

Chapter

Middleware Application, Suitable to Build an Automated and Connected Smart Manufacturing Environment

Muzaffar Rao and Thomas Newe

Abstract

The current manufacturing transformation is represented by using different terms like; Industry 4.0, smart manufacturing, Industrial Internet of Things (IIoTs), and the Model-Based enterprise. This transformation involves integrated and collaborative manufacturing systems. These manufacturing systems should meet the demands changing in real-time in the smart factory environment. Here, this manufacturing transformation is represented by the term 'Smart Manufacturing'. Smart manufacturing can optimize the manufacturing process using different technologies like IoT, Analytics, Manufacturing Intelligence, Cloud, Supplier Platforms, and Manufacturing Execution System (MES). In the cell-based manufacturing environment of the smart industry, the best way to transfer the goods between cells is through automation (mobile robots). That is why automation is the core of the smart industry i.e. industry 4.0. In a smart industrial environment, mobile-robots can safely operate with repeatability; also can take decisions based on detailed production sequences defined by Manufacturing Execution System (MES). This work focuses on the development of a middleware application using LabVIEW for mobile-robots, in a cell-based manufacturing environment. This application works as middleware to connect mobile robots with the MES system.

Keywords: MES, Robots, Cell-based manufacturing, Middleware, ROS

1. Introduction

Initially, the drive of automation was to upturn productivity and to decrease the cost associated with human operators. But now the automation also emphasizes to improve flexibility and quality in a manufacturing process [1]. Automation is an essential part of the Industry 4.0 strategy, which is fully reshaping the old manufacturing process and supporting the growth of the business. Industrial automation refers to a mechanism that combines hardware and software. Industrial automation is the use of control systems and information technologies to handle different types of machinery and processes in an industry to replace a human being. Robotic automation [2–4] is used in several areas of manufacturing industries. Robots can be used to perform tasks like assembly, welding, shipping, product packing, and handling raw materials. Many manufacturers are leveraging robotic automation

for many applications. Robotic automation offers manufacturers growing opportunities and to remain competitive. Currently available industrial robots are multi-functional, so a single robot can be used for several different tasks. Sharing of the same workspace for robots and humans is becoming necessary in a smart manufacturing environment. To achieve this, robots should have the capabilities to sense and communicate. Modern mobile-robots (AIVs) can move autonomously from one place to another to achieve defined goals. For two decades, mobile-robots are in continuous research, and the rapid advances in robotics, computer vision & artificial intelligence are resulting in robots that can potentially hear and see more precisely than humans. This advancement in mobile-robots demands their rapid deployment in an industrial environment. The manufacturing process is evolving and from the robotics aspect, the industries are moving from Autonomous Guided Vehicles (AGVs) to Autonomous Intelligent Vehicles (AIVs). The use of AIVs is also reshaping the production lines in a smart factory.

A production line is a configuration of a factory that consists of a sequence of manufacturing steps. The current ideology of manufacturing depends on the linear production line. This ideology works in case of high volume demand of identical goods. But, the linear production line is not the most efficient way in the case of a large volume of goods with several choices. Manufacturers are looking for a cell-based approach to offer products with different varieties. As the cell-based approach involves more complex production flows so for this, conveyor line production systems are not suitable [5]. In cell-based manufacturing, the best way to handle the goods shifting between cells is the use of mobile-robots. Intelligent mobile-robot that knows the environment in which it operates, and that can calculate the optimized route between different cells are the best to use in the above mentioned cell-based manufacturing system. The linear and cell-based production line concept in a factory that manufactures PCBs is shown in **Figure 1**.

This work is about the development of a middleware application using LabVIEW software to control/manage the mobile-robot. This middleware

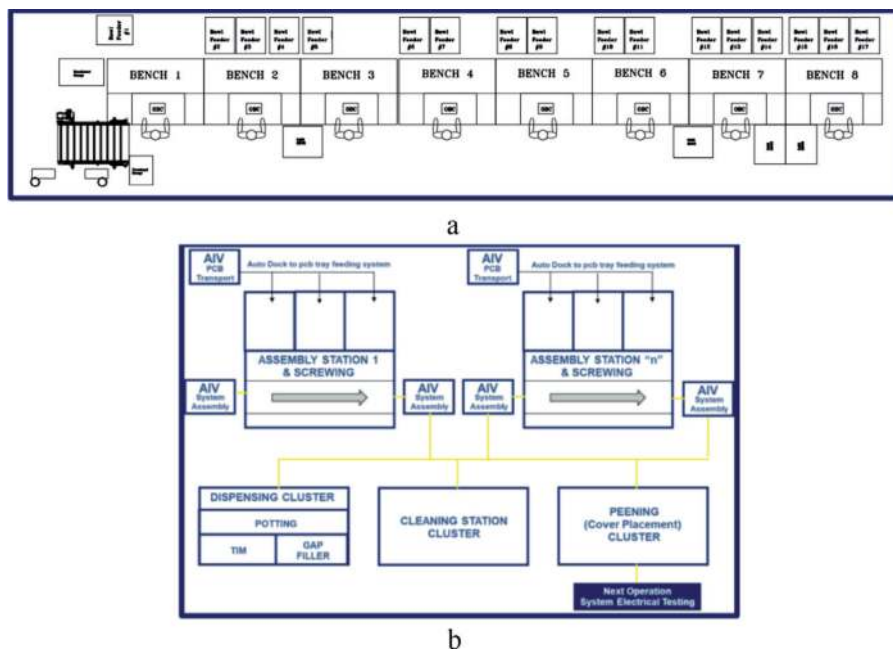


Figure 1. Linear vs. cell-based production line concept. (a) Traditional linear production line. (b) Cell-based production line.

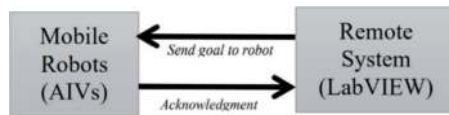


Figure 2.
Controlling of mobile-robots from a remote system using LabVIEW.

application is installed on the remote system mentioned in **Figure 2**. This work is part of an ongoing project, which involves the integration of fleets of mobile-robots and Manufacturing Execution System (MES) [6–8]. Here, the focus is on the development/testing of the LabVIEW application that can communicate with mobile-robots. The LabVIEW software is used by keeping in mind the future advancement of this application, which needs communication of fleets of robots with MES and this can be achieved using LabVIEW as middleware. The designed software application can be used with any ROS (Robot Operating System) based Robot. Here, turtlebot-3 [9, 10] and RB-1 (Robotnik) [11] are used for real-time testing.

This paper is organized as follows: Section 2 covers the industrial automation details and Section 3 discusses Mobile Robots (AIVs). Section 4 is about Robot Operating System (ROS) and Section 5 provides details of the target application. In Section 6 development of the LabVIEW application is given while Section 7 provides a discussion and Section 8 concludes.

2. Industrial automation

Manufacturers continuously face pressure to increase productivity and as mentioned in the introduction section, automation is the key to enable manufacturers to move closer to that goal in addition to meet the flexibility and quality requirements. There are numerous advantages of industrial automation, out of which some are given as [12]:

2.1 Quality control

Quality control is needed to build the customers' trust. A high level of quality compliance can be achieved by using robots in a manufacturing environment.

2.2 Repeatability

To be sure about consistency and the same quality end product is difficult without automation and this is achievable using repeatability. While, repeatability can be achieved using robots that are capable to perform the exact similar task in exactly the same way, repeatedly. This can help to save time as fewer errors result in less wasted time.

2.3 Waste reduction

The repeatability (mentioned above) allows manufacturers to minimize overall waste. Less error, as a result of repeatability, helps not only to save time, but it also minimizes the amount of material needed to yield the product. For example, robots can use less wire for welding, cut closer to the edge, and can use less amount of paint.

2.4 Faster cycle times

The production cycle time can be greatly improved using a robot. The more production helps to get the higher demand and ultimately brings more money.

2.5 Improved workplace

Safety Robots can be used in places which are not suitable for human. So, automation also helps to build safety environment.

2.6 Reduction of labor costs

Robots can replace jobs by removing workers from tough roles, which can help to reduce labor costs.

2.7 Reduced floor space

It's easy to start spreading out across the shop floor with extra materials, machinery, and tools. Robots can help to reduce the footprint of the required workspace by optimizing everything into a smaller, confined space.

2.8 Integration with business systems

Machinery and Robots can talk with each other to give a better view of the operating environment. This can help to make a smart decision on how to improve their process.

The next section describes mobile-robots with the focus on the robotic platform used here for testing.

3. Mobile-robots (AIVS)

Almost half a century ago, the first simple robot stepped onto the factory floor [13]. Today, a modern manufacturing business can be hardly imagined without the involvement of robots. The integration of robots with smart technologies allows the creation of more independent robotic systems that are not only able to carry out basic repetitive operations such as assembling, loading or modifying parts, but can also perform cognitive tasks, improving processes without human intervention and making instant decisions. In this work, turtlebot-3 and RB-1 (Robotnic) robots are used for the real-time testing of the developed LabVIEW application.

Turtlebot is a robot that is based on ROS Standard. There are three versions of turtlebot, turtlebot-1, turtlebot-2 and turtlebot-3 released in 2010, 2012, and 2017 respectively. There are two models of turtlebot-3 namely burger and waffle. Here, the burger model of turtlebot-3 is used, which is an affordable, small, and the ROS based mobile-robot. The core technology of the turtlebot-3 is SLAM (Simultaneous Localization and Mapping), Navigation, and Manipulation. Turtlebot-3 uses a Laser Distance Sensor (LDS-01), which is a 2D laser scanner capable of sensing 360 degrees that collects a set of data around the robot to use for SLAM. The turtlebot is connected/controlled through a remote system using wifi connection. The designed LabVIEW software is also tested with RB-1, which is used for real-time implementation in a factory environment. RB-1 is an autonomous and configurable robot, focused on the field of research in indoor applications. RB-1 is also based on ROS and uses an open architecture and modular control. The RB-1 platform comes with



Figure 3.
Turtlebot-3(burger) and RB-1(Robotnik).

a Hokuyo URG-04LX-UG01 [14], a 2D laser range finder for navigation. Turtlebot-3(burger) and RB-1 are shown in **Figure 3**.

The mobile-robots used in this work are based on Robot Operating System (ROS), which detail is given in the next section.

4. Robot operating system (ROS)

ROS is a meta-operating system for robots [15]. This is an open-source platform which functions are equivalent to what we expect from an operating system. These functions include low-level device control, hardware abstraction, message passing between processes, implementation of commonly-used functionality, and package management. The ROS meta operating system (shown in **Figure 4**) also provides libraries and tools for obtaining, building, writing, and running code across multiple computers. This meta-operating system is different than conventional operating systems in a way that it can be used for a different combination of hardware implementation.

However, unlike conventional operating systems, it can be used for numerous combinations of hardware implementation. Furthermore, it is a robot software platform that gives many development environments specialized for developing robot application programs. ROS runs on Ubuntu operating system; here we used Ubuntu 16.40 LTS. ROS supports many programming languages; here we used C++. There are different versions of ROS; here ROS Kinetic is used. In ROS, the smallest running unit of the processor is called a node. The first step is to run ROS_Master. Upon startup of the master, a node registers information such as name, message type, URI address, and port number of the node. The registered node can act as a publisher or subscriber based on the registered information, and nodes can exchange messages using topics and services. In this work, as mentioned earlier, the software application is developed using LabVIEW, which has support for ROS. The ROS for LabVIEW is a set of LabVIEW VIs that enables two-way communication



Figure 4.
Meta-operating system [15].

between ROS (running on Ubuntu machine) and LabVIEW (running on a windows machine). The LabVIEW ROS allows users to initialize nodes, publish and subscribe to various types of topics, and creates a ROS_Master within LabVIEW.

The next section mentioned target application details, which will help to understand the overall project concept.

5. Target application

This work is part of an ongoing project related to the integration of a fleet of mobile-robots and MES, targeted to design for the cell-based manufacturing environment. The target plant used traditionally linear manufacturing production lines for the assembly of its products. This approach takes a continuous flow in-line with balanced operations. The concern with this approach moving forward is the high up-front investment cost to achieve a Return On Investment (ROI). The line is designed for one product type, it is not flexible and is unsuitable for new product demand due to new product business or product revisions that may arise. The development of new smart manufacturing assembly lines i.e. cell-based will allow for flexibility to volume fluctuations, will support product ramp up/down, and the use of the latest automation technologies such as Industry 4.0 and the Internet of Things (IoT). The smart manufacturing assembly line will be used for multiple products, having the capability of a flexible production system for all production operations.

In cell-based manufacturing, different cells will be connected by an automatic product delivery system which will transport the product through the manufacturing process from process cell to process cell as defined by detailed production sequences using an MES. The automatic product delivery system will take the form of a fleet of AIVs with the ability to collect and deliver products to process cells in a format suitable for product feed. As mentioned earlier, this work focuses on one part of this target application i.e. development of a LabVIEW application that can communicate between the fleet of robots and MES. Details of this development are given in the next section.

6. Development of LabVIEW application

To prepare the ROS package for mobile-robots, ROS Kinetic is installed on Ubuntu 16.04 system and a code is prepared in C++ to publish and subscribe messages. This code declares a subscriber that subscribes to a topic (published by LabVIEW publisher) to receive commands for the robot and publishes a topic (subscribed by LabVIEW subscriber) to send an acknowledgment. Therefore, we can say that here publisher/subscriber sends/receives data to/from LabVIEW respectively. To take coordinates of physical location a map is generated using SLAM. Initially, the robot is taken around the workplace and allowed to scan the surrounding area with its main LiDAR sensor. It stitches that information together to form a complete static map of the workplace. The robot uses the map to calculate the best route between any two points. Here we are using hard coded coordinates of physical locations. The coordinates are taken using amcl node, which takes laser scans, transforms messages and outputs estimated pose.

The LabVIEW code consists of two while loops; one for publisher and the other for a subscriber. The ROS node is initialized using ROS_Topic_init subVI. The input of this subVI is connected with topic name, topic type, action (publisher or subscriber), and update rate and queue size. The node name (/LV1) is also

assigned using the 'node' terminal of ROS_Topic_init.vi. The ROS_Topic_init.vi is connected to ROS_Topic_Read.vi (in case of subscriber) and ROS_Topic_Write.vi (in case of publisher). The 'msg_in' input of ROS_TOPIC_Write.vi is connected with 'msg_out' of add_string.vi. The message which needs to publish is connected with the 'String' terminal of add_String.vi. While, in subscriber the 'Reply' terminal of ROS_TOPIC_Read.vi is connected with 'msg-in' of parse_string.vi. The string output of the parse_string.vi is connected as an indicator, which shows the robot response. The subscriber and publisher topics are closed using ROS_Topic_Close.vi. Multiple instances of the LabVIEW application are used to control more than one Mobile-Robots.

The developed software application is initially tested using turtlebot-3, as mentioned in Section 3. For this testing, two Linux systems and one windows system are used. Out of two Linux systems, one is used as remote-PC (Ubuntu 16.04) and the other is referring to turtlebot-PC (Raspbian). The ROS packages and all dependent packages are installed on remote PC and turtlebot-PC. All steps mentioned in [16] are followed to setup turtlebot, remote-PC, and turtlebot-PC. The Windows PC is used to run the LabVIEW. ROS communicate between systems using IP addresses; so, the turtlebot-PC, remote-PC, and windows-PC connected to the same wifi router. The .bashrc file is edited on turtlebot-PC and remote-PC in such a way that ROS_Master runs on remote-PC. The map is generated using SLAM node and then we noted coordinates of target physical locations. These coordinates are needed for ROS packages. Testing is started by running ROS_Master on remote-PC and then bring-up packages of turtlebot are executed. Navigation node runs as a next step (on remote-PC), which opens the rviz (visualization tool) with the selected map. The map shows turtlebot at some random point, so the estimated posture of robot is set in rviz. Upon running the ROS node, it keeps looking for a message from LabVIEW. The LabVIEW application pops-up a window to enter the Master_IP_address that is the address of remote-PC where ROS_Master is running. LabVIEW application sends some predefined strings that receive by subscriber running on remote-PC. Based on the received strings, target location coordinates are transferred to turtlebot. Upon reaching the target location, turtlebot sends an acknowledgment. This shows a successful communication between LabVIEW and mobile-robots. The same testing is done using RB-1, only the difference is that here we need only two systems (LabVIEW system and RB-1 system) as in this case ROS_Master runs on RB-1.

This work provides the foundation to develop a software application to control mobile- robots that can be used to integrate a fleet of mobile-robots to the MES system. Direct communication between MES and the mobile-robots is not possible that's why we introduced the middleware LabVIEW application and presented this idea in [17], which summary is given below.

Previously presented work [17] emphasizes on the integration of an MES and AIV that is needed to develop a completely automated, connected smart manufacturing environment. This integration requires a middleware application to execute the command translations between MES and AIVs. Here, a LabVIEW based application is built as the middleware. The middleware application development is divided into 03 major parts: (1) LabVIEW & MES communication, (2) LabVIEW & AIV communication, and (3) scheduler. In part (1), the middleware application imports relevant webservices of MES and produce corresponding LabVIEW VIs. The produced LabVIEW VIs are configured using the LabVIEW. NET functions. In part (2), the middleware application utilizes the 'ROS for LabVIEW' (LabVIEW add-on) to establish communication with an AIV. Part (3) comprises of a full cycle of scheduler operation, that consists of eight steps. These eight steps are: checking of pickup locations status, selection of one pickup location, pick the product, confirm

to the MES about completion of pickup operation, checking of drop-off locations status, selection of one drop-off location, drop the product, and confirm to the MES about the completion of the drop-off operation. The middleware application was tested in two stages, during the development phase, the Turtlebot-3 robot was used for testing and finally, the middleware application was commissioned using the Robotnik 'RB-1' robot in an actual factory environment. The developed middleware application supports 03 pickup and 03 drop-off locations. The pickup and drop-off locations are chosen based on pre-defined rules. The interface of the developed middleware application can be easily used by a non-technical operator and it displays a live log of operation.

The presented concept can be expended at a fleet level using the following:

(a) Communication between multiple instances of LabVIEW application with each robot on the factory floor – this is a decentralized approach and the problem with this approach is the risk of assigning the same tasks at the same time to more than one robot. But this approach is ideal when each robot is assigned a dedicated zone in a manufacturing environment (b) Running ROS_Master using LabVIEW application and control all robots in the field – The issue with this centralized approach is that there will be no communication with robots in the field in case if the system running LabVIEW stops working. This chapter focused on the case (a). The LabVIEW application will take the instructions from MES and send the command to robots in the field accordingly. The LabVIEW-MES communication is not the focus of this chapter.

7. Conclusion

This work is part of an on-going project which involves the integration of a fleet of mobile-robots (AIVs) with MES. This integration can be achieved using LabVIEW software as middleware because it can support both LabVIEW-to-Robots and LabVIEW-to-MES communication. This work focuses on the development/testing of LabVIEW-to-Robots communication. In the final setup, the LabVIEW system will work as a middleware between multiple Robots and MES. The designed LabVIEW application will be further developed to meet the overall requirements of fleet management. The updated version of the application will allow more complex production flows and the process of calling robot to pick the load; will be more intelligent by linking with MES in a fleet environment. This will provide an even larger boost to productivity, flexibility, and quality in a smart manufacturing environment.

Acknowledgements

This publication has emanated from research supported in part by a grant from Science Foundation Ireland under Grant numbers 13/RC/2094 and 16/RC/3918. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

Conflict of interest


The authors declare no conflict of interest.

Author details

Muzaffar Rao and Thomas Newe
Centre for Robotics and Intelligent Systems (CRIS), Department of Eelectronic and
Computer Engineering, University of Limerick, Ireland

*Address all correspondence to: muzaffar.rao@ul.ie

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] M. Wiktorsson, A. Granlund, M., Lundin, B. Södergren, "Automation and Flexibility: Exploring Contradictions in Manufacturing Operations" Conference: 23rd EurOMA conference, Trondheim, Norway
- [2] R. Szabó and A. Gontean, "Industrial robotic automation with Raspberry PI using image processing," 2016 International Conference on Applied Electronics (AE), Pilsen, 2016, pp. 265-268
- [3] S. Choi, W. J. Eakins and T. A. Fuhlbrigge, "Trends and opportunities for robotic automation of trim & final assembly in the automotive industry," 2010 IEEE International Conference on Automation Science and Engineering, Toronto, ON, 2010, pp. 124-129
- [4] B. Chu et al., "Robotic automation system for steel beam assembly in building construction," 2009 4th International Conference on Autonomous Robots and Agents, Wellington, 2009, pp. 38-43
- [5] B. Shishir, (2010). Changing from conveyor to work-cell-based systems to deal with fluctuations in demand. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 224(1), 169-176. URL: <https://doi.org/10.1243/09544054JEM1374>
- [6] Hennessey, T. MES is an Essential Part of Smart Manufacturing URL: <https://www.ibaset.com/blog/mes-is-an-essential-part-of-smartmanufacturing/> (online)
- [7] MESA International. MES Explained: A High Level Vision for Executives. Available online: <https://services.mesa.org/resourcelibrary/showresource/334444c5-388f-4360-beb4-3c86dc0f4de4>
- [8] SAP MES. Available online: <https://www.sap.com/uk/products/execution-mes.html>.
- [9] D. Singh, E. Trivedi, Y. Sharma and V. Niranjana, "TurtleBot: Design and Hardware Component Selection," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, Uttar Pradesh, India, 2018, pp. 805-809.
- [10] Turtlebot (online), URL: <https://www.turtlebot.com/>
- [11] Robotnik(RB-1) (online), URL: <https://www.robotnik.eu/manipulators/rb-one/>
- [12] RobotWorx, "Advantages of Industrial Automation with Robots" Robotic Articles(online), URL: <https://www.robots.com/articles/advantages-of-industrial-automationwith-robots>
- [13] SaM Solutions, "Five Reasons to Implement Robotics in Manufacturing" (online), URL: <https://www.samsolutions.com/blog/category/manufacturing/>
- [14] Hokuyo "Distance Data Output/URG-04LX-UG01" (online), URL: <https://www.hokuyo-aut.jp/search/single.php?serial=166>
- [15] Y. Pyo, H. Cho, R. Woon J., T. Lim, "ROS Robot Programming" First Edition Dec 22, 2017 Published by ROBOTIS Co.,Ltd ISBN 979-11-962307-1-5
- [16] Robotics e-manual Turtlebot-3 (online), URL: <http://manual.robotis.com/docs/en/platform/turtlebot3/overview/>
- [17] M. Rao; L. Lynch; J. Coady; D. Toal; T. Neue, "Integration of an MES and AIV Using a LabVIEW Middleware Scheduler Suitable for Use in Industry 4.0" Applications. *Appl. Sci.* 2020, 10, 7054. <https://doi.org/10.3390/app10207054>