systems. Experimental results prove an efficient performance with high success rate under different lightening condition.

Keywords: *mobile robot, multi-floor, elevator handler, floor estimation, labware transportation system, kinematic analysis, robotic arm control, object detection and localization, Intel Real sense F200 sensor*

1. Introduction

In recent years, the development of transportation systems based on mobile robots has progressed rapidly to meet requirements such as high precision, routine task execution, transportation in hazardous areas, and low-cost manufacturing. For automated life science laboratories, transportation task based mobile robots requires high precision of movement, the handling of lab equipment, the integration with automation islands, and scheduling the robot's activity in accordance with the main laboratory's control schedule. Most of the earlier developed transportation systems based on mobile robots work only on a single floor of restricted size [1]–[3].

In this work, H20 mobile robots (Dr. Robot, Canada) are used for labware transportation in multiple floors of life science environment. H20 robot is a wireless networked autonomous humanoid mobile robot. It has a PC tablet, dual arms, and an indoor GPS navigation system (see Figure 1).

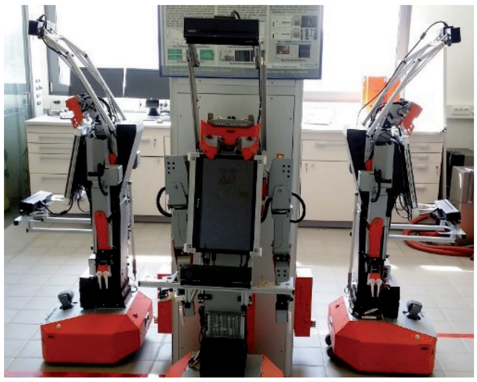


Fig. 1. H20 mobile robots

Since the H20 mobile robot does not have the ability to climb stairs, an elevator operation management system is an essential aspect of moving between different floors in a multi-floor transportation system. The elevator handling system must include many operations such as the locating of the elevator door and its status (open/closed), the detection and the pressing of entry and destination floor buttons, and the estimation of the current floor.

The target button position can be estimated using a suitable visual sensor with a proper detection technique. The detection process includes the extraction of specific features from the image related to the target. In general, these features are divided into two categories. The first is the appearance features of objects such as the color, form, and size. Color detection can be performed using different color systems such as RGB, HSL, HSV, and YCbCr. The second category is the local texture, shape, and edges of the target itself.

The required features have to be extracted from the image to be matched with the features in the database related to the target.

To find the target position, stereo vision system and 3D camera are considered the most appropriate visual sensors. But 3D cameras are more preferable because they provide directly the depth data without applying any complicated processes for the images as in the case of stereo vision.

Many methods have been used to handle elevator operations. Each has achieved some degree of success, albeit with limitations. A template matching technique has been utilized to detect elevator buttons which achieved a success rate up to 85% for the entry button and 63% for floor buttons inside the elevator [4]implementation and experimental evaluation of a semi-humanoid robotic system for autonomous multi-floor navigation. This robot, a Personal Robot 2 named Charlie, is capable of operating an elevator to travel between rooms located on separate floors. Our goal is to create a robotic assistant capable of locating points of interest, manipulating objects, and navigating between rooms in a multi-storied environment equipped with an elevator. Taking the elevator requires the robot to (1. The disadvantage of this method is the excessive time required for buttons detection (4.5 s and 4.3 s, respectively). An artificial neural network has also been proposed to achieve the best matching by discarding weak candidates for entry and internal buttons detection [5]. However, this method has not been validated in a real elevator environment. Multiple symbol methods have been adopted for both external and internal buttons panel detection, while a combination of image processing techniques were utilized to recognize external and internal elevator buttons [6]. These methods achieved satisfactory performance for button recognition but did not provide real coordinates, which are significant for guiding the robot's arm in button pressing operations.

For current floor estimation, many techniques can be utilized, such as Radio-Frequency Identification (RFID) [7], the Received Signal Strength (RSS) of a wireless network [8], and a height measurement system [9]. Each approach can realize floor estimation but with some disadvantages and limitations. The RFID technique requires high costs since RFID antennas have to be installed on each floor [7]. The floor estimation based on RSS has the same disadvantage of being very expensive since it would require at least 4 wireless network sources on each floor to achieve an acceptable success rate [8]. On the other hand, the height measurement system depends on pressure sensors for floor estimation has a low-cost. It requires only one sensor attached to the robot for all floors [9]. But frequent recalibrations have to be performed to overcome the problem of wide daily variations in atmospheric pressure.

The design of elevator finger is one of the important issues which plays a main role to achieve the buttons pressing operation. Some factors have to be taken into consideration to design a suitable finger model like the structure of end effector with gripper and the characteristics of the buttons like its shape and material, etc.

For pressing operation of elevator buttons, the robotic arms have to be controlled reliably. This requires the position estimation of the target button with respect to the arm base followed by using an accurate kinematic model to move the arm end effector to the target in a safe path. The kinematic analysis is how to describe the arm links motion without considering its forces. There are two types of kinematic models: forward kinematics (FK) and inverse kinematics (IK). The FK model is a mathematical model used to calculate the end-effector pose relative to the arm base according to the given joints' angles. On the other hand, the IK model is a mathematical model used to calculate the values of joints angles according to the given end-effector pose with respect to the arm base. The IK model considers as an important issue to enable the arm end-effector to reach the desired position accurately. Generally, the IK problem can be solved using two approaches: analytic [10]–[12] "container-title": "IEEE Transactions on Aerospace and Electronic Systems", "page": "695-706", "volume": "AES-20", "issue": "6", "source": "IEEE Xplore", "abstract": "A geometric approach for deriving a consistent joint solution of a six-point PUMA1 robot is presented. The approach calls for the definition of various possible arm configurations based on the link coordinate systems and human arm geometry. These arm configurations are then expressed in an exact mathematical way to allow the construction of arm configuration indicators and their corresponding decision equations. The arm configuration indicators are prespecified by a user for finding the joint solution. These indicators enable one to find a solution from the possible four solutions for the first three joints, a solution from the possible two solutions for the last three joints. The solution is calculated in two stages. First a position vector pointing from the shoulder to the wrist is derived. This is used to derive the solution of the first three joints by looking at the projection of the position vector onto the xi-1-yi-1(i = 1,2,3, and numeric [13]–[15] this paper investigates methods of resolved motion rate control (RMRC. Also, the arm joints' limits and reachable workspace have to be taken into the consideration to control the robotic arm. Normally, the analytical approach is more preferable because all the exact solutions can be found and it is computationally faster than the numerical approach.

In this paper, an elevator operation management system for H20 mobile robots is presented. In this system, the passive landmarks with the stargazer sensor module are employed to localize the robot in front of the elevator and inside it. The Intel RealSense F200 vision sensor is utilized for detection and position calculation of entry and internal buttons. Several image processing techniques such as HSL color representation, adaptive binary filter, shape and color filters are used for entry button detection. In addition to the previous techniques, the Optical Character Recognition (OCR) is employed to specify the destination floor label for internal buttons recognition. For pressing operation, a new finger design for the H20 mobile robot is presented and the kinematic solution is employed to control the arm joints movements. A robust current floor estimation based on height measurement is utilized. As a height system hardware, the LPS25HP pressure sensor with STM32F411 microcontroller is configured and programmed. The FIR smoothing filter with an adaptive calibration stage is utilized to overcome the problem of oscillation and wide daily variations in atmospheric pressure. Finally, the ultrasonic distance sensor is utilized for checking the elevator door status.

This paper is organized as follows: the architecture of the system is described in section 2, while section 3 explains the selection of the vision sensor. The working process, the kinematic solution, and finger design are detailed in sections 4–6. Section 7 describes the outside elevator procedure while section 8 explains the inside procedure. The complete button pressing operation is explained in section 9. The final section summarizes the results.

2. System Architecture

In recent years, researchers at the Center for Life Sciences Automation in Rostock (Germany), have developed a Hierarchical Workflow Management System (HWMS) to manage the entire process for establishing a fully automated laboratory [16]. The HWMS has a middle control layer which is a Transportation and Assistance Control System (TACS), including the Robot Remote Centre (RRC). The RRC is the layer between the Robot Board Computer (RBC) and the TACS. The RRC is a GUI developed to manage transportation tasks as follows. The transportation task is received and will be forwarded to the appropriate robot with the highest battery charge value. Finally, the transportation results are reported back to the TACS.

The Multi-Floor System (MFS) is the core component of the RBC level [17]. It has been developed to execute the transportation tasks in multiple locations in automated life science laboratories. To execute the transportation task, the MFS is developed to realize the functions of a multiple floor navigation system with a Robot Arm Kinematic Module (RAKM) [18], Elevator Handler System (EHS) [19] a new system is presented to manage the elevator operations. A Wi-Fi socket is established to connect with the ADAM module for calling the elevator and requesting the destination floor. This technique does not provide any feedback on the elevator's door status or its current floor which in some situations can make the robot losing its way to the destination. Computer vision can be utilized to identify the current floor. In some special situation (human obstacle between the robot camera and the floor number indicator, difficult light conditions etc., and Collision Avoidance System (CAS) [20].

A multiple floor navigation system including mapping, indoor localization, path planning, an Internal Management Automated Door Controlling System (IMADCS), communication system, Internal Battery Charging Management System (IBCMS) has been reported [17]. The MFS has been integrated with the Elevator Handling System (EHS) to interact with the elevator environment and to realize the necessary information regarding button positions, current floor number, and elevator door status. For grasping/placing/pressing operation, the MFS guides the robot to the destination position and sends a grasp, place, or press order to the RAKM with the required information. The Robot motion center provides the way to control the robot hardware by receiving the robot sensors readings and executing the required movements. The client-server connection architecture module (asynchronous socket) is used to control the

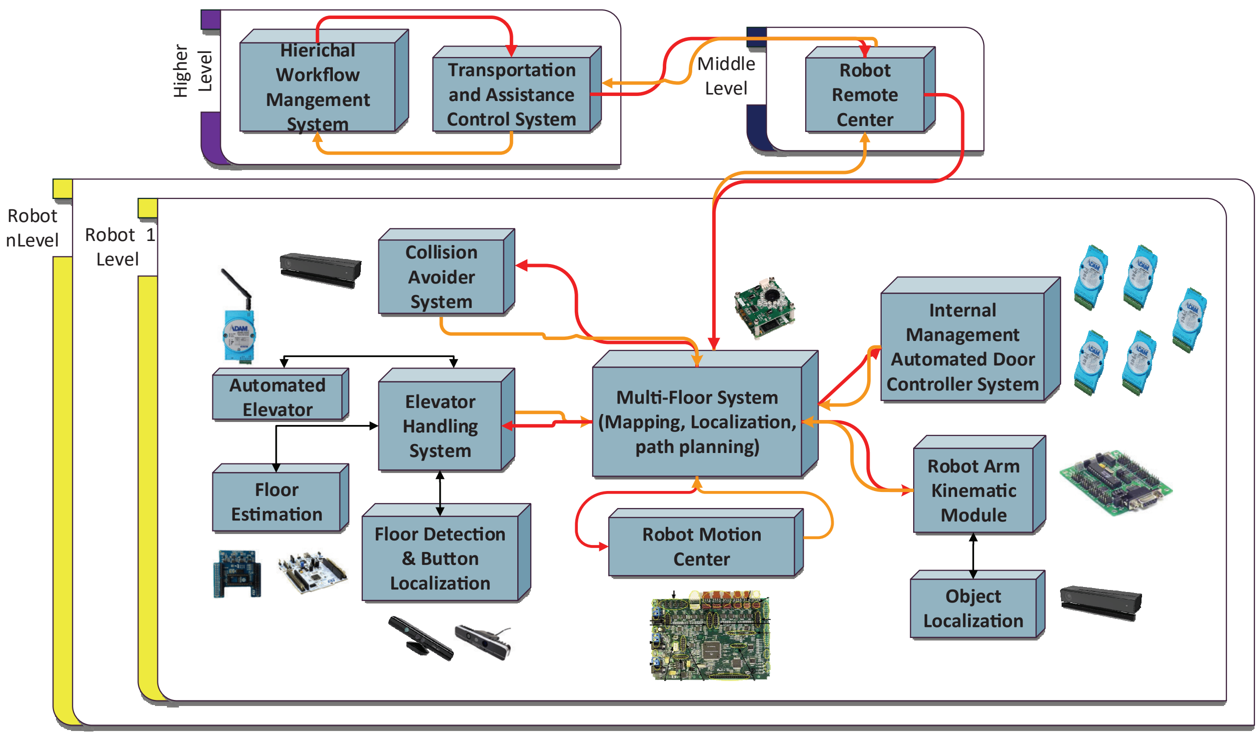
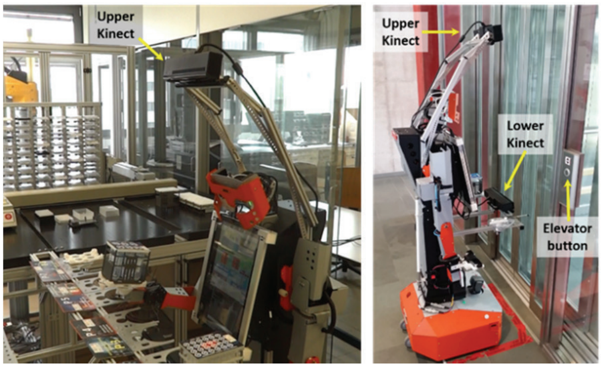


Fig. 2. Complete structure for mobile robot transportation system

interaction of the MFS with these sub-systems over Ethernet. A TCP/IP command protocol based server-client structure is used to guarantee the reliability and the expandability. So, any kind of devices can be added into the communication network conveniently with a new IP. Figure 2 shows the full structure of the mobile robot transportation system.

3. Selection of Suitable Visual Sensor

Multiple H20 mobile robots are used to perform multiple labware transportation between different workstations. Labware manipulation requires an accurate process to achieve the grasping and placing tasks in a safe way [21]. Two Kinect sensors V2 are used with H20 mobile robot as shown in Figure 3. An



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

Fig. 3. a) H20 robot in grasping task; b) the H20 robot is in front of the elevator door

upper Kinect is installed on a holder with a suitable tilt angle to provide a clear view of the workstation which has multiple positions of labware containers as shown in Figure 3.a. The upper Kinect is used to identify and localize the required labware and holder for the manipulation task [22] "container-title":"2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI. Polarization and intensity filters are fixed on the Kinect camera to decrease the effects of sunlight and glossy light on the identification process. The upper Kinect can't be used for buttons detection because it doesn't provide a clear view for the panel. Its position on the holder with its tilt angle makes the view direction of the camera somehow parallel with the button panel (see Figure 3.b). On the other hand, the lower Kinect is used for obstacle avoidance and human-robot interaction [23]. It is not possible to detect the elevator button using the lower Kinect because it is not within the Kinect FOV as shown in Figure 3.b. This requires to increase the distance between the robot and the button panel to detect the button. But, in this case, the button will be out of the arm workspace. Furthermore, it is not feasible to move the robot closer to the panel after the step of button detection and localization. This is related to the lack of accurate feedback of robot pose after the movement step [24], where there are always errors in robot positioning after reaching the required location [18]. To cope with all these issues, a 3D hand camera is used to perform the task of pressing the elevator buttons instead of the Kinect sensors. The Intel RealSense F200 camera is fixed on the H20 arm where an appropriate camera base has been designed for this reason as shown in Figure 4.

Fig. 4. 3D hand camera with its base

The Intel RealSense F200 camera was developed by Intel Company to work over short distances based on coded light technology to extract the depth information. It has a full HD camera resolution (1,080 pixels at 30 FPS), a working range (20–120 cm) for depth sensing, gesture tracking at (20–60 cm), face recognition at (25–100 cm), and a microphone array for speech recognition. Figure 5 shows the components of RealSense F200 camera where it works under Windows 8, or 10 operating system. Since this camera extracts the depth information at a short range (20–120 cm), it has a limited view. This feature can be considered as a challenge because the camera has to be located at a suitable position in front of the buttons panel. Therefore, it requires a high robot positional accuracy to make the button within the camera FOV.

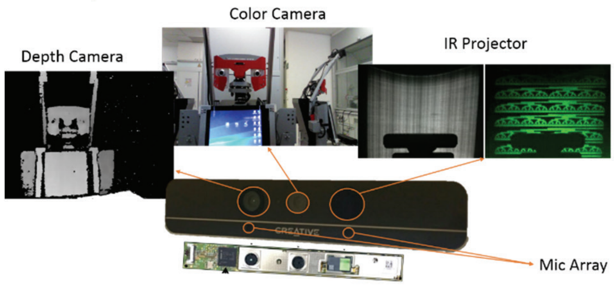


Fig. 5. RealSense F200 camera components

4. Working Process

The essential processes for the mobile robot to use the elevator are as follows. The elevator door has to be located, starting by finding the way to the elevator entrance based on a navigation system developed earlier [17]. Then the elevator entry button should be identified by finding the button landmark and its center point in image coordinates and extracting the position information. This information is subsequently fed to the next stage of pressing the entry button. The button position is sent to the arm's kinematic module which performs the required calculation to press the button and informs the MFS if the pressing operation is not possible. After this, the robot must enter the elevator, move to a predefined position after the door has opened. Afterward, the robot localizes itself based on recognizing a landmark inside the elevator. Next, the control panel which includes the buttons of floors numbers has be found and the destination floor must be chosen. Then, the robot moves towards the control panel according to the ceiling landmark, finds the required X, Y, Z coordinates of the target button and then feeds this information to the next stage. The required calculation is then performed to press the destination floor button, using the kinematic module. After pressing the floor button, the elevator moves to the required floor and its door eventually opens. The reached destination floor is checked and the destination map is loaded. Then, the robot can finally leave the elevator to complete the transportation task in the required floor.

Object detection process requires separating the image into regions to extract useful information which leads to find the target. The required button can be detected according to its specific global or local features to be distinguished from the other objects in the view. Specific object features like its color, shape, or texture can be used to identify the target in the view. In case, that the target doesn't have distinct adequate features for detection, attached marks can be used. The position of mark or label can be considered as a reference to localize the target. This position has to be transformed to find the button position related to the camera. Then, another transformation step has to be performed to find the button position related to the robot finger which will press it. Afterward, the kinematic model calculates the required joints values to guide the arm finger to the button. The design of the finger is one of the most important issues which has to be taken into consideration to guarantee a successful task. Figure 6 shows the architecture of a pressing system for elevator buttons.

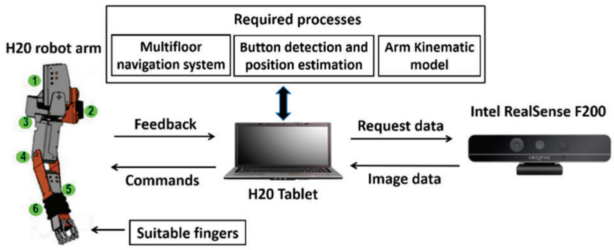


Fig. 6. Architecture of button pressing system

The Multi-Floor transportation with mobile robots is very important to connect workstations in different floors which in turn increases the productivity and saves human resources. This includes also the collective work of stationary robots and mobile robots to transport the labware to multiple stations to perform different analytical tests [25]. Figure 7 shows an example of workstation which is combined between H20 mobile robots and Orca stationary robot. The H20 mobile robot uses the elevator to move to other floors for delivering the labware to the required workstations.

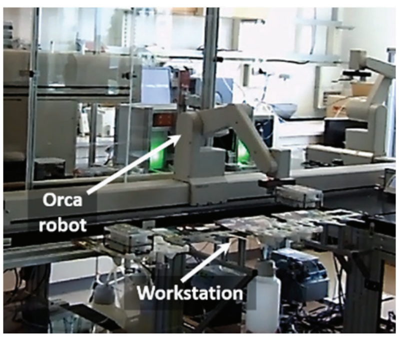


Fig. 7. Combined workstation of H20 robots

5. Kinematic Analysis of H20 Arms

The H20 mobile robot has dual arms with 6-DOF and 2-DOF grippers. Figure 8 shows the joints structure of arms where the values of d3, d5, and de are 0.236 m, 0.232 m, and 0.069 m, respectively [18]. The Denavit-Hartenberg (D-H) representation is used to describe the translation and rotation relationship between the arm links. According to D-H notation, there are four parameters to analyze the robotic arm: the link length (ai-1), the link twist (αi-1), the link offset (di), and the joint angle (θi) where (i) refers to the link number [26]. By following the D-H rules, the homogeneous transformations between adjacent links are defined. The D-H parameters and the limit of each joint are described in Table 1 [18].

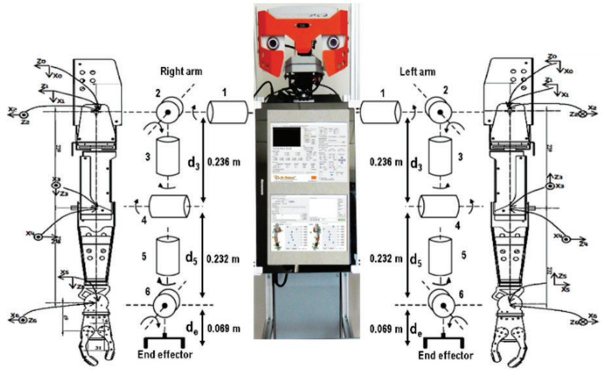


Fig. 8. H20 arms structure [27]

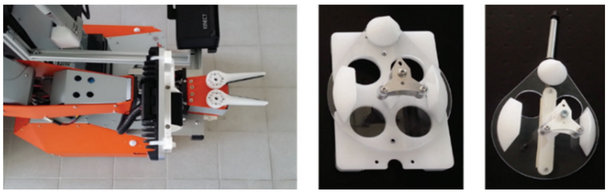
Table 1. D-H parameters and joints limit

Left and Right Arms						
θ_i	$\alpha_{(i-1)}$ (L)	$\alpha_{(i-1)}$ (R)	$a_{(i-1)}$ (LR)	$d_i(m)$ (L)	$d_i(m)$ (R)	Joints limit (LR)
θ_1	0°	0°	0	0	0	-20°~192°
θ_2	90°	-90°	0	0	0	-200°~-85°
θ_3	90°	-90°	0	-0.236	0.236	-195°~15°
θ_4	-90°	90°	0	0	0	-129°~0°
θ_5	90°	-90°	0	-0.232	0.232	0°~180°
θ_6	-90°	90°	0	0	0	-60°~85°

The analytic IK solution of H20 arms has been found using the reverse decoupling mechanism method [18], [28], [29]. The strategy of this method depends on viewing the kinematic chain of the manipulator in reverse order with decoupling the position and orientation. In other words, the arm can be viewed in reverse order so that the pose of the arm base can be described relative to the end effector. The kinematics solutions have been validated and simulated using MATLAB with robotics toolbox [18]. Also, a selecting algorithm is used to choose the suitable solution within multiple solutions. Moreover, the IK solutions for three cases of singularity have been found [18]. The singularity is the case when some joint axes are aligned with each other. This leads to eliminate one or more degrees of freedom.

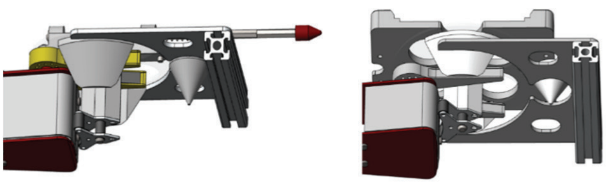
6. Finger Design

For labware transportation with the mobile robots, suitable grippers with labware containers have been designed [21]. Since the labware contains chemical and/or biological components, any kind of spilling has to be avoided. Therefore, a specific design of grippers and labware containers is required to guarantee secure grasping and placing operations. Figure 9.a shows the designs of the grippers and Figure 9.b shows the bottom design of labware containers [21]. Related to the operation of button pressing, the arm gripper can't be used directly for this task. This is related to the arm workspace and robot position in front of the button panel. The robot has to be at a specific distance to the panel to press the button and to enter the elevator. Therefore, an elevator finger with a specific length has been designed as shown in Figure 9.c. The bottom design of labware container and elevator finger is the same to fit the gripper design. Figure 10 shows the 3D model of elevator finger with labware container and how the gripper grasps them. For labware transportation and buttons pressing purposes, a holder has been installed on the H20 body. This holder has two placing positions for left and right arms to be used for secure transportation of labware and elevator finger. The elevator finger has to be placed on the right position of the holder to be manipulated by the right arm for pressing the required button. Figure 11 shows the holder and how the elevator finger sits on it. The finger has a spring in the middle and a rubber end tip for reliable pressing tasks.



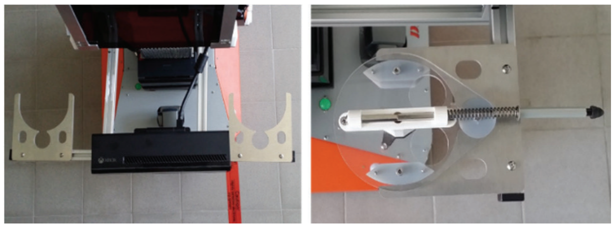
(a) (b) (c)

Fig. 9. Design of a) arm gripper, b) labware container bottom, c) button finger bottom



(a) (b)

Fig. 10. 3D model of, a) gripper with elevator finger, b) gripper with labware container [21]



(a) (b)

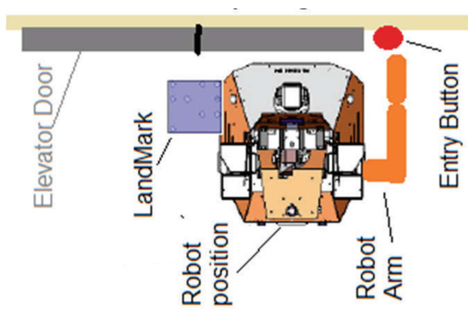
Fig. 11. a) Holder for elevator finger and labware container, b) the elevator finger sits on the holder

7. Elevator Entry Process

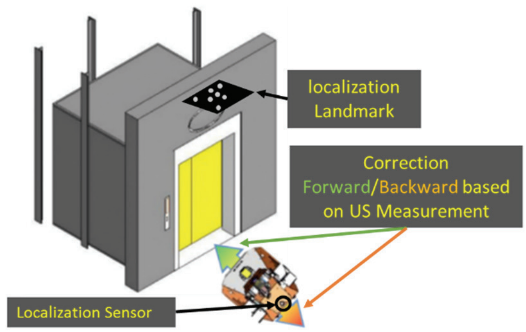
This section covers the robot requirements before entering the elevator. This includes finding the elevator entrance door, the recognition of the entry button, the determination of the real position, the detection of the elevator door's status, and finally going inside the elevator. All these aspects are explained in detail in the following sections.

7.1. Movement to Elevator Area

The first step in elevator handling is the determination of the elevator zone. Then, the required analysis is performed and processes are activated to reach the target destination. This stage is performed based on predefined waypoints on the map of the elevator area. This depend on the reflective artificial landmarks installed near the elevator to read the exact position and direction of the robot. Based on this information, the robot moves until it reaches the best position, as shown in Figure 12.a. This movement is the basis for the next two stages which are entry button detection and elevator entrance door status rec-



a) Entry position movement idea



b) Higher accuracy robot positioning

Fig. 12. Robot position outside elevator

ognition. The H20 mobile robot arms have a limited workspace, and thus it is necessary to control the robot until it reaches a desired predefined position with high accuracy so that the arm is able to work within its workspace. The arm should move in a vertical straight path to the panel to press the button where its workspace is increased or decreased according to the robot's position near the elevator area. The robot must locate itself in a specific position that allows to press the entry button and to enter the elevator after changing its orientation when the door opens. Therefore, a hybrid method is used to achieve a higher positional accuracy, which utilizes a correction function based on SGM (see Figure 12.b) and a motor encoder correction based on an ultrasonic distance sensor as follows. First, the movement core is utilized until the predefined position is achieved. The movement repeatability reaches 5 cm in the x-axis and 3.5 cm in y-axis [17]. Then a correction procedure is performed to reach the predefined position with a repeatability of 3 cm in x-axis and 2 cm in y-axis based on the last movement direction. Finally, the ultrasonic distance sensor with the motor encoder is employed to ensure that the robot reaches the exact distance to the door within an accuracy range of 1–2 cm.

7.2. External Button Detection

The glass elevator used has a passenger cabin with two sides made of glass panels and metal frames. The transparent surfaces add a big challenge for entry button detection, especially in sunny weather. The entry button detection technique uses computer vision and a depth sensor. It uses an Intel F200 sensor to acquire the RGB image and depth information. Since the entry button and its panel are made from the same reflective material, the detection of the button is difficult to be realized. Therefore, a landmark has been fixed close to the entry button with a specific shape and color to perform an easy and applicable recognition process by the mobile robot as shown in Figure 13. The elevator entry button detection process starts by initializing the F200 camera with the required frame rate, image resolution, and depth resolution. Then an image is captured from the F200 RGB sensor and a filtering step is applied to remove the unwanted band of colors. The edges of the detected object are extracted to find the shape of the button landmark. Afterward, the position of mark center point is calibrated to be

Fig. 13. Elevator entry button and its colored landmark

related to the button center point Various filters related to shape recognition can be applied such as size and distortion in order to increase the success rate. Finally, the mapping between the RGB and depth pixels is performed to derive the real and accurate position coordinates of the landmark related to the 3D camera.

Two kinds of color representation filters are used to remove the background and retain the required color range for the button which are RGB and HSL filters. The working environment has different lighting and sunlight conditions which may easily affect the detection process with RGB color system. Thus, the success rate may be reduced significantly. This problem has been solved by using HSL color representation which is more stable against dynamic lighting conditions. The flowchart of button detection is shown in Figure 14.

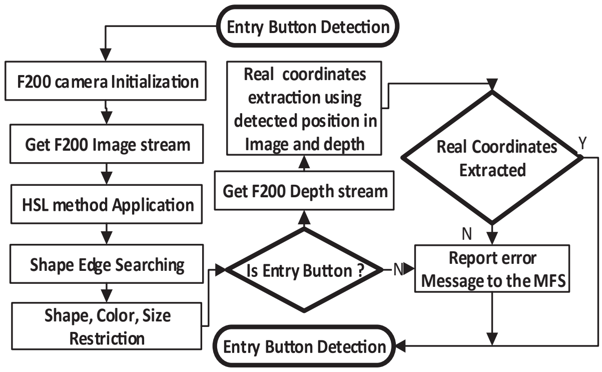


Fig. 14. HSL-based entry button detection

An experiment was performed to examine the performance of the F200 vision sensor with the HSL filter method for entry button detection. The F200 camera was installed on a camera stand in front of the external button as shown in Figure 15.a. The experiment was repeated 100 times for each distance (13, 20, 30, and

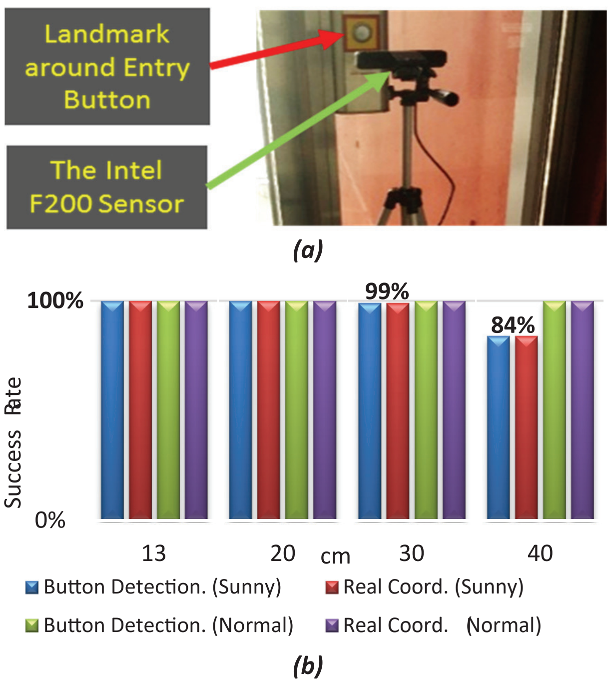


Fig. 15. Entry button detection based on F200

40 cm) and under different lighting conditions. The experiment results are shown in the chart in Figure 15.b, and it is clear that the entry button detection based on the F200 camera gives stable depth detection even at 13 cm under different lighting conditions, and entry button recognition was 100% successful in normal weather. Sunny weather reduced the success rate of the detection operation to 96% when the distance was 40 cm. The standard deviations (STDEV), means, and tolerance of entry button detection in sunny and cloudy weather are shown in Figure 15: Entry button detection based on F200.

Table 2 and Table 3, where tolerance can be defined as the possible limits of variation in the value of positional error that can be found by calculating the difference between the maximum and the minimum readings value. The calculated tolerance values cannot be improved since it depends on the F200 hardware. In these tables, the extracted entry button position referenced to the origin point of the utilized camera. Thus, the x and y values can be positive/negative due to entry button position from the camera.

Table 2. Entry button detection in sunny weather and at different distance

Distance to Entry Button	Cartesian Coordinates	Mean	STDEV	Tolerance	Unit
13 cm	X	-3.955	0.004	± 0.014	cm
	Y	0.869	0.014	± 0.014	cm
	Z	13.100	0.00	0	cm
20 cm	X	-3.665	0.008	± 0.0215	cm
	Y	1.290	0.011	± 0.021	cm
	Z	20.000	0.00	0	cm
30 cm	X	-4.434	0.025	± 0.0315	cm
	Y	1.775	0.057	± 0.066	cm
	Z	29.978	0.042	± 0.05	cm
40 cm	X	-3.337	0.037	± 0.046	cm
	Y	2.156	0.003	± 0.003	cm
	Z	39.937	0.049	± 0.05	cm

Table 3. Entry button detection in cloudy weather at different distances

Distance to Entry Button	Cartesian Coordinates	Mean	STDEV	Tolerance	Unit
13 cm	X	0.870	0.019	± 0.0305	cm
	Y	0.632	0.014	± 0.016	cm
	Z	12.974	0.044	± 0.05	cm
20 cm	X	1.074	0.024	± 0.063	cm
	Y	1.072	0.018	± 0.021	cm
	Z	20.000	0.00	0	cm
30 cm	X	0.515	0.034	± 0.0945	cm
	Y	1.500	0.011	± 0.066	cm
	Z	30.101	0.010	± 0.05	cm
40 cm	X	-1.107	0.00	± 0.0015	cm
	Y	1.901	0.001	± 0.002	cm
	Z	39.798	0.014	± 0.05	cm

Another possibility to avoid the sunlight effects on color detection process are local features matching algorithms [30]–[32]. These algorithms are somehow independent to the changes in scale, illumination, and orientation. The speeded-up robust features (SURF) algorithm can be considered as an efficient object recognition method with a fast scale- and rotation-invariant detector and descriptor [33]. The process starts with an offline step by capturing an image of the target to be saved in the database as a matching reference. Then, the matching process is performed by extracting local features from the reference image to be identified with the current image. A specific textured mark has been fixed on the panel of the entry button. This mark is considered as a reference to localize the button for pressing operation. The reference mark is recognized and localized using SURF algorithm with Intel F200 camera. Then, this information is used to guide the finger to the target and press the button as shown in Figure 16.

Fig. 16. SURF algorithm with F200 camera for button detection and localization

7.3. Elevator Opening Detection

Once the entry button has been pressed, the robot has to enter the elevator. The mobile robot has to monitor the status of the entrance door and checks if there is free space in the elevator when the door has opened. The H20 mobile robot is equipped with many sensor modules, including IR distance approximation and ultrasonic detection. Since the elevator door consists of a metal frame with glass panels, the IR sensor module cannot be used since the IR beam does not reflected from glass surfaces. Therefore, the ultrasonic distance sensor has been chosen as a range finder sensor to detect the door's status. A new method was established for checking the elevator door status based on the ultrasonic distance sensor. The data from the ultrasonic distance sensor determines whether the elevator door is opened or closed, and in addition detects whether the elevator has a free space or not.

8. Destination Floor Transition

It is important for the mobile robot to localize itself inside the elevator to reach the control panel. Two method can be used for inside elevator localization. The first is based on the ceiling light as a natural landmark as shown in Figure 17.a. This landmark has unique features and it can be extracted easily using the vision sensor. However, using the ceiling light as a landmark is not secure since it may break at any time. The second method is based on an artificial passive landmark and SGM to localize the robot inside the elevator as shown in Figure 17.b. After performing many experiments, localization inside the elevator based on artificial landmarks was adopted due to its stability. Next, the buttons on the control panel with the destination floor have to be detected to read the current floor and to leave the elevator when the door is opened. The movement strategy inside the elevator, internal button detection, and current floor reading methods are explained in detail in this section.

(a) *(b)*

Fig. 17. a) Ceiling light natural landmark and b) artificial landmark inside the elevator

8.1. Robot Movement Inside Elevator

The arm's limited workspace and the small size of the cabin add further challenges for the robot movement inside the elevator. To overcome these difficulties, the movement core utilizes the installed landmark with a correction procedure to reach a predefined position accurately. Then, the robot rotates towards the control panel. Finally, the ultrasonic range finder sensor with the motor encoder is employed to reach the exact position in front of the panel to conform to the arm's limited workspace. These procedures (clarified in Figure 18) are the basis for destination floor button pressing operation. After completion, the movement core returns the robot back towards the starting position so that it can leave the elevator safely when the destination floor is reached.

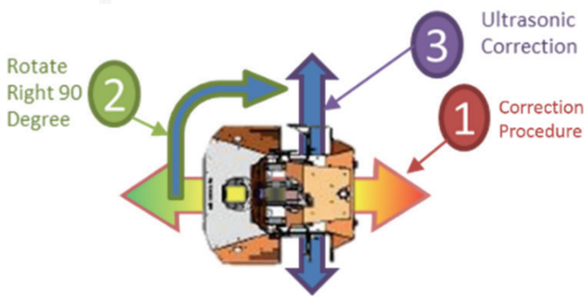


Fig. 18. Robot behavior inside the elevator

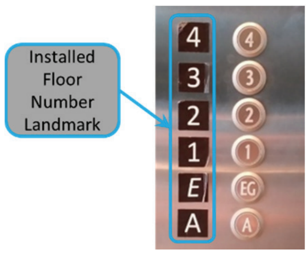


Fig. 19. Buttons inside the elevator

8.2. Internal Button Detection

As a first stage to detect the destination floor button, the movement core is used to specify the destination floor. At this level, the movement core depends on the transportation task status (Grasp Position Done, Place Position Done, and Charge Position Done) to determine the current destination. When the robot enters the elevator, the movement core checks the current destination floor based on the current intermediate goal, as clarified in Table 4. For example, if the grasping operation has been completed, the placing operation floor has to be requested. Then the elevator is directed to go to the required destination floor.

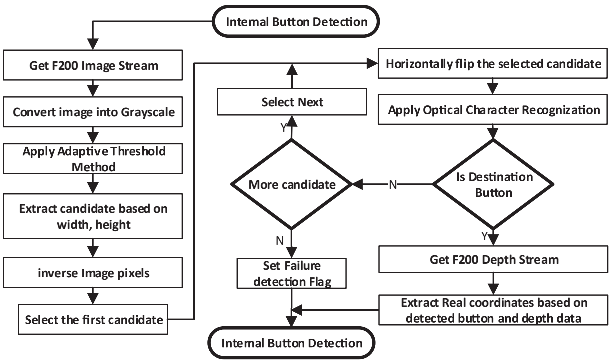


Fig. 20. Internal button detection flowchart

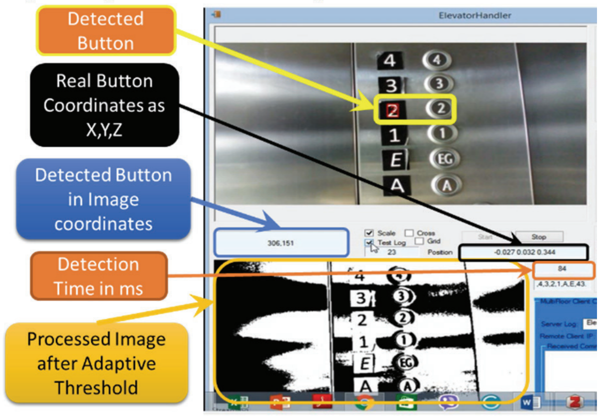
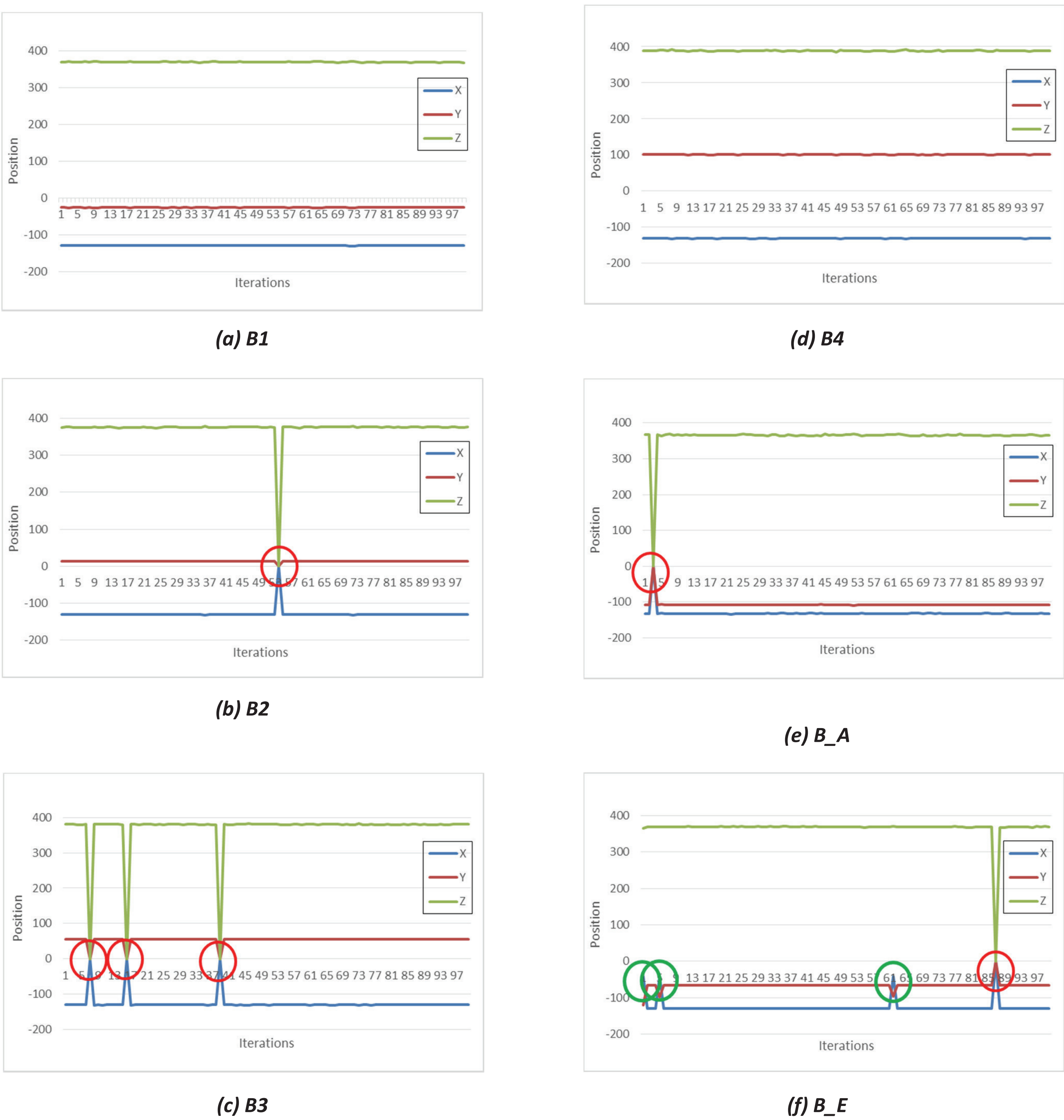


Fig. 21. Elevator handler GUI and Internal button operation



(a) B1

(d) B4

(b) B2

(e) B_A

(c) B3

(f) B_E

Fig. 22. Internal button detection stability

Table 4. Destination floor selection strategy

Grasp Station Done	Place Station Done	Charge Station Done	Entry Button	Internal Button
not yet	not yet	not yet	Current Floor	Grasp Station Floor
Done	not yet	not yet	Current Floor	Place Station Floor
Done	Done	not yet	Current Floor	Charge Station Floor

At this level of the elevator handling system, the Optical Character Recognition (OCR) is embedded with a new method to find the internal buttons (see Figure 19). In this method, it is important to recognize each button label separately, and thus the developed method applies a combination of filters. These include grayscale conversion, which makes the captured image suitable for subsequent stages, stretch contrast to improve the contrast in the image by stretching the domain of intensity, and an adaptive threshold to choose the best threshold under different light conditions for binary image conversion. Next, the search among each button candidate uses specific width and height features and takes the inverse value of the pixel and flips the candidate images horizontally (as the F200 image stream has a mirror image) to make them suitable for the OCR stage. Each extracted candidate passes to the OCR engine for comparison with the required destination and finally, based on the position of the matching candidate's the image and depth information, the real coordinates are extracted and translated into a robot arm reference.

Figure 20 demonstrates the internal button detection flowchart.

An experiment was conducted to verify the internal button detection method. The Intel RealSense F200 (RGB-D) sensor was positioned facing the internal control panel experiment was repeated a hundred times for each button. The F200 captured an image and the processed image with the related information (extracted coordinate, the real button coordinate, the detection time, etc...) are demonstrated in Figure 21. The extracted real button coordinates were measured based on the F200 as a reference position. Button real coordinates are reported as a chart for each button separately as clarified in Figure 22.

In in Figure 22, the red circles indicate the false negative cases while the green circles represent false positive cases. The false positive appeared when the system detects the original number on the button instead of the installed landmark. Therefore, the numbers on the buttons were covered and the detection experiments have been repeated. The new results of 600 experiments show that the false positive errors were removed and the success rate reaches to 99.5% as shown in Figure 23.

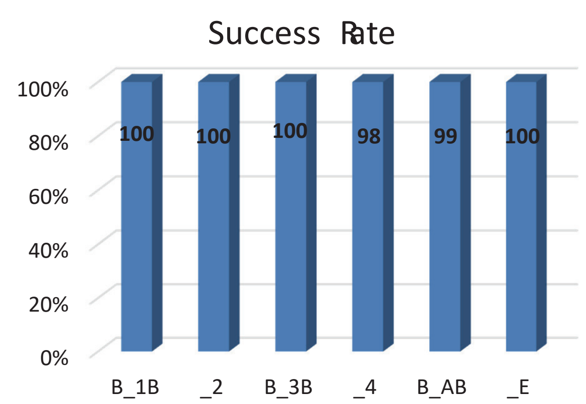


Fig. 23. Internal buttons detection success rate comparison

8.3. Current Floor Estimation

The floor estimation technique is applied to inform the robot about its current position inside the elevator. When the destination matches the current floor and the ultrasonic distance sensor recognizes the door status as opened, the robot can leave the elevator. As a first method, a computer vision method was used to read the current floor number indicator in the glassy elevator environment. As explained later in this section, this approach has many limitations and thus a current floor estimation method based on a height measurement system is utilized instead.

8.3.1. Floor Reader Based On Computer Vision

This system recognizes the current floor number which is installed in the elevator shaft for each floor based on computer vision as shown in Figure 24. As a first step, the image is converted to grayscale to make it suitable for the adaptive binary threshold filter. Secondly, the objects inside the processed binary image are collected to choose the best candidate based on size. Finally, the OCR engine is applied to the candidate object to recognize the floor number. When the destination matches the current floor number, the robot leaves the elevator. This approach has been validated in normal weather conditions and its functionality proved with a success rate reaching 99% [34]. However, this approach has many limitations. Since the floor number mark is installed over a glassy surface, the robot may not be able to read the number at some times of the day because of sunlight reflecting over the number mark. The F200 camera is installed in the robot's arm thus; the arm should be lifted all the

Fig. 24. Floor number installed on elevator door

time so it can read the number mark which consumes a lot of power. The robot must also detect the floor number before the door is opened, and in certain situations such as a human forming an obstacle between the robot camera and the floor number indicator the robot will fail to detect the floor number mark. Incorrect current floor estimation can make the robot lose its way to the destination. Thus, an innovative current floor estimation method based on a height measurement system is used instead.

8.3.2. Floor Estimation System Based On Height Measurement

As a hardware platform for the height measurement system, the LPS25HB pressure sensor and STM32L053 microcontroller were configured and programmed to detect the current floor position. Many challenges must be solved to use the pressure sensor as a floor estimation system which are as follows. Firstly, a soldering drift, which is defined as the difference between the accuracy of the sensor before and after soldering, appeared when the pressure sensor was attached to the STM32L053 microcontroller. A one-point calibration technique was used to solve the soldering drift problem by comparing the pressure sensor readings after attachment with a precision barometer. The difference was calculated and added to each pressure sensor reading. Secondly, the absolute digital barometer (pressure sensor) readings at the same floor of the building keep changing in a day due to various weather condition. The oscillation of the output signal and wide variations in readings during the week would reduce the utility of this technique for floor detection. Two methods were applied to deal with the variations in pressure. Firstly, a smoothing filter with a finite impulse response (FIR) structure was used to solve the problem of small variations in pressure sensor readings, and an adaptive calibration method was used to calibrate the sensor readings for the robot's current floor before entering the elevator in order to overcome the wide variation in daily pressure readings [19]a new system is presented to manage the elevator operations. A Wi-Fi socket is established to connect with the ADAM module for calling the elevator and requesting the destination floor. This technique does not provide any feedback on the elevator's door status or its current floor which in some situations can make the robot losing its way to the destination. Computer vision can be utilized to identify the current floor. In some special situation (human obstacle between the robot camera and the floor number indicator, difficult light conditions etc. This method proves its efficiency with 100% success rate.

9. Elevator Handling System and Robot Arm Kinematic Module Sockets

The EHS is a stand-alone program created to handle the elevator operations for multi-floor navigation. This system has the ability to detect the elevator environment through the following process. The exact position of the elevator entry button is first extracted. Then, the destination key is selected, and the current floor number is read. All this information is sent back over the socket to the MFS.

The system of button pressing operation has been realized using three main coding platforms which are the MFS, RAKM, and the button detection and localization using F200 camera. These 3 systems are connected with each other through an asynchronous socket. The 3 coding platforms exchange the orders and information between each other using 2 client-server communication models as shown in Figure 25. The MFS receives the X, Y, Z position information from the module of button detection and localization. Then, the MFS sends this information as an order to the RAKM to press the button. Afterward, the RAKM sends back the performance status to the MFS. Additionally, the RAKM socket is used also for transportation tasks to send a grasping and placing order. The RAKM informs the MFS when the requested operation (grasp, place, or press button) has been performed. The MFS system waits for a predefined time and then, if the RAKM fails to complete the task, it informs the RRC and stops the current transportation task to save time. Figure 26 shows the data flow between MFS, RAKM, and elevator handler.

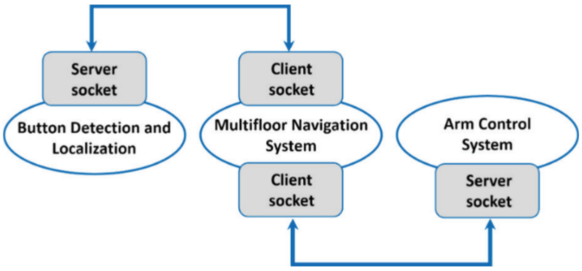


Fig. 25. Client-server models for button pressing operation

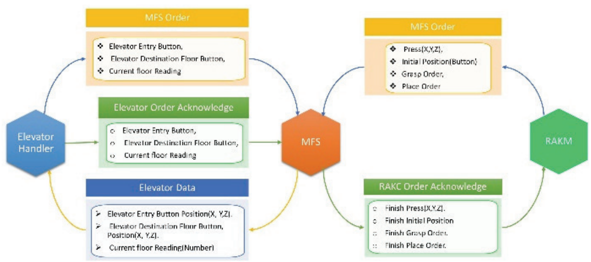


Fig. 26. Data flow between the MFS, elevator handler System, and RAKM

10. The Button Operation (Complete Scenario)

The entry and internal buttons detection and current floor reading methods have been developed as the main operations of elevator handling system. In the EHS, the F200 camera is utilized as an RGB-D sensor with HSL filtering for a stable color detection. The MFS with the EHS and the arm's kinematic module are developed to perform the detection and pressing operation for the required button. The button pressing operation procedures can be explained as follows. The movement core unit is used to reach the accurate elevator button detection position. The MFS is then sending an initial movement request via a socket to the arm's kinematic module. Afterward, the EHS

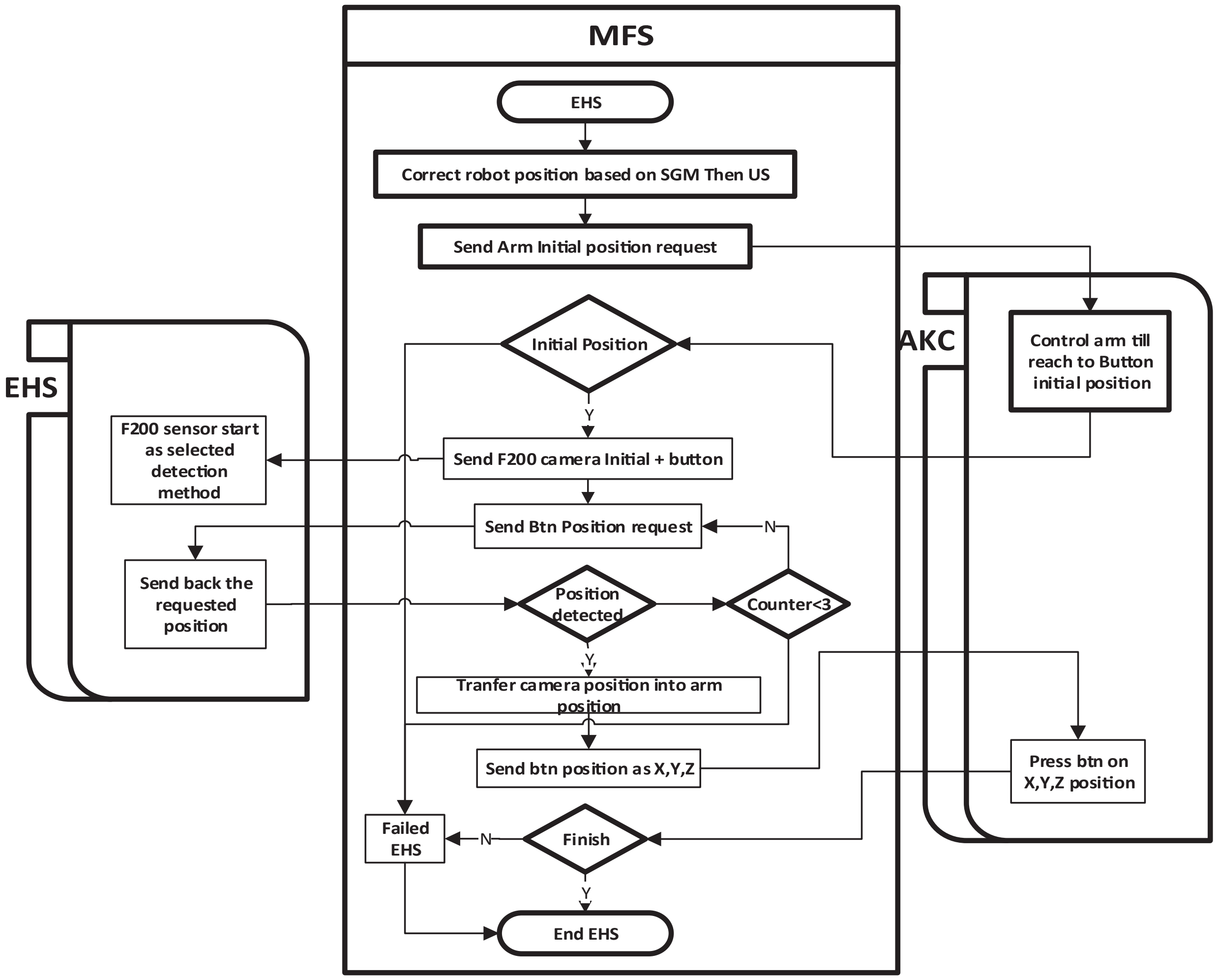


Fig. 27. Complete scenario

sends an initial request via a socket to the F200 camera after the arm reaches its initial position. The arm initial movement is essential to place the camera with arm finger in a close position in front of the required button. The destination button label is detected and its real coordinate's position is extracted. A checking process related to button detection is performed. The detection request is re-sent to the EHS if it fails to detect the button after 3 attempts. After the button detection step, its extracted position is sent back to the MFS. This position information belongs to the position of button label related to the camera. Therefore, two calibration steps are required. The first is to find the button position related to the camera according to the position difference between the button and its label. The second is to find the position of button related to elevator finger according to the position difference between the camera and finger.

Finally, a pressing order is sent with the required x, y, and z coordinate to the kinematic module to move the arm to the target. For pressing operation, the inverse kinematic solution is utilized to calculate the required angles of the arm joints based on the destination button position. The calculated joint angles are then sent via socket to the arm's servo motors. The flowchart of this scenario is shown in Figure 27. In the case of failure at any of these stages, the MFS rearranges the transportation task to complete it or to return to the charge station and informs the RRC level about the failure.

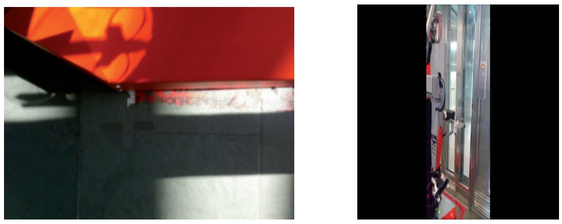
To press the required button, the arm should firstly grasp the elevator finger from the H20 holder and then moves to the initial position to be close to the button. The initial position of arm movement depends on the position of the button related to the arm shoulder. The elevator entry button and the internal buttons inside the elevator (see Figure 28) have a fixed height related to the arm shoulder. According to the height of each button and the range of error positioning in front of the button panel, the initial position of the finger can be estimated. This step is essential to provide a clear view for the 3D camera to detect the required button. Also, this step makes the finger close to the target to realize a small arm movement to press the button.

Fig. 28. Positions of buttons inside the elevator

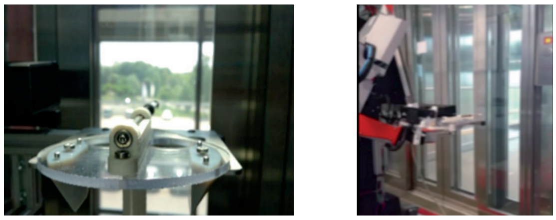
The process of entering the elevator has to be organized carefully by the robot. There is an IR transmitter and receiver on the both sides of the elevator door. After pressing the entry button, the elevator door keeps open for 15 seconds in case that nobody blocks the IR signals by entering the elevator. This time is quite enough for the robot to enter the elevator. In case that someone enters the elevator, the door keeps open for just 3 seconds. This time is very short which causes collision of the door with the H20 robot body. Another problem is related to the interference of IR signals of F200 camera with the IR signals of the elevator door. This causes the same effect of blocking the IR signals when someone enters the elevator. To cope with this issue, the left arm of the H20 robot is configured to be in front of the body as shown in Figure 29. The left arm keeps blocking the elevator signals which in turn keeps the door open and avoids the collision of the door with the robot during its movement.

Fig. 29. Left arm configuration to keep the elevator door open

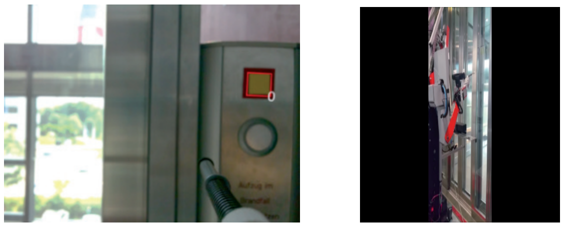
An experiment was performed to validate the elevator entry button detection with RAKM in a real transportation task. In this experiment, the robot speed was 0.2 m/s while the angular speed was 0.34 rad/s. The H20 mobile robot was employed for the transportation task. The MFS started to execute a multi-floor transportation task between the second and third floors at the celisca building. When the MFS reached the predefined elevator entry position (the position that allows the robot to enter the elevator directly after opening the door), the MFS requested that the EHS initialize the F200 camera and order the RAKM to control the robot arm. The RAKM starts its working process by grasping the finger as a new end effector for button pressing then controlling the arm to move to the initial detection position. After completing the entry button detection method and extracting the button's real x, y, and z coordinates, the RAKM will press the button. This experiment was repeated for ten times. Each time the communication sockets success to transport the MFS orders to the EHS and the RAKM and the EHS detect the entry position and send the x, y, and z coordinate to the MFS to add the calibration value then send it to RAKM for pressing operation. The robot succeeds to reach the elevator entry button position with repeatability range is ±1.25cm for x-axis and ±1cm for the y-axis. The steps of pressing scenario are shown in Figure 30: Complete operation scenario.



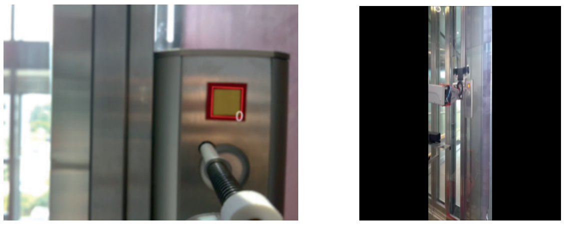
(a) Starting step (arm at rest configuration)



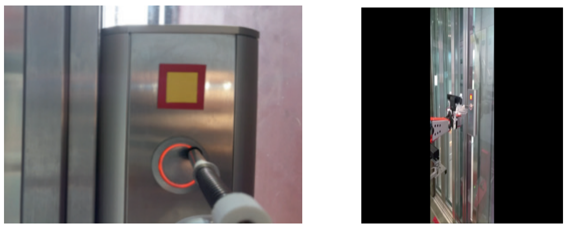
(b) RAKM pick the finger



(c) Finger attached to the robot arm



(d) Arm reach to initial position and button is detected



(e) Entry button is pressed

(f) Elevator door is opened

Fig. 30. Complete operation scenario

In the pressing operation of entry button, the robotic arm moves firstly from the rest configuration to the manipulation point of the elevator finger to grasp it. The elevator finger is placed on the robot holder. Then the arm moves with the finger to the initial position according to the required button. At the rest configuration, the position of the end effector related to the arm shoulder is (X= 0.566m, Y= 0 m, Z= 0m). This position information is according to the shoulder coordinates which is shown in Figure 8 where X represents the arm length in the rest configuration. Figure 31 shows a chart for the changes in the end effector position during the arm movement from the rest configuration to the initial position for entry button. In this example, the position of finger manipulation point related to the arm shoulder is (X= 0.312m, Y= 0.302m, Z= 0.005m). On the other hand, the initial position of end effector for entry button related to the arm shoulder is (X= 0.00m, Y= 0.38m, Z= 0.20m). The approximate time required to reach the manipulation point of the finger is about 16 seconds. But the complete time required to reach the initial position of entry button is 34 seconds. Figure 32 shows the path of the end effector in XY plane for the same example. Furthermore, Table 5 shows the changes of arm joints' values in degrees from the rest configuration to the finger grasping point and then to the initial position. full required time for pressing the entry button is about 43 seconds. The work has been developed using Microsoft Visual Studio 2015 with C# programming language. The project is running on a Windows 10 platform in the H20 tablet.

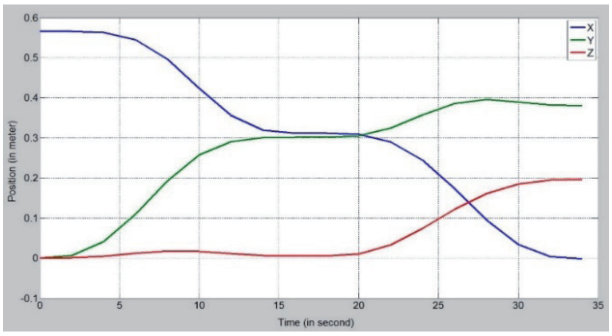


Fig. 31. End effector position versus time

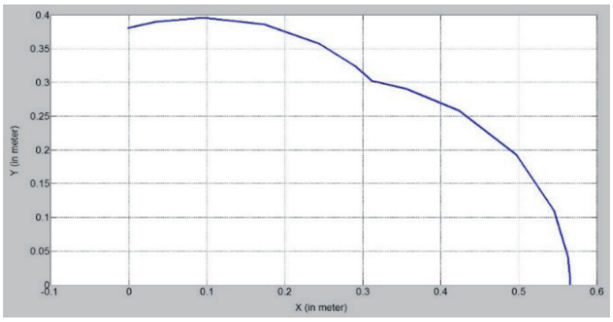


Fig. 32. End effector path in XY plane

Table 5. Joints values in degrees for button pressing

Configuration	J1	J2	J3	J4	J5	J6
Rest	0°	-90°	-90°	0°	90°	0°
Finger grasping	-3°	-92°	-91°	-75°	2°	-18°
Initial position	37°	-146°	-98°	-85°	57°	27°

11. Conclusion

A new approach to handle the glassy elevator operations based mobile robot is presented to enable multi-floor transportation in life science laboratories. Passive landmarks with stargazer sensor model are utilized to localize the robot in front of the elevator. The Intel RealSense F200 vision sensor is used for entry and internal buttons detection and localization. This sensor is fixed on the robot arm to reduce the effect of the sunlight on the entry button detection and to compensate the weakness of the H20 arm joints. A landmark has been installed around the entry button with a specific shape and color so as to enable an easy and applicable recognition process. The Optical Character Recognition (OCR) algorithm is used to recognize the numbers of elevator internal buttons. The LPS25HB pressure sensor and STM32L053 microcontroller were configured and programmed to work as a hardware platform for robust current floor estimation approach. For buttons pressing operation, a kinematic solution is employed to control the arm joints movement and a special finger model is designed. An IEEE 802.11g communication network with server-client structure and TCP/IP command protocol is used to establish a reliable and extendable communication network. Three sets of the experiment have been performed to validate the presented systems. The experiment results approve that the presented elevator's operations handler system has an efficient performance, which meets the requirement of life science transportation task based on mobile robots.

ACKNOWLEDGMENTS

The authors would like to thank the Ministry of Higher Education and Scientific Research in Iraq for the scholarship provided by Mosul University (Ph.D. stipend A. A. Abdulla), the German Academic Exchange Service (Ph.D. stipend M. M. Ali) and the German Federal Ministry of Education and Research for the financial support (FKZ: 03Z1KN11, 03Z1KI1). We also wish to thank DrRobot Company (Canada) for the technical support of the H20 mobile robots, and Mr. Lars Woinar for his contributions in the 3D modelling and printing designs.

AUTHORS

Ali A. Abdulla* – Center for Life Science Automation (celisca), University of Rostock, Rostock 18119, Germany, 2- College of Engineering, University of Mosul, Mosul, Iraq.
E-mail: Ali.abdulla@celisca.de.

Mohammed M. Ali* – Center for Life Science Automation (celisca), University of Rostock, Rostock 18119, Germany.
E-mail: mohammed.myasar.ali@celisca.de.

Norbert Stoll – Institute of Automation, University of Rostock, Rostock 18119, Germany.
E-mail: Norbert.Stoll@uni-rostock.de.

Kerstin Thurow – Center for Life Science Automation (celisca), University of Rostock, Rostock 18119, Germany
E-mail: Kerstin.Thurow@celisca.de.

Underlined authors made equal contributions to this paper.

* Corresponding authors

REFERENCES

[1] M. Wojtczyk, A. Knoll, "Utilization of a mobile manipulator for automating the complete sample management in a biotech laboratory. A real world application for service Robotics." In: *6th International Symposium on Mechatronics and its Applications, 2009. ISMA '09*, 2009, 1–9. DOI: 10.1109/ISMA.2009.5164800.

[2] H. Liu, N. Stoll, S. Junginger, K. Thurow, "Mobile robotic transportation in laboratory automation: Multi-robot control, robot-door integration and robot-human interaction." In: *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2014, 1033–1038. DOI: 10.1109/ROBIO.2014.7090468.

[3] B. Siemiątkowska, B. Harasymowicz-Boggio, M. Wiśniowski, "The Application of Mobile Robots for Building Safety Control," *J. Autom. Mob. Robot. Intell. Syst.*, vol. 10, 2016, no. 02, 9–14. DOI: 10.14313/JAMRIS_2-2016/11.

[4] D. Troniak *et al.*, "Charlie Rides the Elevator – Integrating Vision, Navigation and Manipulation towards Multi-floor Robot Locomotion." In: *2013 International Conference on Computer and Robot Vision (CRV)*, 2013, 1–8. DOI: 10.1109/CRV.2013.12.

[5] J.-G. Kang, S.-Y. An, S. Oh, "Navigation strategy for the service robot in the elevator environment." In: *International Conference on Control,* Automation *and Systems, 2007. ICCAS '07*, 2007, 1092–1097.

[6] X. Yu, L. Dong, L. Li, K. E. Hoe, "Lift-button detection and recognition for service robot in buildings." In *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, 313–316. DOI: 10.1109/ICIP.2009.5413667.

[7] Chen L.-K., Hsiao M.-Y., "Control of service robot by integration of multiple intermittent sensors." In: *AMPT 2013*, Taipei; Taiwan, 2014, vol. 939, 609–614. DOI: 10.4028/www.scientific.net/AMR.939.609.

[8] K. Maneerat, C. Prommak, K. Kaemarungsi, "Floor estimation algorithm for wireless indoor multi-story positioning systems." In: *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2014, 1–5. DOI: 10.1109/ECTICon.2014.6839893.

[9] L. Chen, B. Sun, X. Chang, "MS5534B Pressure Sensor and Its Height Measurement Applications." In: *2011 International Conference on* Information *Technology, Computer Engineering and Management Sciences (ICM)*, 2011, vol. 1, 56–59. DOI: 10.1109/ICM.2011.276.

[10] C. Lee, M. Ziegler, "Geometric Approach in Solving Inverse Kinematics of PUMA Robots," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-20, no. 6, Nov. 1984,695–706. DOI: 10.1109/TAES.1984.310452.

[11] T. Ho, C.-G. Kang, S. Lee, "Efficient closed-form solution of inverse kinematics for a specific six-DOF arm," *Int. J. Control Autom. Syst.*, vol. 10, no. 3, 567–573, Jun. 2012. DOI: 10.1007/s12555-012-0313-9.

[12] C. Man, X. Fan, C. Li, Z. Zhao, "Kinematics Analysis Based on Screw Theory of a Humanoid Robot," *J. China Univ. Min. Technol.*, vol. 17, no. 1, 49–52, Mar. 2007.

[13] G. Tevatia, S. Schaal, "Inverse kinematics for humanoid robots." In: *Proceedings 2000 ICRA. Millennium* Conference*. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2000, vol. 1, 294–299. DOI: 0.1109/ROBOT.2000.844073.

[14] M. Mistry, J. Nakanishi, G. Cheng, S. Schaal, "Inverse kinematics with floating base and constraints for full body humanoid robot control." In: Humanoids *2008 – 8th IEEE-RAS International Conference on Humanoid Robots*, 2008, 22–27. DOI: 10.1109/ICHR.2008.4755926.

[15] L. Nie, Q. Huang, "Inverse kinematics for 6-DOF manipulator by the method of sequential retrieval." In: the Proceedings of the International Conference on Mechanical Engineering and Material Science, China, 2012, 255–258.

[16] X. Gu, S. Neubert, N. Stoll, K. Thurow, "Intelligent scheduling method for life science automation systems." In: *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2016, 156–161. DOI: 10.1109/MFI.2016.7849482.

[17] A. A. Abdulla, H. Liu, N. Stoll, K. Thurow, "A New Robust Method for Mobile Robot Multifloor Navigation in Distributed Life Science Laboratories," *J. Control Sci. Eng.*, vol. 2016, Jul. 2016. DOI: 10.1155/2016/3589395.

[18] M. M. Ali, H. Liu, N. Stoll, K. Thurow, "Kinematic Analysis of 6-DOF Arms for H20 Mobile Robots and Labware Manipulation for Transportation in Life Science Labs," *J. Autom. Mob. Robot. Intell. Syst.*, vol. 10, no. 4, 2016. DOI: 10.14313/JAMRIS_4-2016/30.

[19] A. A. Abdulla, H. Liu, N. Stoll, K. Thurow, "An automated elevator management and multi-floor estimation for indoor mobile robot transportation based on a pressure sensor." In: *2016 17th*

International Conference on Mechatronics – Mechatronika (ME), 2016, 1–7.

[20] H. Liu, N. Stoll, S. Junginger, K. Thurow, "Human face orientation recognition for intelligent mobile robot collision avoidance in laboratory environments using feature detection and LVQ neural networks." In: *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015, 2003–2007. DOI: 10.1109/ROBIO.2015.7419067.

[21] M. M. Ali, H. Liu, R. Stoll, K. Thurow, "Multiple Lab Ware Manipulation in Life Science Laboratories using Mobile Robots." In: *2016 17th International Conference on Mechatronics(ME2016)*, Prague, Czech Republic, 2016.

[22] M. M. Ali, H. Liu, N. Stoll, K. Thurow, "An identification and localization approach of different labware for mobile robot transportation in life science laboratories." In: *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2016, 000353–000358. DOI: 10.1109/CINTI.2016.7846432.

[23] M. Ghandour, H. Liu, N. Stoll, K. Thurow, "A Hybrid Collision Avoidance System for Indoor Mobile Robots based on Human-Robot Interaction." In: *2016 17th International Conference on Mechatronics(ME2016)*, Prague, Czech Republic, 2016.

[24] "Dr Robot Inc.: WiFi 802.11 robot, Network-based Robot, robotic, robot kit, humanoid robot, OEM solution." [Online]. Available: http://www.drrobot.com/products_H20.asp. [Accessed: 19-Jan-2015].

[25] X. Chu, T. Roddelkopf, H. Fleischer, N. Stoll, M. Klos, K. Thurow, "Flexible robot platform for sample preparation automation with a user-friendly interface," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2016, 2033–2038. DOI: 10.1109/ROBIO.2016.7866628.

[26] J. Denavit, R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Trans. ASME Journal,* 1955, vo. 22*,* 215–221.

[27] M. M. Ali, H. Liu, N. Stoll, K. Thurow, "Intelligent arm manipulation system in life science labs using H20 mobile robot and Kinect sensor." In: *2016 IEEE 8th International Conference on Intelligent Systems (IS)*, 2016, 382–387. DOI: 10.1109/IS.2016.7737449.

[28] M. A. Ali, H. A. Park, C. G. Lee, "Closed-form inverse kinematic joint solution for humanoid robots." In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, 704–709. DOI: 10.1109/IROS.2010.5649842.

[29] R. O'Flaherty *et al.*, "Kinematics and Inverse Kinematics for the Humanoid Robot HUBO2+," Georgia Institute of Technology, Technical Report, 2013.

[30] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, Nov. 2004, 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94.

[31] E. Rosten, R. Porter, T. Drummond, "Faster and Better: A Machine Learning Approach to Corner Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, Jan. 2010, 105–119. DOI: 10.1109/TPAMI.2008.275.

[32] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, "ORB: An efficient alternative to SIFT or SURF." In: *2011 International Conference on Computer Vision*, 2011, 2564–2571. DOI: 10.1109/ICCV.2011.6126544.

[33] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, 346–359, Jun. 2008.

[34] A. A. Abdulla, H. Liu, N. Stoll, K. Thurow, "A Robust Method for Elevator Operation in Semioutdoor Environment for Mobile Robot Transportation System in Life Science Laboratories." In: *International Conference on Intelligent Engineering Systems (INES), 2016 IEEE International*, Budapest, Hungary, 2016, 45–50.