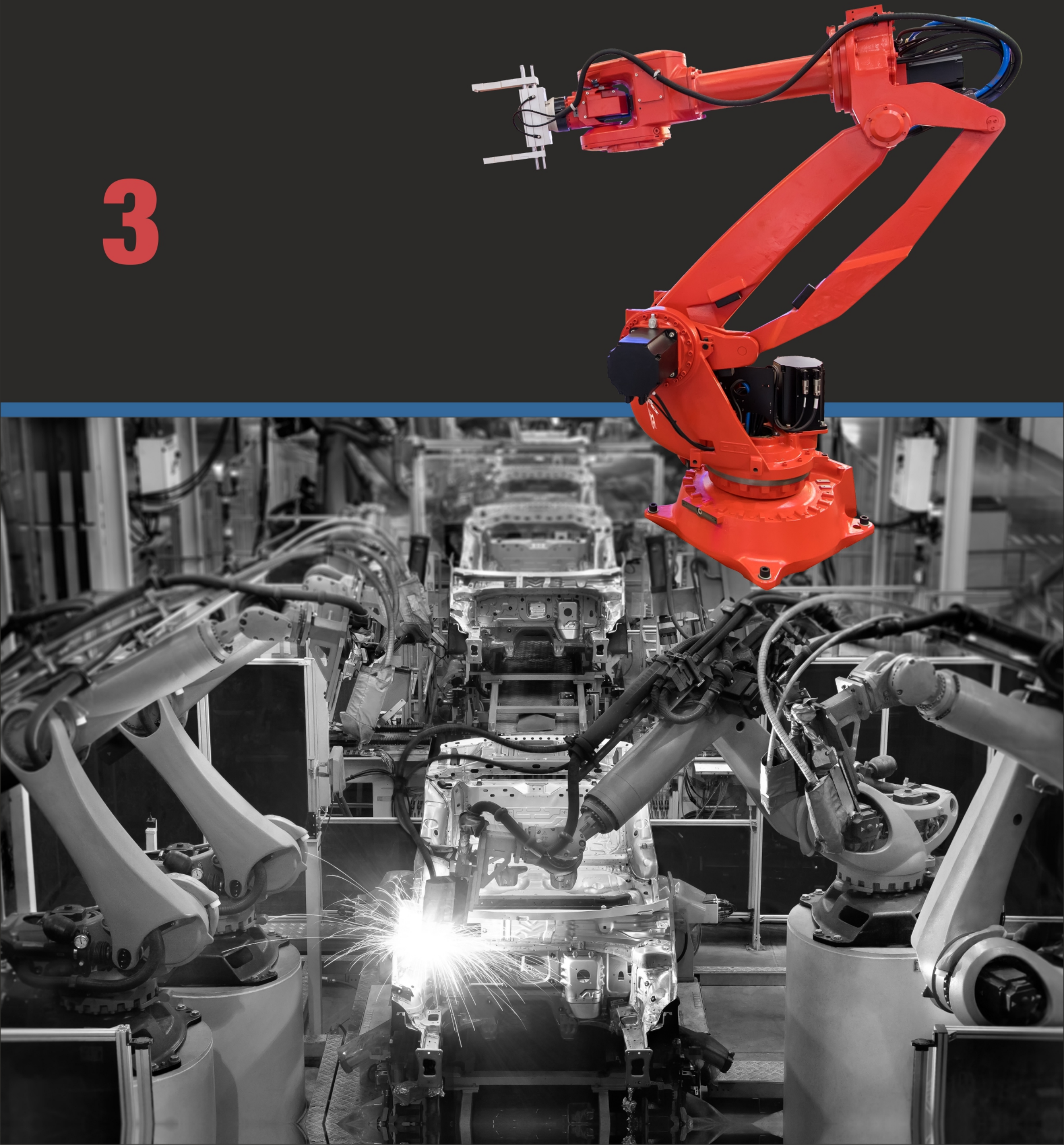
Sergey Y. Yurish, Editor

Advances in Robotics and Automatic Control

3



Advances in Robotics and Automatic Control

Book Series, Volume 3

S.Yurish
Editor

Advances in Robotics and Automatic Control

Book Series, Volume 3

IFSA International Frequency Sensor Association Publishing

S. Yurish, *Editor*
Advances in Robotics and Automatic Control, Book Series, Vol. 3

Published by IFSA Publishing, S. L., 2024

E-mail (for print book orders and customer service enquires):

ifsa.books@sensorsportal.com

Visit our Home Page on http://www.sensorsportal.com

Neither the authors no IFSA Publishing, S.L. accept any responsibility or liability for loss or damage occasioned to any person or property through using the material, instructions, methods or ideas contained herein, or acting or refraining from acting as a result of such use.

ISBN: 978-84-09-57872-6
BN-20240705-XX
BIC: TJFM1

Contents

Preface

It is a great pleasure to present the third volume of the open access Book Series ‘*Advances in Robotics and Automatic Control*’. Building upon the strong foundation laid by the first two volumes, published in 2018 and 2021, this edition continues to explore the cutting-edge developments and research in the rapidly evolving fields of robotics and automatic control.

The landscape of robotics and automatic control is characterized by relentless innovation and interdisciplinary integration. According to the recent market study, the global robotics market is expected to witness a significant growth in revenue, reaching a projected value of US $ 42.82 bn by the year 2024. This sector is expected to experience a steady annual growth rate, with a CAGR of 11.25 % from 2024 to 2028, and to reach US $ 65.59 bn by 2028. In its turn, the global industrial automation and control system market size is projected to reach around US$ 369.10 billion by 2032.

The third volume reflects this dynamic environment by featuring contributions from leading researchers and practitioners across various countries. Each chapter offers deep insights into both theoretical advancements and practical applications, underscoring the symbiotic relationship between academic research and industrial implementation.

The chapters in this volume cover a broad spectrum of topics, including but not limited to, advanced control algorithms, industrial autonomous vehicles, human-robot collaboration, sensor integration, AI applications in robotics, and emerging trends in automation. By presenting a diverse range of perspectives and methodologies, this volume aims to foster a comprehensive understanding of the current state and future directions of the field.

A notable highlight of this volume is the emphasis on real-world applications and case studies. These contributions demonstrate the tangible impact of research innovations on industries such as manufacturing, transportation, and beyond. The detailed case studies not only showcase the practical utility of theoretical models but also inspire further exploration and experimentation.

In compiling this volume, we have had the privilege of collaborating with a global community of experts whose passion and dedication drive the field forward. We extend our deepest gratitude to the authors for their invaluable contributions and to the reviewers for their meticulous evaluations and constructive feedback. Their collective efforts have ensured that this volume maintains the high standards of excellence set by its predecessors.

As we present this third volume, we hope it serves as a vital resource for researchers, students, and professionals engaged in the field of robotics and automatic control. We envision it sparking new ideas, fostering collaborations, and ultimately contributing to the advancement of technology that will shape our future.

We invite you to delve into the pages of this volume with curiosity and enthusiasm, confident that you will find it both informative and inspiring.

Sergey Y. Yurish

Book Series Editor Barcelona, Spain

Contributors

J.-M. Bonnin
IMT Atlantique, IRISA, ECAM Rennes, Louis de Broglie, Campus de Ker Lann, Bruz, Rennes 35091, France

Javier Bracamonte
School of Engineering, University of Applied Sciences and Arts Western Switzerland, HES-SO/HE-ARC, 2000 Neuchâtel, Switzerland

C. Couturier
IMT Atlantique, IRISA, ECAM Rennes, Louis de Broglie, Campus de Ker Lann, Bruz, Rennes 35091, France

M. Djoko-Kouam
IMT Atlantique, IRISA, ECAM Rennes, Louis de Broglie, Campus de Ker Lann, Bruz, Rennes 35091, France

A.-J. Fougères
IMT Atlantique, IRISA, ECAM Rennes, Louis de Broglie, Campus de Ker Lann, Bruz, Rennes 35091, France

J. Grosset
IMT Atlantique, IRISA, ECAM Rennes, Louis de Broglie, Campus de Ker Lann, Bruz, Rennes 35091, France

Loïck Jeanneret
School of Engineering, University of Applied Sciences and Arts Western Switzerland, HES-SO/HE-ARC, 2000 Neuchâtel, Switzerland

P. Madhavasarma
SASTRA University, Thanjavur 604301,Tamilnadu, India

Nabil Ouerhani
School of Engineering, University of Applied Sciences and Arts Western Switzerland, HES-SO/HE-ARC, 2000 Neuchâtel, Switzerland

Aïcha Rizzotti-Kaddouri
School of Engineering, University of Applied Sciences and Arts Western Switzerland, HES-SO/HE-ARC, 2000 Neuchâtel, Switzerland

M. Sridevi
Rajalakshmi Engineering College, 60010 Chennai, Tamil Nadu, India

Brendan Studer
School of Engineering, University of Applied Sciences and Arts Western Switzerland, HES-SO/HE-ARC, 2000 Neuchâtel, Switzerland

P. Veeraragavan
Anna University Tindivanam campus 604001, Tamilnadu, India

Takayoshi Yokota
Tokyo Information Design Professional University, Japan

Chapter 1

Collision and Obstacle Avoidance for Industrial Autonomous Vehicles – Simulation and Experimentation Based on a Cooperative Approach

J. Grosset, A.-J. Fougères, M. Djoko-Kouam, C. Couturier and J.-M. Bonnin

1.1. Introduction

One of the challenges of Industry 4.0, is to determine and optimize the flow of data, products and materials in manufacturing companies. To realize these challenges, many solutions have been defined [1] such as the utilization of automated guided vehicles (AGVs). However, being guided is a handicap for these vehicles to fully meet the requirements of Industry 4.0 in terms of adaptability and flexibility: the autonomy of vehicles cannot be reduced to predetermined trajectories. Therefore, it is necessary to develop their autonomy. This will be possible by designing new generations of industrial autonomous vehicles (IAVs), in the form of intelligent and cooperative autonomous mobile robots.

In the field of road transport, research is very active to make the car autonomous. Many algorithms, solving problematic traffic situations similar to those that can occur in an industrial environment, can be transposed in the industrial field and therefore for IAVs. The technologies standardized in dedicated bodies (e.g., ETSI TC ITS), such as those concerning the exchange of messages between vehicles to

J. Grosset
IMT Atlantique, IRISA, ECAM Rennes, Louis de Broglie, Campus de Ker Lann, Bruz, Rennes 35091, France

increase their awareness or their ability to cooperate, can also be transposed to the industrial context.

The deployment of intelligent autonomous vehicle fleets raises several challenges: acceptability by employees, vehicle location, traffic fluidity, vehicle perception of changing environments (dynamic), vehicle-infrastructure cooperation, or vehicles heterogeneity. In this context, developing the autonomy of IAVs requires a relevant working method. The identification of reusable or adaptable algorithms to the various problems raised by the increase in the autonomy of IAVs is not sufficient, it is also necessary to be able to model, to simulate, to test and to experiment with the proposed solutions. Simulation is essential since it allows both to adapt and to validate the algorithms, but also to design and to prepare the experiments.

To improve the autonomy of a fleet, we consider the approach relying on a collective intelligence to make the behaviours of vehicles adaptive. In this chapter, we will focus on a class of problems faced by IAVs related to collision and obstacle avoidance. Among these problems, we are particularly interested when two vehicles need to cross an intersection at the same time, known as a *deadlock situation*. But also, when obstacles are present in the aisles and need to be avoided by the vehicles safely.

This chapter is organized as follows: state of the art on algorithms and techniques to improve the autonomy of an IAV in Section 1.2. The improvement of a collision avoidance algorithm [2] in order to handle the problem of o is described in Section 1.3. In order to set up simulations and experiments, we propose an agent model which is presented in Section 1.4. The results related to the simulation are explained in Section 1.5. Section 1.6 describes the implementation of experiments with real robots. The results and a discussion of these experiments are initiated in Section 1.7. Finally, conclusions and future work are presented in Section 1.8.

1.2. State of the Art

In researches on Intelligent Transport System (ITS), autonomy of vehicles is well determined with 6 levels of autonomy [3]. However, no such scale exists in the industrial context, and too little research exists in this area [4, 5]. A few articles establishing a state of the art on the algorithms and techniques proposed to improve the control and relevance of the reactions of IAVs in the face of complex situations make it possible to verify the importance of this subject for industry 4.0 [4, 6]. The study of these articles shows that more and more proposed solutions relate to decentralized control algorithms [6-12]. Among the problems to be solved to make IAVs more autonomous, we can note: task allocation [13-16], localization and vehicles positioning estimation [17-22], path planning [23, 24], motion planning [23, 25, 26] with particularly centralized collision avoidance [27, 28] and decentralized collision avoidance [24, 29, 30], deadlock avoidance [31-33], and vehicle resources management (battery for instance) [4, 34, 35].

The objective of our research is to improve the IAV autonomy integrated in a fleet based on collective intelligent strategies. The capacity to exchange information between the different IAVs of a fleet is necessary to improve this autonomy [36, 37]. Thus, the collision avoidance problem can be solved by the cooperation between IAVs [2, 24, 29, 30]. The study [2] proposed a cooperation strategy based on the exchange of messages to determine the priority to pass an intersection between IAVs. The solution requires the vehicle to know its own position, and to be able to communicate with the other vehicles. The collision avoidance algorithm presented in [2] allows IAVs to communicate and cooperate using different types of messages. The communication between IAVs is done with 3 different types of messages:

- *Hello_msg*: message to indicate its presence with its position;
- *Coop_msg*: message before an intersection area to determine priority;
- *Ack_msg*: message to confirm receipt of a *Coop_msg*.

The European Institute of Telecommunications Standards (ETSI) has published a standard for this kind of *Cooperative Awareness Message* (CAM) (ETSI EN 302 637-2 standard [38, 39]) and *Decentralized Environmental Notification Message* (DENM) (ETSI EN 302 637-3 standard [40]). These specifications and messages are approved and constitute building blocks for the safety of future ITS [41]. The purpose of CAM messages is similar to *Hello_msg* in [2]. Under, the assumption that each vehicle is able to localize itself (e.g., using GNSS), they allow to exchange positions, and thus activate cooperative awareness. Indeed, it allows the cooperative vehicles in the surrounding to be positioned in real time. This is based on a strong assumption: the vehicles must be able to locate themselves precisely. Localization is generally done with GPS, which is not very precise. Moreover, GPS does not work inside buildings, and so in our Industry 4.0 context, GPS is not the tool that IAVs will be able to locate themselves with. DENM messages are alert messages. They are issued at the time of an unexpected event in order to cooperate, warn and disseminate information in the geographical area concerned.

ETSI has also published a standard for *Cooperative Perception Messages* (CPM) (ETSI TR 103 562 standard [42]). They allow vehicles to broadcast information about objects perceived in their detection area by their sensors to other vehicles such as obstacles, pedestrians or other vehicles. Another way to cooperate is to inform other vehicles of these intentions. In this regard, the ITS WG1 is currently working on the definition of a *Maneuver Coordination Service* (MCS) and its associated *Maneuver Coordination Messages* (MCM) [43]. The outcome of this work item is planned for end of 2023. We expect MCM messages could be used or enhanced to schedule the access to crossroads.

CAM, DENM and CPM messages are important messages standardized. Therefore, we will propose a model of these messages adapted to the industrial context. Then, we will show their possible use to cooperate and avoid collisions for the IAVs with the example of the Bahnes et al. algorithm [2]. Furthermore, we will discuss the MCM messages and their possible use from an experimental perspective in the results and discussion section.

1.3. Algorithm Improvement

The collision avoidance algorithm of [2] makes it possible to deal with the priority of different vehicles when approaching an intersection. However, it does not deal with the problems of detection, communication and avoidance of fixed or moving obstacles (e.g., human operators).

In order to address the problem of fixed and mobile obstacles in warehouse aisles, we extended the algorithm of Bahnes et al. to handle the presence of fixed or moving obstacles, in our prior research [44]. In this context, we proposed two new types of messages for collaborative perception:

- *Obstacle_msg*: a message sent by an IAV agent to other IAV agents circulating in the warehouse to indicate the presence of a perceived obstacle;

- *Alert_msg*: a message sent by an IAV agent to other IAV agents circulating in the warehouse to indicate an unavoidable obstacle.

Then, we simulate the algorithm staying within the framework of the three scenarios proposed by [2]. These simulations rely on an agent-based model where IAVs are agentified [45, 46]. Indeed, agent-based simulation for IAVs is the most common in the same way as simulations based on discrete events or robotics software [47]. IAV agents have the ability to exchange messages and are equipped with radar. This allows them to detect vehicles in front of them. For instance, given an IAV agent α_i, if another IAV agent α_j in front of it is stopped or is travelling at a slower speed, the IAV agent α_i can detect it with its radar and stop accordingly to avoid hitting it. To improve the collective autonomy of the IAVs it is essential that they have a good capacity for individual autonomy. The individual autonomy of the IAVs strengthens their collective autonomy.

The Bahnes algorithm proposed in our previous work [44] has been adapted with the standard ETSI messages used for ITS, in Fig. 1.1. These messages exchanged by the different IAVs remain consistent with Bahnes' algorithm and are adapted to the industrial context. The messages used that substitute the messages defined by Bahnes et al. are the CAM, MCM and ACK_MCM messages respectively to the *Hello_Msg*, *Coop_Msg* and *Ack_Msg* messages. Finally, the two types of message proposed for collaborative perception in our previous work [44], *Obstacle_msg* and *Alert_msg*, are replaced by CPM and DENM messages respectively.

The algorithm presented in Fig. 1.1 does not take into account the priority of passage in the case where 2 IAVs request passage at the same time at an intersection. This means that it does not propose a solution in case of a deadlock situation. We will discuss the different possibilities in the results and discussion section for our future work.

1.4. Modeling

In order to implement the Bahnes algorithm in simulation, we have modelled the different agents that will be implemented in the simulation. Fig. 1.2 shows the model agentified we have implemented in the simulation.

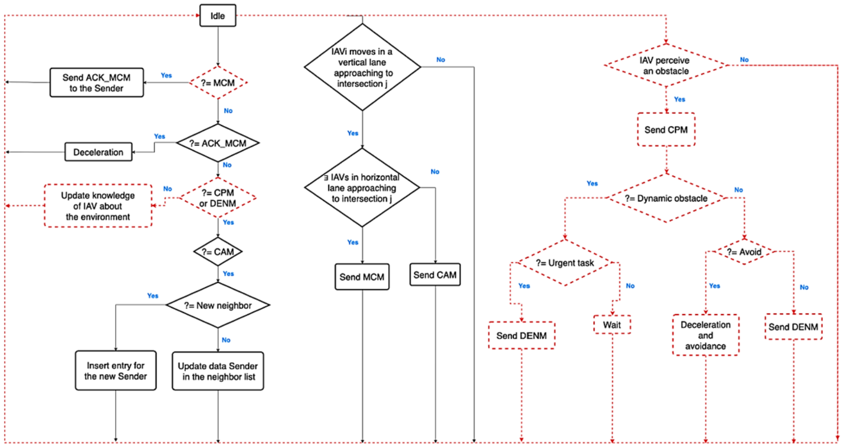

Idle
Send ACK_MCM to the Sender
Yes
?= MCM
No
Deceleration
Yes
?= ACK_MCM
No
Update knowledge of IAV about the environment
No
?= CPM or DENM
Yes
?= CAM
Yes
Yes
?= New neighbor
No
Insert entry for the new Sender
Update data Sender in the neighbor list
IAVi moves in a vertical lane approaching to intersection j
No
Yes
∃ IAVs in horizontal lane approaching to intersection j
No
Yes
Send MCM
Send CAM
IAV perceive an obstacle
No
Yes
Send CPM
Yes
?= Dynamic obstacle
No
?= Urgent task
Yes
No
Send DENM
Wait
?= Avoid
Yes
No
Deceleration and avoidance
Send DENM

Fig. 1.1. Improvement of Bahnes's algorithm to treat the problem of collision and obstacles.

IAVs are equipped with radar to detect pedestrians, other IAVs (dynamic obstacles) and goods (static obstacles) present in their activity area. Then, to move in their environment and accomplish their mission, they have knowledge about their environment through their own perception of the environment and through the information received from other IAVs. Indeed, they exchange information with other IAVs or even with the infrastructure thanks to different types of messages: CAM, DENM, CPM, MCM and ACK_MCM. This allows IAVs to build up their own dynamic mapping of the environment. Thus they are cooperative, pro-active and autonomous to carry out their missions without colliding with static and dynamic obstacles.

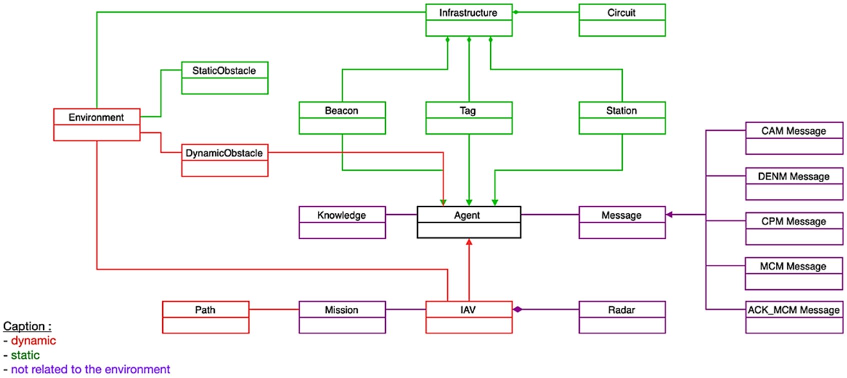
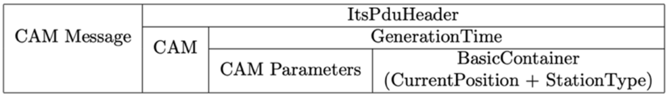


Fig. 1.2. Class diagram of the simulation.

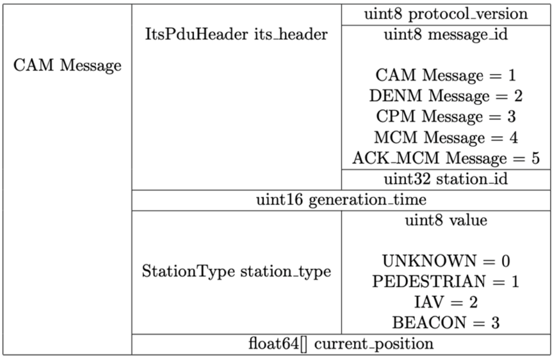
The different messages presented in the Fig. 1.2 are used in our cooperation protocol for simulation and experiments. Their modelling and representation are inspired by the messages standardized in the TC ITS (Intelligent Transportation System) of the European Institute of Telecommunications Standards (ETSI) standard messages. That means, we transpose the messages of the Bahnes algorithm to equivalent ETSI messages we adapted to the industrial context which has different constraints than the road sector.

The purpose of *Hello_msg* proposed by Bahnes correspond to the *Cooperative Awareness Message* (CAM, ETSI EN 302 637-2 standard [38, 39]). This is a message sent by the vehicle to indicate its position in real time. We propose a model for the industrial context in Fig. 1.3 and its associated implementation as a message in ROS2 in Fig. 1.4.



CAM Message	ItsPduHeader		
	CAM	GenerationTime	
		CAM Parameters	BasicContainer (CurrentPosition + StationType)

Fig. 1.3. Representation of CAM Message for the industrial context.



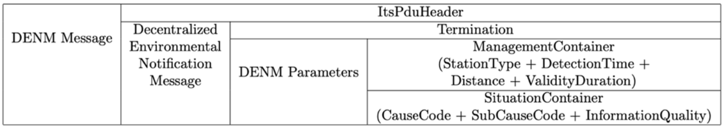
CAM Message	ItsPduHeader its_header	uint8 protocol_version
		uint8 message_id CAM Message = 1 DENM Message = 2 CPM Message = 3 MCM Message = 4 ACK_MCM Message = 5
		uint32 station_id
	uint16 generation_time	

	StationType station_type	uint8 value
		
		UNKNOWN = 0
		PEDESTRIAN = 1
		IAV = 2
		BEACON = 3

	float64[] current_position	

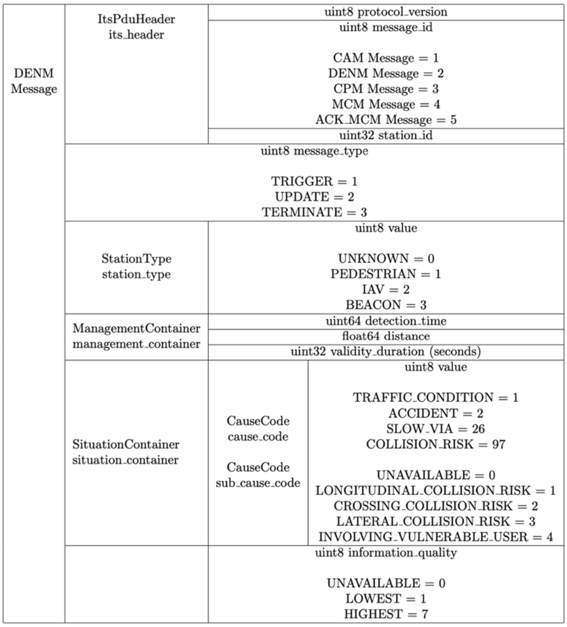
Fig. 1.4. Modeling of CAM Message in ROS2.

The alert message we proposed for the simulation of the augmented Bahnes algorithm will be implemented using the *Decentralized Event Notification Message* (DENM, ETSI EN 302 637-3 standard [40]). This can be of 3 different types: TRIGGER, UPDATE and TERMINATE (*message_type* in the modelling of the message in Fig. 1.6). TRIGGER corresponds to the first alert message, UPDATE to an alert message that allows to update information related to the first measurement, and finally the TERMINATE type that allows to inform others that the alert that has been issued is no longer active. This message contains the cause and sub-cause of the alert that is issued in the *SituationContainer* block of the message (see Fig. 1.5). Several alert message codes have been transposed from the standard for our experiments such as *Collision_RISK* with sub-causes associated with this code such as a longitudinal, lateral, intersection-related or vulnerable user collision risk (modelling of *SituationContainer* in Fig. 1.6).



DENM Message	ItsPduHeader		
	Decentralized Environmental Notification Message	Termination	
		DENM Parameters	ManagementContainer (StationType + DetectionTime + Distance + ValidityDuration)
			SituationContainer (CauseCode + SubCauseCode + InformationQuality)

Fig. 1.5. Representation of DENM Message for the industrial context.



DENM Message	ItsPduHeader its_header		uint8 protocol_version
			uint8 message_id CAM Message = 1 DENM Message = 2 CPM Message = 3 MCM Message = 4 ACK_MCM Message = 5
			uint32 station_id

	uint8 message_type
	TRIGGER = 1
	UPDATE = 2
	TERMINATE = 3		

	StationType
	station_type
			uint8 value
			UNKNOWN = 0
			PEDESTRIAN = 1
			IAV = 2
			BEACON = 3

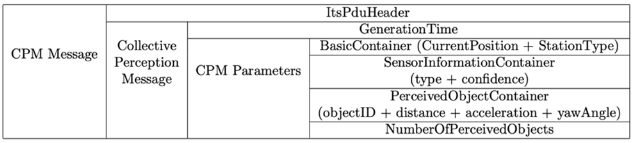
	ManagementContainer management_container		uint64 detection_time
			float64 distance
			uint32 validity_duration (seconds)

	SituationContainer
	situation_container
		CauseCode
		cause_code
		CauseCode
		sub_cause_code
			uint8 value
			TRAFFIC_CONDITION = 1
			ACCIDENT = 2
			SLOW_VIA = 26
			COLLISION_RISK = 97
			UNAVAILABLE = 0
			LONGITUDINAL_COLLISION_RISK = 1
			CROSSING_COLLISION_RISK = 2
			LATERAL_COLLISION_RISK = 3
			INVOLVING_VULNERABLE_USER = 4

			uint8 information_quality
			UNAVAILABLE = 0
			LOWEST = 1
			HIGHEST = 7

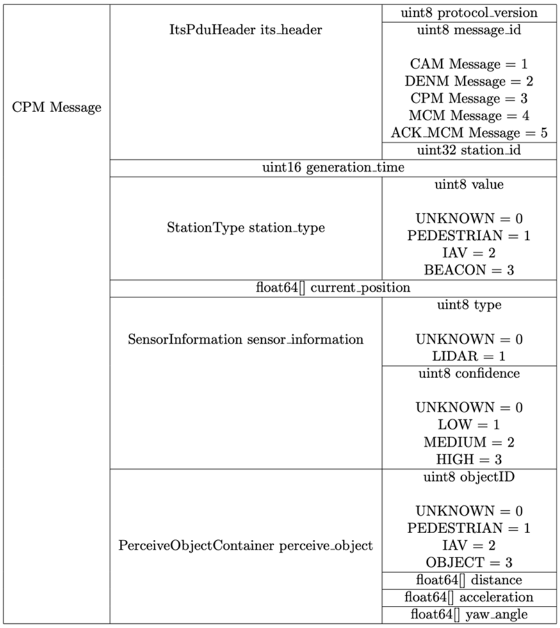
Fig. 1.6. Modeling of DENM Message in ROS2.

Bahnes' augmented algorithm allows vehicles to take obstacle detection into account. This augmentation has seen the arrival of a new message: *Obstacle_message* in our previous work [44]. ETSI has standardized a message for the ITS domain for the exchange of obstacle perception or other information called the *Cooperative Perception Message* (CPM, ETSI TR 103 562 [42]). Our CPM-inspired adaptation for the industrial context is shown in Fig. 1.7 and our ROS2 implementation is modeled in Fig. 1.8.



CPM Message	ItsPduHeader		
	Collective Perception Message	GenerationTime	
		CPM Parameters	BasicContainer (CurrentPosition + StationType)
			SensorInformationContainer (type + confidence)
			PerceivedObjectContainer (objectID + distance + acceleration + yawAngle)
			NumberOfPerceivedObjects

Fig. 1.7. Representation of CPM Message for the industrial context.



CPM Message	ItsPduHeader its_header	uint8 protocol_version
		uint8 message_id CAM Message = 1 DENM Message = 2 CPM Message = 3 MCM Message = 4 ACK_MCM Message = 5
		uint32 station_id
	uint16 generation_time	

	StationType station_type	uint8 value
		UNKNOWN = 0
		PEDESTRIAN = 1
		IAV = 2
		BEACON = 3

	float64[] current_position	

	SensorInformation sensor_information	uint8 type
		UNKNOWN = 0
		LIDAR = 1

		uint8 confidence
		UNKNOWN = 0
		LOW = 1
		MEDIUM = 2
		HIGH = 3

	PerceiveObjectContainer perceive_object	uint8 objectID
		UNKNOWN = 0
		PEDESTRIAN = 1
		IAV = 2
		OBJECT = 3

		float64[] distance
		float64[] acceleration
		float64[] yaw_angle

Fig. 1.8. Modeling of CPM Message in ROS2.

1.5. Simulation Results

Before moving on to real experiments, we simulate the algorithm proposed in Fig. 1.1 to verify that this cooperative solution is able to solve the collision and obstacle avoidance problems. This allowed us to validate our model proposed in the previous section, and to verify that other difficulties and problems do not appear before the real experimentation phase.

The simulation framework was taken from the Bahnes' algorithm. Therefore, our model presented in Fig. 1.2 was simulated with the different scenarios and the traffic plan proposed by Bahnes et al. [2] presented in Figs. 1.9, 1.10. Indeed, this choice is justified by our conviction to use it as a benchmark plan to compare results.

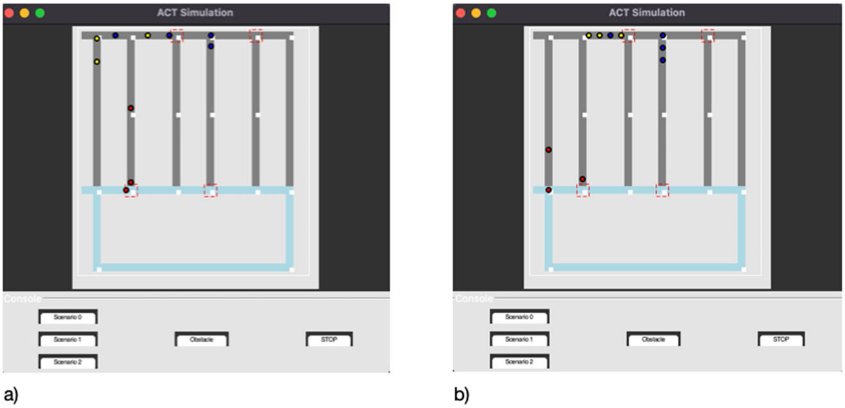


Fig. 1.9. Simulation of radar use: a) at the top of the picture: one blue and three yellow IAVs arrive near the intersection, b) while waiting for the yellow IAV to pass the intersection, the radar of the blue IAV and the two others yellow IAV allow them to stop and keep their distance to avoid colliding.

Ten IAVs start from the horizontal central aisle on the left. The red IAVs do the first circuit corresponding to the first loop at the top left, the blue ones follow them but will go down the 4th aisle (3rd loop). Finally, the yellow ones do the big loop going clockwise just like the red and yellow IAVs. The upper part of this traffic plan illustrates the possibility of checking the correct management of the intersection. In addition, our simulation allows us to place obstacles at random positions in the lanes. This makes it possible to check the correct reaction of the IAVs in relation to their perception of the obstacles and the logs of their message exchanges. It involves different intersections, where vehicles can arrive from different sides like in a warehouse (4 intersections are shown in Fig. 1.10). Thus, it provides the different characteristics of an industrial environment and allows us to realize simulated experimental tests in line with realistic scenarios.

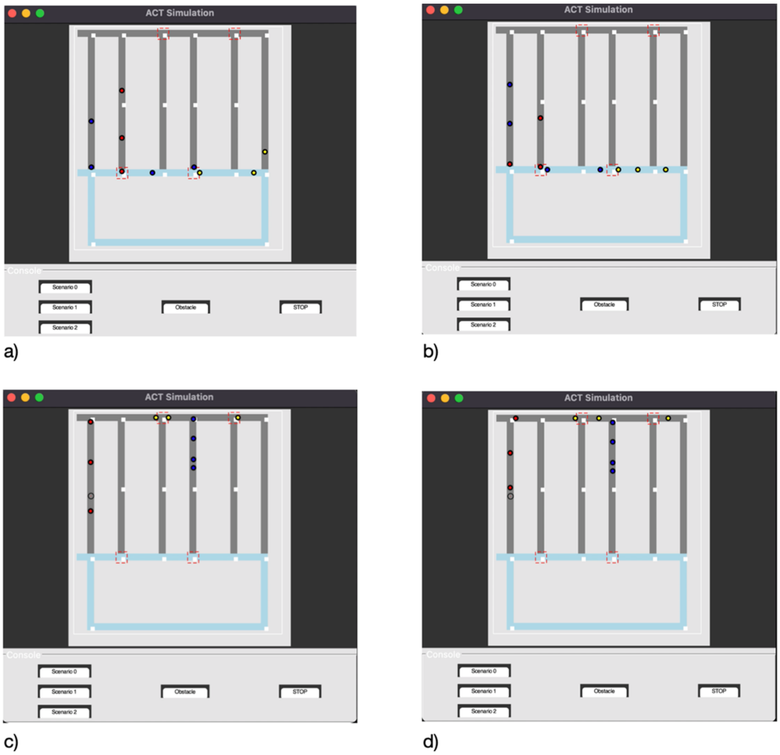


Fig. 1.10. Simulation of the scenarios: a) in the center of the picture: a blue and yellow IAV arrive at an intersection, b) the yellow IAV passed the intersection after communicating with other IAVs, c) on the left side of the picture: a red IAV perceives a fixed obstacle in front of him, d) a red IAV avoided the obstacle.

We notice in the simulation that the avoidance is well respected, and the obstacles are perceived by the IAVs. Therefore, the simulation validates the extended Bahnes's algorithm with collision avoidance and fixed or dynamic obstacle detection processing.

1.6. Experimentations

1.6.1. Context

After having carried out simulations, the objective is to move on to the stage of real experiments to test cooperation algorithms. As a first step, we want to verify that the augmented Bahnes algorithm presented in Fig. 1.1 does indeed allow industrial vehicles to avoid collisions with obstacles and to manage decision making at intersections. The main objective is to validate our hypothesis that collective intelligence strategies between vehicles will increase their individual and collective autonomy to perform their tasks efficiently.

In the industry 4.0 context, we use Turtlebot3 ‘burger’ robot as our representation of autonomous industrial vehicles. These robots are equipped with different components as described in Fig. 1.11 and use a *Raspberry Pi* and the *Robot Operating System* (ROS). Indeed, ROS is an open-source framework for the development of robotics applications and is the tool favored by researchers and even industrialists today. For our research we use ROS2 because it provides real-time control systems and large-scale distributed architectures [48]. Compared to ROS, there is no master entity, and ROS2 utilized Data Distribution Service (DDS). Thus, as we are working on distributed robotic systems to evaluate our cooperation algorithms, the choice focused on ROS2 naturally for our robot fleets. To facilitate real experiments, we first used the Gazebo simulation environment which is easily integrated with ROS2 and where Turtlebot3 can be simulated. This part of the simulation in the Gazebo environment makes it possible to check the correct operation of message exchanges and the cooperation algorithm and thus avoid collisions in experiments with physical robots. In this simulation environment, we first created a world representing an intersection where we deployed 4 Turtlebot3 ‘burger’ equipped with their LIDAR (see Fig. 1.4).

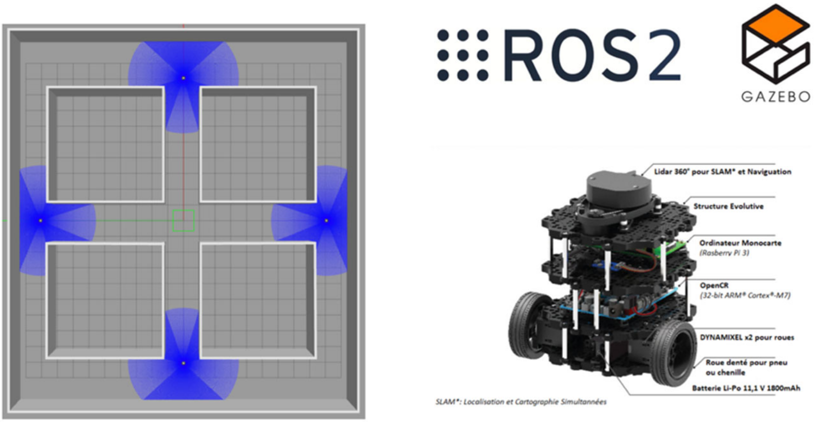

ROS2
GAZEBO
Lidar 360° pour SLAM* et Naviguation
Structure Evolutive
Ordinateur Monocarte
(Rasberry Pi 3)
OpenCR
(32-bit ARM® Cortex®-M7)
DYNAMIXEL x2 pour roues
Roue denté pour pneu ou chenille
Batterie Li-Po 11,1 V 1800mAh
SLAM*: Localisation et Cartographie Simultannées

Fig. 1.11. Simulation environment for experimentations.

1.6.2. Collective Cooperation Strategy

Before simulating the Bahnes algorithm or other cooperative algorithms with standardized messages with the ROS2 framework and Gazebo, we worked on the task assignment for autonomous vehicles to implement robot control and movement utility related to the notion of tasks/missions.

We have assumed that the tasks are known and therefore configured in a file with the destination of each robot for each task. The destinations where the robots must perform tasks are represented by *PositionAction* which is (x, y) coordinates relative to the 2D world simulated in Gazebo. Our robotic architecture is similar to that used by Choirbot [49], a ROS2 toolbox for cooperative robotics.

That is, we have a layer to guide and a layer to plan the destinations of the robots. Indeed, each robot is associated to a *planner_client* node which sends the *PositionGoal* related to the destination of the task associated to the robot. Then, the *planner_server* node which allows us to standardize this destination point which it publishes to the goal topic associated to the robot. Afterwards, the guidance layer will subscribe to this topic to control the robot until it reaches its mission position. Once the robot has reached its destination, the odometry server sends feedback to the *planner_client*, which can reply the next *PositionAction* to the *planner_server*.

The communication between the different nodes for the control of robot 1 for example is shown in Fig. 1.12.

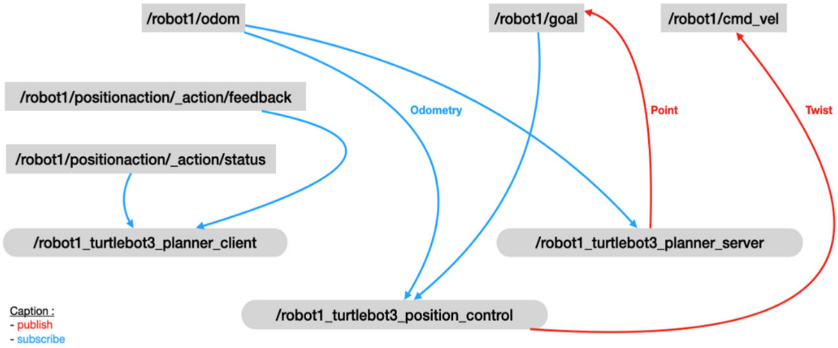

/robot1/odom
/robot1/goal
/robot1/cmd_vel
/robot1/positionaction/_action/feedback
/robot1/positionaction/_action/status
Odometry
Point
Twist
/robot1_turtlebot3_planner_client
/robot1_turtlebot3_planner_server
/robot1_turtlebot3_position_control
Caption :
- publish
- subscribe

Fig. 1.12. Nodes and topics representation for robot1 for path planning.

Once the tasks have been assigned and the robots have been controlled, the robots must be able to communicate. This will allow the exchange of information on their positions, their vision of the environment, or their trajectory intention in order not to collide with each other or with obstacles.

1.6.3. Communication

In order to implement algorithms for collective cooperation between autonomous vehicles, we have implemented the different standardized messages presented in Section 1.4. The implementation of those messages in ROS2 will allow robots to exchange these types of messages through topics.

The *turtlebot3_position_control* node of a robot allows it to control its speed and orientation towards its destination point defined by the */goal* topic as explained in Subsection 1.6.2. This node will also allow to exchange messages related to its observations of the environment thanks to its LIDAR. We have defined an observation distance and a safety distance. When the LIDAR detects something corresponding to the observation distance or less it publishes a CPM message in the topic corresponding to the robot with the related information. Similarly, if it detects something at a distance less than its safety distance, it sends an alert message, i.e., a DENM message with the information associated with the DENM topic. This process of subscribing and publishing messages similar to all robots is modelled in Fig. 1.13 using robot 1 as an example.

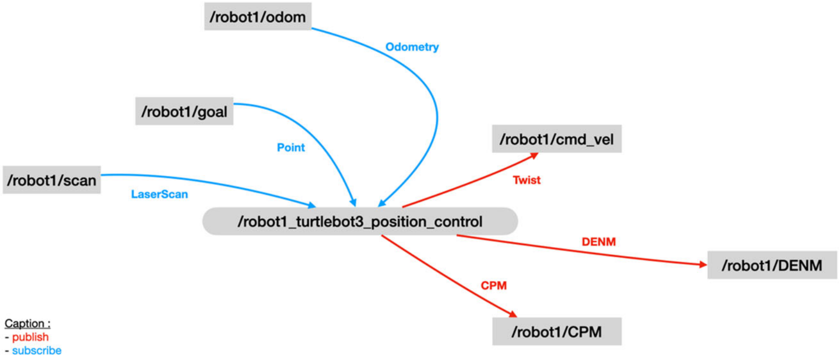


Fig. 1.13. Process of publishing/subscribing topics for position_control node of each robot.

CAM messages are published in CAM topics similar to CPM and DENM messages. They are standardized using information from the robot's odometry sensor. This information published by one robot is retrieved by all other robots in the manner of a broadcast exchange. The architecture has been implemented so that each robot has an *exchange_messages* node which allows it to subscribe to all the message topics of the other robots, i.e., CAM, CPM and DENM messages in our current experiment. These exchanges are modelled in Fig. 1.14 for 2 robots, but the process is similar regardless of the number of robots.

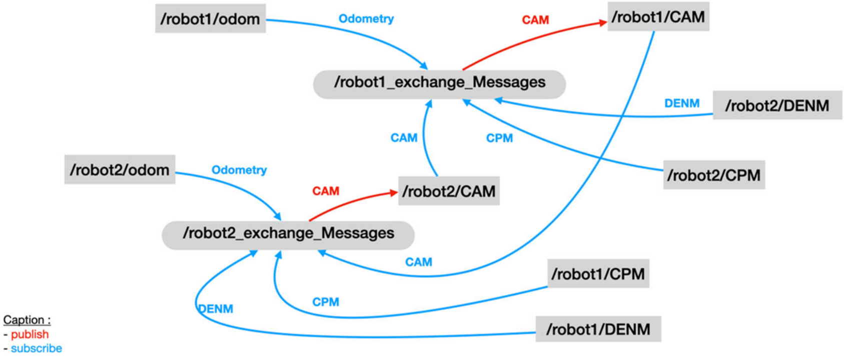


Fig. 1.14. Nodes and topics representation for exchanging messages.

1.7. Results and Discussion

Before actually experimenting with the strategic cooperation of the robots to avoid real collisions, we tested a simple scenario in a gazebo simulation. Each robot was given the task of crossing the intersection shown in Fig. 1.11, and they were asked to go to the *PositionGoal* in front of them.

We will take Fig. 1.15 as an example scenario. The robots broadcast their positions using CAM messages while moving towards the intersection. The LIDAR of robot2 detects an obstacle and sends a CPM message to share its information. But as we have not implemented obstacle avoidance control, it will stop at the safety distance of the LIDAR and send a DENM message with a risk of longitudinal collision. Similarly, robot3 and robot4 did not encounter any problems on their paths but when crossing the intersection, their LIDAR detects at the safe distance the other robot with an angle of less than 45°. They will immediately send a DENM message of risk of longitudinal collision as well.

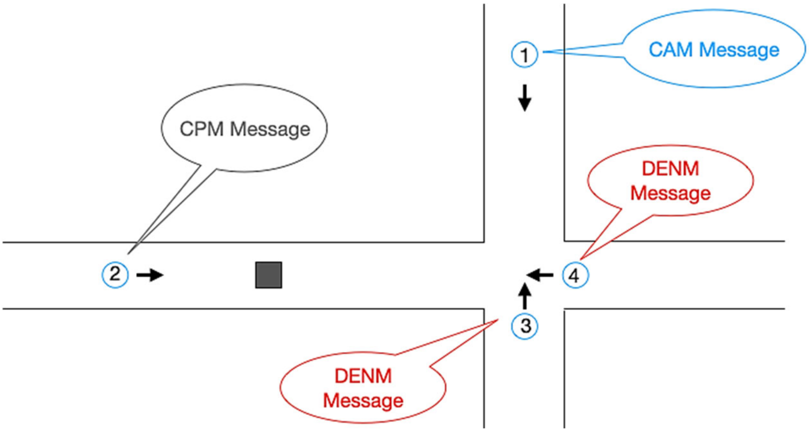
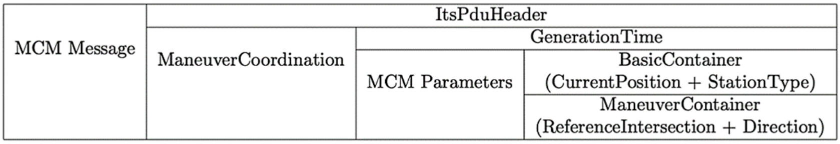


Fig. 1.15. Scenarios of message exchanging in experiments.

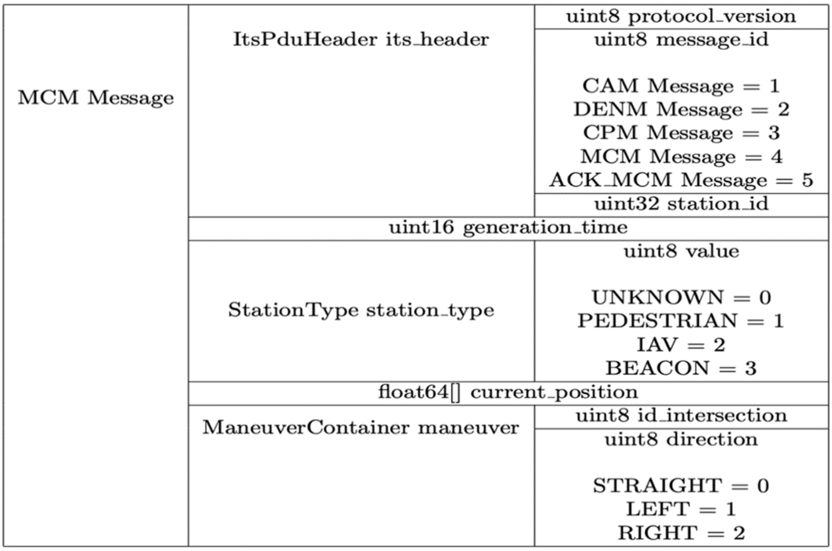
It can be noted that unlike robot2, robot3 and robot4 did not send a CPM message before, because at the moment of the intersection they were already too close. Robot3 and robot4 stop to avoid collision. Our goal was to find a cooperative solution for VIAs to avoid collisions and obstacles, but also to avoid unnecessary braking and stopping to optimise energy and speed. As a first step, these results show that our cooperative strategy allows the robots not to collide with each other or with obstacles.

Our results of communication between the different robots are therefore validated in the Gazebo simulation environment. Nevertheless, in order to be able to find a cooperation to cross the intersection it would be necessary to implement the cooperative messages presented by the Bahnes algorithm. For this purpose, a service not yet implemented by ITS WG1 and ETSI is discussed: *Maneuver Coordination Service* (MCS) and its associated *Maneuver Coordination Messages* (MCM) [43]. We then propose in the same way a representation for the industrial context in Fig. 1.16 and a modelling of the message for ROS2 in Fig. 1.17. This message would share the information that one wishes to cross an intersection by indicating *ManeuverContainer* information, that is *id* of the intersection, as well as the direction one would take.



MCM Message	ItsPduHeader		
	ManeuverCoordination	GenerationTime	
		MCM Parameters	BasicContainer (CurrentPosition + StationType)
			ManeuverContainer (ReferenceIntersection + Direction)

Fig. 1.16. Representation of MCM Message for the industrial context.



MCM Message	ItsPduHeader its_header	uint8 protocol_version
		uint8 message_id CAM Message = 1 DENM Message = 2 CPM Message = 3 MCM Message = 4 ACK_MCM Message = 5
		uint32 station_id
	uint16 generation_time	

	StationType station_type	uint8 value
		UNKNOWN = 0
		PEDESTRIAN = 1
		IAV = 2
		BEACON = 3

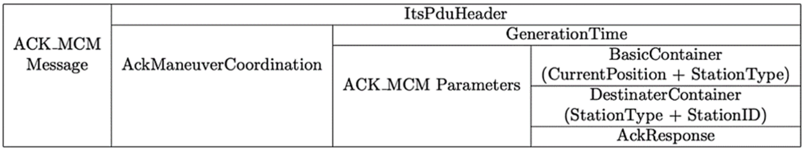
	float64[] current_position	
	ManeuverContainer maneuver	uint8 id_intersection

		uint8 direction
		STRAIGHT = 0
		LEFT = 1
		RIGHT = 2

Fig. 1.17. Modeling of MCM Message in ROS2.

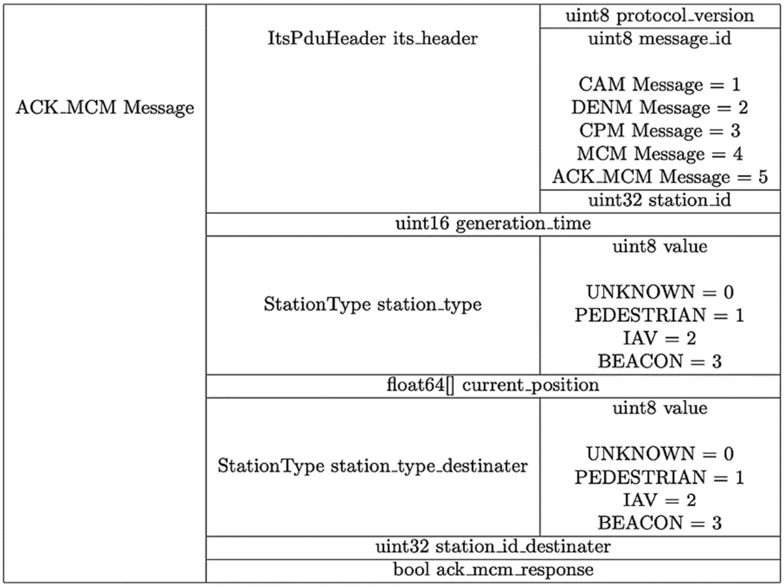
Thus, in a future work, we would assume that the vehicles know the positions of the different intersections, or are able to locate them, and send an MCM message to the other robots indicating their planned trajectory in the intersection. The robots concerned by the request will be able to respond to an *ACK_Message* indicating their agreement or disagreement with the request. The industrial representation of this message and the ROS2 modelling we propose are detailed in Figs. 1.18, 1.19.

This discussion around MCM messages to enable vehicle cooperation when approaching an intersection raises an issue: if several vehicles request to cross the same intersection at the same time, or if one vehicle indicates that it does not agree to the request of another vehicle, a deadlock situation arises. There are 3 possibilities to manage this concern. The first is the idea of strong cooperation, i.e., all vehicles always agree. The second is to set up a supervisor who can arbitrate the situation and therefore choose which vehicle should pass. This case takes us out of a distributed architecture. Finally, the last solution is to add an algorithmic layer that is known to all the robots, and which therefore serves as a decisive judgment. All robots should take the same decision in a given situation. For example, if one robot disagrees with the crossing of another, it is then the level of priority and urgency of the task between the two robots that will decide who will be the first to cross. This solution would make it possible to always remain in the idea of strong collaboration, cooperation between the different agents in the situation.



<table>
<tr><td rowspan="5">ACK_MCM Message</td><td colspan="3">ItsPduHeader</td></tr>
<tr><td rowspan="4">AckManeuverCoordination</td><td colspan="2">GenerationTime</td></tr>
<tr><td rowspan="3">ACK_MCM Parameters</td><td>BasicContainer (CurrentPosition + StationType)</td></tr>
<tr><td>DestinaterContainer (StationType + StationID)</td></tr>
<tr><td>AckResponse</td></tr>
</table>

Fig. 1.18. Representation of ACK_MCM Message for the industrial context.



<table>
<tr><td rowspan="9">ACK_MCM Message</td><td rowspan="3">ItsPduHeader its_header</td><td>uint8 protocol_version</td></tr>
<tr><td>uint8 message_id
CAM Message = 1
DENM Message = 2
CPM Message = 3
MCM Message = 4
ACK_MCM Message = 5</td></tr>
<tr><td>uint32 station_id</td></tr>
<tr><td colspan="2">uint16 generation_time</td></tr>
<tr><td>StationType station_type</td><td>uint8 value
UNKNOWN = 0
PEDESTRIAN = 1
IAV = 2
BEACON = 3</td></tr>
<tr><td colspan="2">float64[] current_position</td></tr>
<tr><td>StationType station_type_destinater</td><td>uint8 value
UNKNOWN = 0
PEDESTRIAN = 1
IAV = 2
BEACON = 3</td></tr>
<tr><td colspan="2">uint32 station_id_destinater</td></tr>
<tr><td colspan="2">bool ack_mcm_response</td></tr>
</table>

Fig. 1.19. Modeling of ACK_MCM Message in ROS2.

1.8. Conclusions and Future Work

In an Industry 4.0 context, many actors cross paths in different areas of a warehouse or a factory: vehicles, operators, obstacles (objects that fall or left in the aisles may appear).

A specific state of the art on the used a message-based communication protocol between vehicles to prioritise the passage through an intersection allowed us to identify the Bahne's algorithm [2], well representative of the cooperative strategies developed in the field. As a first step, we carried out an algorithmic work to extend this algorithm in order to have the possibility to manage the detection of fixed and mobile obstacles (Fig. 1.1). Then, we proposed an agent model as well as a model adapted for the Industry 4.0 context of the ETSI standard messages for ITS. This proposed cooperation protocol is implemented for the exchange of information on location, perception of the environment, and notification of dangerous events.

We validated the extended algorithm by a simulation approach with a traffic plan presented in the literature. Finally, we emulated these different exchanges of awareness and perception messages in a virtual world of Turtlebot3 'burger' robots with ROS2 and Gazebo.

As an extension of this work and in order to perform real experiments, we discussed these results as well as a cooperation message named *Maneuver Coordination Messages* (MCM). This implementation perspective would allow us to validate the Bahnes algorithm augmented with the exchange of cooperation messages to describe one's intention to cross an intersection. In a second step, we aim at implementing an algorithmic layer for decision making in case of deadlock situation as discussed in Section 1.7.

Finally, we also plan to involve the intersection proximity infrastructure in the exchange of communication through camera information in order to help the VIAs to cooperate in crossing the intersection by avoiding unnecessary braking and stopping. This would optimize the energy and efficiency with which the VIAs carry out their missions.

Acknowledgements

The authors would like to thank the region Bretagne-France for funding the VIASIC project as part of the ARED-2021-2024 call for projects concerning the strategic innovation area: Economics of industry for intelligent production.

References

[1]. H. Andreasson, A. Bouguerra, M. Cirillo, D. Nikolaev Dimitrov, D. Driankov, L. Karlsson, A. J. Lilienthal, F. Pecora, J. Pekka Saarinen, A. Sherikov, T. Stoyanov, Autonomous Transport Vehicles: Where We Are and What Is Missing, *IEEE Robotics & Automation Magazine*, Vol. 22, Issue 1, 2015, pp. 64-75.

[2]. N. Bahnes, B. Kechar, H. Haffaf, Cooperation Intelligent Autonomous Vehicles to enhance container terminal operations, *Journal of Innovation in Digital Ecosystems,* Vol. 3, Issue 1, 2016, pp. 22-29.

[3]. H. Khayyam, B. Javadi, M. Jalili, R. N. Jazar, Artificial Intelligence and Internet of Things for autonomous vehicles, in Nonlinear Approaches in Engineering Applications: Automotive Applications of Engineering Problems (R. N. Jazar, L. Dai, Eds.), *Springer International Publishing*, 2020, pp. 39-68.

[4]. I. F. A. Vis, Survey of research in the design and control of automated guided vehicle systems, *Eur. J. Oper. Res.*, Vol. 170, Issue 3, 2006, pp. 677-709.

[5]. B. Esmaeilian, S. Behdad, B. Wang, The evolution and future of manufacturing: a review, *Journal of Manufacturing Systems*, Vol. 39, 2016, pp. 79-100.

[6]. M. De Ryck, M. Versteyhe, F. Debrouwere, Automated guided vehicle systems, state-of-the-art control algorithms and techniques, *Journal of Manufacturing Systems*, Vol. 54, 2020, pp.152-173.

[7]. L. Monostori, P. Valckenaers, A. Dolgui, H. Panetto, M. Brdys, B. C. Csáji, Cooperative Control in Production and Logistics, *IFAC Proceeding Volumes (IFACPapersOnline)*, Vol. 19, 2014, pp. 4246-4265.

[8]. C. Blesing, D. Luensch, B. Korth, Concept of a Multi-agent Based Decentralized Production System for the Automotive Industry, *Lecture Notes in Computer Science*, Vol. 10349, 2018, pp. 19-30.

[9]. M. P. Fanti, A. M. Mangini, G. Pedroncelli, W. Ukovich, A decentralized control strategy for the coordination of AGV systems, *Control Eng. Pract.*, Vol. 70, 2018, pp. 86-97.

[10]. I. Draganjac, D. Miklic, Z. Kovacic, G. Vasiljevic, S. Bogdan, Decentralized Control of Multi-AGV Systems in Autonomous Warehousing Applications, *IEEE Trans. Autom. Sci. Eng.*, Vol. 13, Issue 4, 2016, pp. 1433-1447.

[11]. J. Xie, C. C. Liu, Multi-agent systems and their applications, *J. Int. Council. Electr. Eng.*, Vol. 7, Issue 1, 2017, pp. 188-197.

[12]. H. Meissner, R. Ilsen, J. C. Aurich, Analysis of Control Architectures in the Context of Industry 4.0, *Proc. CIRP*, 2017, Vol. 62, pp.165-169.

[13]. X. Jia, M. Q. Meng, A survey and analysis of task allocation algorithms in multi-robot systems, in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO'13)*, 2013, pp. 2280-2305.

[14]. A. R. Mosteo, L. Montano, A survey of multi-robot task allocation, Report AMI-009-10-TEC, *Instituto de Investigación en Ingeniería de Aragón*, 2010, pp. 1-27.

[15]. B. P. Gerkey, M. J. Mataric, A formal analysis and taxonomy of task allocation in multi-robot systems, *Int. J. Robot. Res.*, Vol. 23, Issue 9, 2004, pp. 939-954.

[16]. A. Khamis, A. Hussein, A. Elmogy, Multi-robot Task Allocation: A Review of the State-of-the-Art, *Coop. Robots Sensor. Netw.*, Vol. 2, 2015, pp. 31-51.

[17]. G. J. Cawood, I. A. Gorlach, Navigation and locomotion of a low-cost Automated Guided Cart, in *Proceedings of the Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech'15)*, 2015, pp. 83-88.

[18]. H. M. Barbera, J. P. C. Quinonero, M. A. Z. Izquierdo, A. G. Skarmeta, i-Fork: a flexible AGV system using topological and grid maps, in *Proceedings of the IEEE International Conference on Robotics & Automation (ROBOT'03)*, Vol. 2, 2003, pp. 2147-2152.

[19]. S. Lu, C. Xu, R. Y. Zhong, L. Wang, A RFID-enabled positioning system in automated guided vehicle for smart factories, *J. Manuf. Syst.*, Vol. 44, 2017, pp. 179-190.

[20]. C. Feledy, S. Luttenberger, A state of the art map of the AGVS technology and a guideline for how and where to use it, Tech. Rep., *University of Lund*, 2017.

[21]. S. W. Yoon, S. B. Park, J. S. Kim, Kalman Filter Sensor Fusion for Mecanum Wheeled Automated Guided Vehicle Localization, *J. Sensors*, Vol. 2015, 2015, 347379.

[22]. A. Ndao, M. Djoko-Kouam, A.-J. Fougères, Matrix Beaconing for the Location of Autonomous Industrial Vehicles on a Simulation Platform, in *Proceedings of the 3rd Int. Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2021)*, Porto, Portugal, November 17-19, 2021, pp. 80-86.

[23]. V. Kunchev, L. Jain, V. Ivancevic, A. Finn, Path planning and obstacle avoidance for autonomous mobile robots: a review, in *Proceedings of the 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES' 06)*, 2006, pp. 537-544.

[24]. D. H. Kim, N. T. Hai, W. Y. Joe, A Guide to Selecting Path Planning Algorithm for Automated Guided Vehicle (AGV), *Lect. Notes Electr. Eng.*, Vol. 465, 2018, pp. 587-96.

[25]. E. Cardarelli, V. Digani, L. Sabattini, C. Secchi, C. Fantuzzi, Cooperative cloud robotics architecture for the coordination of multi-AGV systems in industrial warehouses, *Mechatronics*, Vol. 45, 2017, pp. 1-13.

[26]. A. M. Nascimento, L. F. Vismari, C. B. S. T. Molina, P. S. Cugnasca, J. B. Camargo, J. R. de Almeida, A. Y. Hata, A Systematic Literature Review About the Impact of Artificial Intelligence on Autonomous Vehicle Safety, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, Issue 12, 2019, pp. 4928-4946.

[27]. S. C. Srivastava, A. K. Choudhary, Development of an intelligent agent-based AGV controller for a flexible manufacturing system, *Int. J. Adv. Manuf. Technol.*, Vol. 36, 2008, pp. 780-797.

[28]. K. Jose, D. K. Pratihar, Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods, *Robot. Autonom. Syst.*, Vol. 80, 2016, pp. 34-42.

[29]. V. Digani, F. Caramaschi, L. Sabattini, C. Secchi, C. Fantuzzi, Obstacle avoidance for industrial AGVs, in *Proceedings of the IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP'14)*, 2014, pp. 227-232.

[30]. A. Winkler, J. Suchý, Dynamic Collision Avoidance of Industrial Cooperating Robots Using Virtual Force Fields, *IFAC Proceedings Volumes*, Vol. 45, 2012, pp. 265-270.

[31]. M. Jäger, B. Nebel, Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (IROS'01)*, Vol. 3, 2001, pp. 1213-1219.

[32]. D. Marino, A. Fagiolini, L. Pallottino, Distributed collision-free protocol for AGVs in industrial environments, *arXiv Preprint*, 2011, arXiv:1101.2270.

[33]. G. Demesure, M. Defoort, A. Bekrar, D. Trentesaux, M. Djemai, Decentralized Motion Planning and Scheduling of AGVs in an FMS, *IEEE Trans. Industrial Informatics*, Vol. 14, Issue 4, 2018, pp. 1744-1752.

[34]. Q. S. Kabir, Y. Suzuki, Comparative analysis of different routing heuristics for the battery management of automated guided vehicles, *Int. J. Prod. Res.*, Vol. 57, Issue 2, 2018, pp. 624-641.

[35]. M. De Ryck, M. Versteyhe, K. Shariatmadar, Resource management in decentralized industrial Automated Guided Vehicle systems, *J. Manuf. Syst.*, Vol. 54, January 2020, pp. 204-214.

[36]. C. Medrano-Berumen, M. İlhan Akbaş, Validation of decision-making in artificial intelligence-based autonomous vehicles, *Journal of Information and Telecommunication*, Vol. 5, Issue 1, 2021, pp. 83-103.

[37]. A. Daniel, K. Subburathinam, B. Muthu, N. Rajkumar, S. Pandian, Procuring Cooperative Intelligence in Autonomous Vehicles for Object Detection through Data Fusion Approach, *IET Intel. Transport. Syst.*, Vol. 14, Issue 11, 2020, pp. 1410-1417.

[38]. Intelligent Transport Systems (ITS), Vehicular Communications, Basic set of Applications, Part 2: Specification of Cooperative Awareness Basic Service, ETSI EN 302 637-2 V1.3.2, *ETSI*, 2014.

[39]. N. Lyamin, A. Vinel, M. Jonsson, B. Bellalta, Cooperative Awareness in VANETs: On ETSI EN 302 637-2 Performance, *IEEE Transactions on Vehicular Technology*, Vol. 67, Issue 1, 2017, pp. 17-28.

[40]. Intelligent Transport Systems (ITS), Vehicular Communications, Basic set of Applications, Part 3: Specification of Decentralized Environmental Notification Basic Service, ETSI EN 302 637-3 V1.2.1, *ETSI*, 2014.

[41]. J. Santa, F. Pereniguez-Garcia, A. Moragón, A. Skarmeta, Experimental evaluation of CAM and DENM messaging services in vehicular communications, *Transportation Research Part C: Emerging Technologies,* Vol. 46, 2014, pp. 98-120.

[42]. Intelligent Transport Systems (ITS), Vehicular Communications, Basic set of Applications, Analysis of the Collective Perception Service (CPS), Release 2, ETSI TR 103 562 V2.1.1, *ETSI*, 2019.

[43]. A. Correa, et al., On the Impact of V2X-based Maneuver Coordination on the Traffic, in *Proceedings of the IEEE 93rd Vehicular Technology Conference (VTC2021-Spring'21)*, 25 April – 19 May, 2021, pp. 1-5.

[44]. J. Grosset, A.-J. Fougères, M. Djoko-Kouam, C. Couturier, J.-M. Bonnin, Simulation of a collision and obstacle avoidance algorithm for cooperative industrial autonomous vehicles, in *Proceedings of the 2nd IFSA Winter Conf. on Automation, Robotics & Communications for Industry 4.0 (ARCI'22)*, Andorra la Vella, Andorra, February 2-4, 2022, pp. 88-91.

[45]. A.-J. Fougères, A Modelling Approach Based on Fuzzy Agents, *Int. J. of Computer Science Issues*, Vol. 9, Issue 6, 2013, pp. 19-28.

[46]. C. M. Macal, Everything you need to know about agent-based modelling and simulation, *Journal of Simulation*, Vol. 10, Issue 2, 2016, pp. 144-156.

[47]. D. Bechtsis, N. Tsolakis, D. Vlachos, J. S. Srai, Intelligent autonomous vehicles in digital supply chains: a framework for integrating innovations towards sustainable value networks, *Journal of Cleaner Production*, Vol. 181, 2020, pp. 60-71.

[48]. Y. Maruyama, S. Kato, T. Azumi, Exploring the performance of ROS2, in *Proceedings of the 13th International Conference on Embedded Software (EMSOFT'16)*, 2016, pp. 1-10.

[49]. A. Testa, A. Camisa., G. Notarstefano, ChoiRbot: A ROS 2 Toolbox for Cooperative Robotics, *arXiv Preprint*, 2020, arXiv:2010.13431.

Chapter 2

Platform-agnostic Digital Twins for Safer Human-robot Collaboration

Aïcha Rizzotti-Kaddouri, Loïck Jeanneret, Brendan Studer, Javier Bracamonte and Nabil Ouerhani

2.1. Introduction

Robots are becoming an essential enabling technology in almost all economic and societal sectors. The number of deployed industrial robots evolved from 1 million units in 2009 to 3 million units in 2022 [1]. Their declining manufacturing costs explains, on one hand, the increasing massive deployment of collaborative robots (cobots) in the workspace. Their increased flexibility and versatility, on the other hand, makes now robots suitable for a new range of tasks other than the traditional heavy-duty operations. This means it is now possible not only the coexistence of humans and robots in the same workplace, but also their collaboration. The robots would execute tedious and repetitive tasks, whereas human operators would undertake those manipulations which require a higher degree of dexterity and unexpected task adaptation. Notwithstanding, a full-scale deployment of this human-machine collaboration is still in many cases challenged by the fact that the proximity between robots and humans can raise a serious issue related to human safety. In this chapter we address this challenge and explore solutions to tackle this issue.

The content of this chapter reports extended material of our journal paper [2] in which we presented a solution to the complex task of programming robots in view of a safe human-machine collaboration. We also report

Aïcha Rizzotti-Kaddouri
School of Engineering, University of Applied Sciences and Arts Western Switzerland, HES-SO/HE-ARC, 2000 Neuchâtel, Switzerland

additional research on two developed frameworks: one is a virtual-reality environment while the second is an augmented-reality one. The objective being to address the design and the operation of safer workspaces (e.g., zero collisions) in industrial human-robot collaboration. The virtual reality VR environment is intended to design and to set up a safer the human-cobot workspace, while the augmented reality AR environment is intended to increase the safety during the operational phase, i.e., during the real-time, real-world human-robot collaboration. We present the results of using both environments along with a System Usability Scale SUS-based assessment.

2.2. Related Works

The LfD (Learning from Demonstration) robot-programming approach has attracted a lot of interest in the field of human-robot interaction [3, 4]. This topic encompasses several disciplines and scientific fields.

In general, there are two main approaches for LfD. The first one is based on observational learning which usually exploits a vision system for the perception of movements and gestures, by using for example, cameras. The second approach is based on kinesthetic guidance which refers to the manual movements of the robot by interaction through haptic sensors.

It should be noted that the second approach simplifies the corresponding learning task, but the movements remain spatially limited. Moreover, it is not possible to perform kinesthetic guidance on all types of robots. Therefore, in this study we favored the vision-based learning approach. We note that there are several important phases in this workflow, namely: *data capture and fusion > learning phase > reproduction by the robot of the learned movement or action.*

Relevant existing work in this area is reported below.

In the framework of the European project PRACE (Productive Robot ApprentiCE), whose aim is the development of a mobile robotic platform for the automation of assembly operations, a system based on several Kinect-type cameras to calculate 3D positions has been investigated. In [5], the author adopted a top-down approach: first, an estimation of the positions was made, followed by a refinement of the data captured by each sensor.

In the LfD domain, there are other approaches [6] that are based on the fusion of data recorded with gloves and video cameras. The work in [7] up to a certain point inspired our approach of fusing data from Microsoft's Kinect V2 to get the 3D data in skeletal form and supplementing it with data from Intel's RealSense sensor to refine the depth.

One example of this approach is also reported in [8], where the robot must perform assembly tasks in the industrial Peg-in-Hole (PiH) domain. Here, a learning phase and a reproduction phase reinforced by a kinesthetic guidance phase are listed. With a camera on the robot's wrist, the object is detected, located and captured. The objects are labeled, and the detection algorithms are based on conventional computer vision algorithms such as SIFT (Scale-Invariant Feature Transform) and KNN (K-Nearest-Neighbors) for classification.

We note that in [9] the human arm and the robot arm are physically attached for learning. The trajectory of each action is followed by the robot arm giving a representation of the trajectory in space. For the demonstration, the authors used the 3D coordinates of the three joints (shoulder, elbow, and wrist) for the tracking. They also used the procedure named Gaussian Process Latent Variable Model (GPLVM) and RANSAC (RANdom SAmple Consensus) to map the motion behavior.

In [10] the imitation is done by decomposing the main task into hierarchical subtasks, based on an RNN neural network to predict the next task to be performed. This will of course depend on the observed input of the current state with the control of a closed loop.

The work reported in [11] is carried out by using the ABB YuMi robot which is the same robot we use in our laboratory. The aim of the experiment was to test the effects of reusing non-expert programming parameters and skills for assembly tasks on industrial robots.

Our approach leveraged the results from previously reported studies. Our objective nonetheless was to increase accuracy (a recurrent problem in the cited works) either by using multi-channel vision systems for the data captures (unlike [4, 9]), or by obtaining finer resolution movements (unlike [5-7]).

Various international standards have addressed the safety aspects of industrial robots in general and of collaborative robots in particular [12-14]. In [15], the authors exposed clear definitions of the different terminologies related to safety in collaborative robots and the link to the different ISO standards.

In the frame of our studies, we followed the ISO recommendations for safeguarding in human-robot collaborative tasks. Our research was particularly oriented to develop Speed and Separation Monitoring (SSM) methods for collision avoidance while continuously monitoring the motion of humans in the workspace. Several works have been reported on this subject. For example, an in-depth review of vision-based systems for enhancing human safety in human-robot collaboration scenarios was presented in [16].

2.3. Demonstration-based Robot Learning

Efficient robot programming is one of the challenges faced during the deployment of cobots. Demonstration-based robot learning [17] is an active research area which studies robot programming. In this area, capturing and replicating the motion with high accuracy remains a recurrent issue. Our approach [2] to tackle the accuracy problem was to use a system composed of multiple sensors [18] and then to extract a trajectory from the motions and the (programmer's) fingers' positions using machine learning.

We tested different sensors including gloves and cameras. Due to accuracy and occlusion problems, we opted for a solution which uses an HTC Vive Tracker sensor [19], which is indeed accurate enough for robotic applications [20]. In our workflow, the HTC VIVE Tracker sensor is attached to the application-dependent tool which is used by the operator "to demonstrate" the motion. The Tracker sensor can then track the movement of the tool in three dimensions by locating its position relative to a VIVE Base Station.

Since the tool position and rotation is directly tracked, it is not necessary to interpret (or to process) the intent and gestures of the user, which would have required more complex machine learning models. Moreover, our solution is coupled with AR smart glasses which allows the user to have immediate feedback. The used AR glasses (the Microsoft HoloLens 2) also provide tools to correct, in an intuitive way, possible recording errors.

2.3.1. Safe Human-robot Collaboration Testing

Robotic Task Planning, which is a sub-category of automated planning and scheduling, aims at solving complex robot use-case scenarios [21]. While some problem solvers such as STRIPS [22] can integrate human safety, as described for example in [23], we cannot integrate such a system when a task is learned from demonstration. As an alternative, we approached this issue by testing the learned tasks in a high-fidelity virtual reality environment, where the human operator does not risk any harm. In our safe testing environment multiple digital twins can be placed in the scene along with a real human operator. The digital twins can be connected either to an alternative real simulator or to real robots located in a separate room.

2.4. VR and AR Environments for Safe Cobotics

Safety in the workplace is a non-negotiable requirement for any human-robot collaboration task. One of the main challenges for a wide deployment of robots/cobots is related to safety issues. To unlock the full potential of collaborative robotics in industry and society, human safety must be guaranteed. At the same time, investment in the safety of human-robot collaboration must not reduce the promised return on investment.

Below we report the current status of the development of a VR/AR system intended for achieving safer workspaces (e.g., zero collisions) and increased productivity in industrial human-robot collaborations.

During the real-time cobot operation phase, the same digital twin used in Virtual Reality is used in Augmented Reality with AR smart glasses (in this case, the Microsoft HoloLens 2), while simultaneously calculating and displaying other kinds of metrics. One, for example, is the Safety Score metric. This score indicates to the operator the degree of safety of the current situation (or position). If the situation is deemed too risky, or when the operator crosses a dangerous zone, the smart glasses can send a signal to slow down or to stop the robot.

2.4.1. System Architecture

The system is composed of two main sub-systems as can be seen in Fig. 2.1. Each of the two main parts of this figure can operate independently on its own. They communicate together through robots and simulators by sharing programmed trajectories.

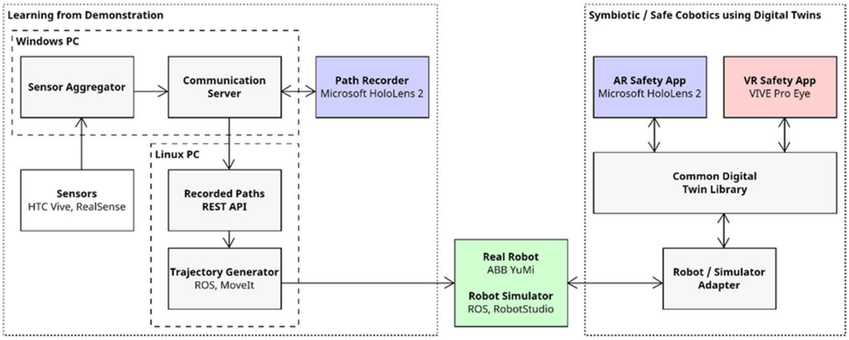


Fig. 2.1. Learning from demonstration and safe cobotics using digital twins.

The “Common Digital Twin Library” component (on the right side of Fig. 2.1) provides a generic way to define robots for Virtual Reality and Augmented Reality called Robot Components. Robot Components allow us to define an existing robot or to prototype a new one by using small re-usable parts.

Robot Components can be very specific and accurate (e.g., an accurate 3D model of an existing robot) or very generic and re-usable (e.g., a robotic joint, a 3D collider, a generic gripper, etc.). They can then be combined, using a parent/child system to define a fully working robot. This allows us to add support for many robots' features with minimal effort, as long as the simulator or the real robot provides an API for it.

Robot Components can communicate to real robots and simulators through the “Robot / Simulator Adapter” component, which is a custom NodeJS server with an adapter system.

Plugins can add new adapters, which allows it to communicate with both real robots and simulators. Robot Components and the plugin system make it possible and easy to add support for new robots if they have a public API.

Robots are installed in a modeled room (which represents a real-life industrial room), which can then be loaded in Virtual Reality to test the scenario, or in Augmented Reality (without furniture) to show to the operator where the robots are. Fig. 2.2 shows the class diagram of how robots are stored in a room.

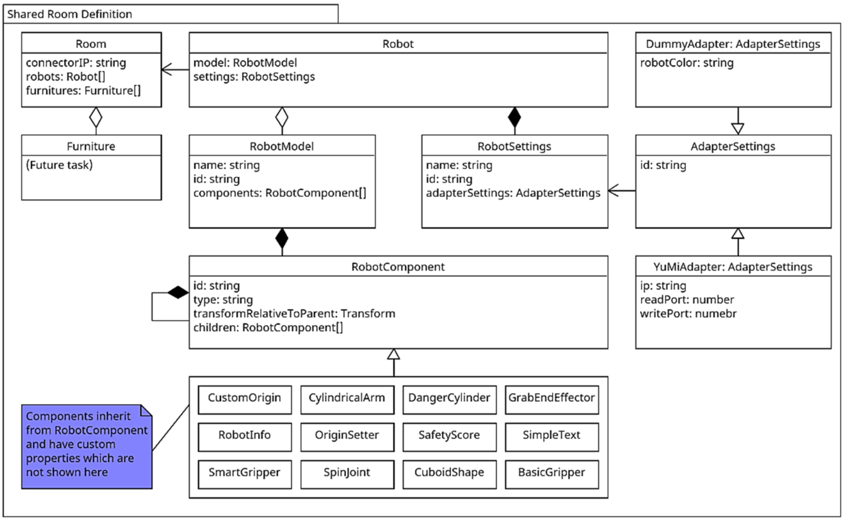


Fig. 2.2. Room definition class diagram used in the VR and AR environments.

Our *Learning from Demonstration system* (on the left in Fig. 2.1) is composed of multiples modules and several physical sensors. Two *sensor aggregator applications* retrieve and merge data from the sensors. A communication bus exchanges data between the HoloLens 2 module, the web server, and the sensor aggregators. The server module stores the recorded trajectories. It is then accessed by the ROS (Robot Operating System) module, which converts the operator trajectories into robot trajectories, before sending them to the robot. The sensor aggregator is built in a way that makes it easy to add new sensors, by allowing each sensor to validate or improve the accuracy of previous sensors.

2.4.2. Sensors

2.4.2.1. Leap Motion and Deep Learning

The goal of this part is to set up a system for the detection and classification of gestures performed by a human hand. The acquisition of gesture data is done using a Leap Motion, and the analysis and classification with a deep learning model (see Fig. 2.3). The process is carried out in three parts: The first one is the data acquisition, the second one is the creation and training of an MLP (Multilayer Perceptron) neural network, and the third part consists in importing the previously created model and making predictions on the fly with it. The features used are the relative distances between the fingers, which allows reducing the neural network complexity.

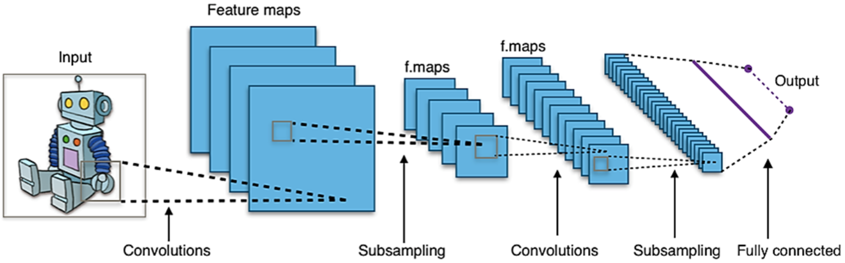


Fig. 2.3. Classification with the deep learning model.

The main objectives for this part were successfully attained. For example, the “pinch” and “thumb up” hand gestures were correctly recognized. Some others hand signs can of course be added at the expense of longer training and slightly lower performance. One recurrent issue we had was the misinterpretation of gestures that are differentiated only by hand rotation. A straightforward solution to this problem and/or a possible improvement will be the inclusion of hand orientation as an additional feature for the neural networks training.

2.4.2.2. Camera

In order to increase accuracy, we opted to use the RealSense D435 sensor (depth camera), which allows the detection of objects with high precision. Subsequently we merged the captured information with the Trackers’ data. The depth camera is placed above the workspace. An algorithm detects the plane of the table and then a threshold is applied to the depth of each pixel. This allows to detect items dimensions with accuracy (±3 mm delta error) on different types of boxes. See Fig. 2.4.

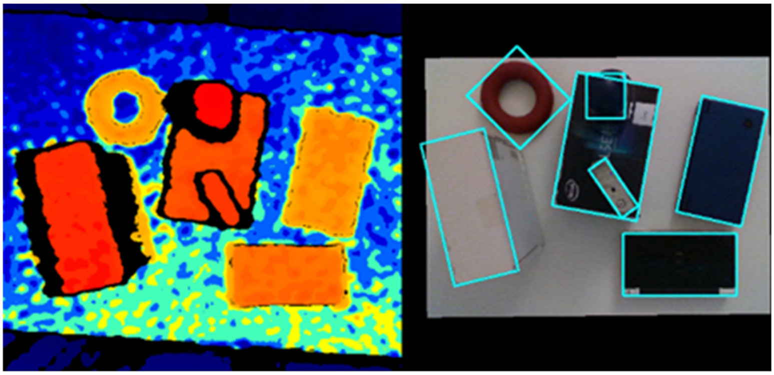


Fig. 2.4. Camera sensor and boundary detection algorithm.

2.4.2.3. Hand Tracking with Gloves

Multiple hand tracking sensors were tested, including the Hi5 Glove and the Senso Gloves before settling for the HTC VIVE Tracker. The main disadvantages of the non-selected options were that they could be cumbersome to wear and that they featured a lower tracking accuracy. Additionally, their main purpose is to track the fingers and hand position of the operator, and for our application, it was not always possible to convert (post-process) the obtained data to accurately describe the motion of the *tool*.

To solve this in a pragmatic way, we decided to attach the tool to the Tracker. Since we could not use the gripper of the robot directly, we 3D-printed a “pen” representing a gluing tool (see use-case in the next section). However, in future work, it will be possible to print custom tips, either to attach existing tools, or to print “smart” tools. For example, a gripper with a small electronic circuit could automatically track the opening of the clamp during the recording, which would be feasible via the HTC VIVE Trackers’ programmable pins.

2.4.2.4. HTC VIVE Tracker and AR Smart Glasses

A pair of Augmented-Reality/Mixed-Reality glasses were used in combination with two HTC VIVE Trackers. The selected AR/MR glasses were the Microsoft Hololens 2 glasses [24]. The system architecture is, all the same, open to the use of other types of glasses or models from other manufacturers.

One of the HTC VIVE trackers was used to track the tool and the other to define the origin. A quick response QR code (see Fig. 2.5) was used to synchronize the origins of the Tracker and the HoloLens 2. The operator can view the limits of the workspace through the AR/MR smart glasses and use controls to start and stop the recording (Fig. 2.5).

Once the task is recorded, it can be uploaded to a web server for future usage. The communication between the web server, the HoloLens 2, and the Trackers is done in a modular way.

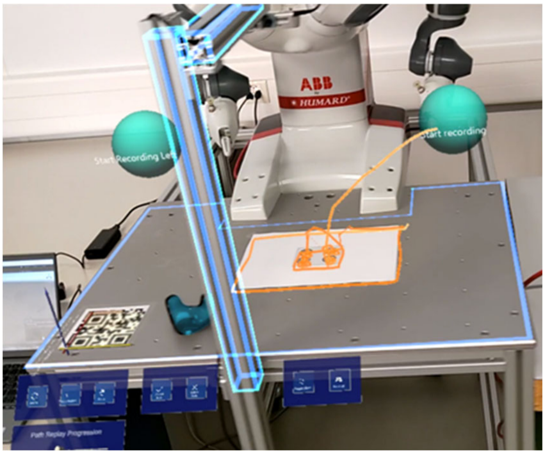
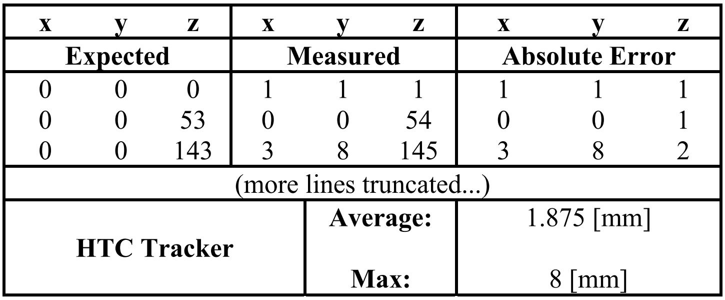


Fig. 2.5. AR smart glasses screen capture.

The accuracy of the Tracker with the pen was measured by touching with the tip of the pen at specific (known distance) points drawn on a sheet of paper. The results of the positions are summarized below in Fig. 2.6.

The use case for our demonstration was a “gluing task”, which has the following sequence: the operator picks up an item, moves it, applies glue on it, and then glues the other half together. Fig. 2.7 shows the operator during the recording of the trajectory, which is displayed in orange color. While this figure displays a 2D path, the path recorded is indeed three-dimensional.

For the reported use case, an ABB YuMi robot was used. This robot has two arms. The operator can decide which arm is used *while in teaching mode* by simply touching the arm of the robot with the pen. Technically, two Trackers could be used to record trajectories for both arms at the same time.



x	y	z	x	y	z	x	y	z
Expected			Measured			Absolute Error		
0	0	0	1	1	1	1	1	1
0	0	53	0	0	54	0	0	1
0	0	143	3	8	145	3	8	2
(more lines truncated...)								
HTC Tracker				Average: Max:		1.875 [mm] 8 [mm]		

Fig. 2.6. Accuracy of the HTC Vive Tracker.

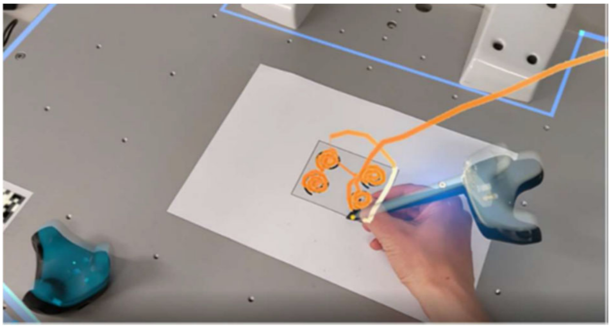


Fig. 2.7. Recording the trajectory.

2.4.3. VR Environment for a Safe and Efficient Human-robot Workspace

Once the trajectories are recorded using the method described in the previous sections, they are validated in a Virtual Reality environment. Contrary to traditional approaches that would try to test the trajectories in real conditions with a physically present robot, we opted to equip the operator with an HTC Vive Pro eye head mounted display [25] and use Unity’s Virtual Reality features [26] coupled with digital twins. Such a system reduces the injury risks of the human operator to practically zero when testing new trajectories. This riskless procedure also minimizes the overall testing time. Furthermore, a virtual environment allows it to create new testing scenarios that would be complex and/or more expensive to set up in real life. The VR system also facilitates the testing of different or alternative configuration scenarios in search of increasing productivity.

The following paragraphs describe the operation of the VR system in more detail:

1. The operator can (virtually) observe both the robot and the surrounding environment;
2. The operator can then at real-time speed interact with the robot and the environment as in a classic Virtual Reality application. The simulation is as close as possible to the real-world case from the operator's standpoint. A connection with the Robot Adapter allows the robot to mimic real-life robot trajectories;
3. For post-analysis and optimization purposes, the VR system offers the capability of recording the scenario and all the involved actors (e.g., the robot(s), the operator in the room);
4. Diagnostic tools and metrics are also constantly shown to the human operator. For example, we can display the minimum distance between the operator and the robot(s) for every timestamp during the execution of the use case scenario; or the number of occurred human-robot (virtual) collisions. Other metrics can be easily implemented and added in graphic form to the displayed information. Fig. 2.8 shows the diagnostic view for a very simple use-case example. On the top left part of the display the number of collisions is reported. On the right part, the minimum distances and robot speed plots are displayed.

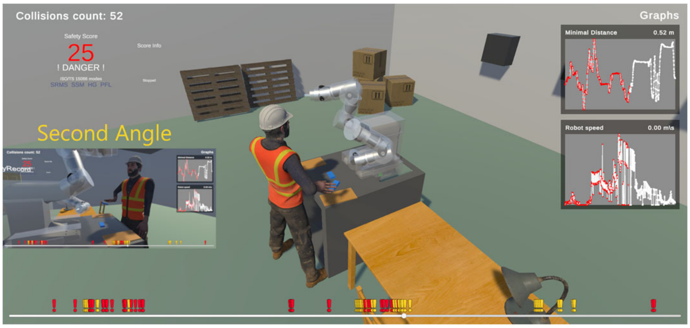


Fig. 2.8. Diagnostic view in a simple use case.

2.4.3.1. Testing the System

2.4.3.1.1. VR-AR Working Environment Setting

Fig. 2.9 depicts the main components and tools of the AR-VR working environment in which we carried out the developments and experimentations.

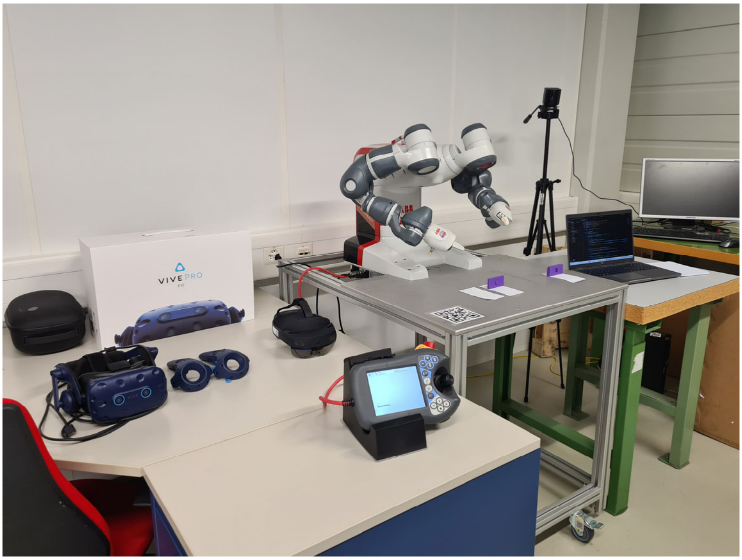


Fig. 2.9. VR-AR development and experimentation setting.

For the VR experiments, we had a computer with an Nvidia GTX 2070 GPU and an HTC Vive Pro Eye HMD [25]. An open free space of 2×2 m^2 just in front of the computer was enough to test all kinds of movements with the six degrees of freedom necessary to test and tune up our VR application.

In a relatively larger open space, we developed and tested our AR application. The ABB Yumi shown in the picture is mounted on a metallic table that is used both by the "human operator" and by the robot as a collaborative workspace. It is emphasized that the dimensions of the table, the objects required in the testing, and the robot itself, are accurately represented in the VR and AR scenes. This allowed it to make it possible for an almost-perfect representation of the real scene in VR and to superimpose the robot and its virtual representation in AR.

2.4.3.1.2. Results: AR System

In order to test our system on a practical application, we implemented a (robot-executed) pick-and-place operation which included an intermediate (human-executed) assembly task, leading to the sequence: pick-assembly-place. We used multiple sensors and a real (not a virtual one) ABB YuMi robot (see Fig. 2.9).

During the tests, it was confirmed that the hand- tracking accuracy (by using the Microsoft HoloLens 2) was high enough to control the ABB YuMi robot.

Our plugin-based system permits us to quickly and easily add new robots and simulators, which makes our solution fully platform-agnostic. That means in case we have to integrate a new robot model; we just have to create a new plugin for this robot (while respecting its interface) and the robot can then be straightforwardly integrated into the system. For example, we exercised this feature by providing multiple plugins for multiple entities: a) a plugin that can communicate with a real ABB YuMi, b) another plugin with a simulated ABB robot, and c) a third plugin with a modeled "dummy" (robot) which was used to simply test the feasibility with any prototyped robot.

The developed modular Robot Components system was used to define an ABB Robot, which can be animated using data from real robots and simulators. These digital twins can be viewed in Virtual Reality and in Augmented Reality.

The implemented solution allows the managing of the tasks on a running robot. When the robot is stopped, it can be controlled in an intuitive manner.

Safety features were implemented. For example, displaying a danger zone (the circular yellow strips in Fig. 2.10); or slowing or stopping the robot depending on the value of the safety score or on the distance of the operator to the robot.

Furthermore, the robots can be visualized in the real world, in virtual reality or in a third-party 3D application.

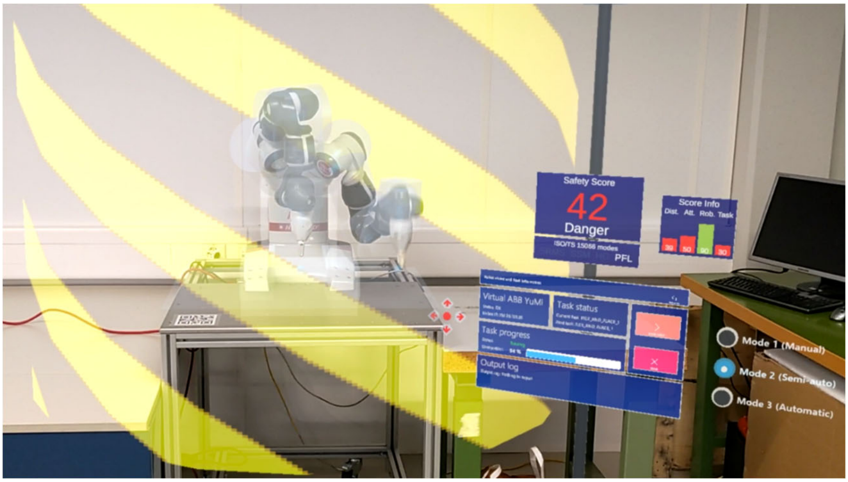


Fig. 2.10. Screen capture of the AR smart glasses during regular human-robot operation. The human operator can visualize in real-time the danger zone, the safety score, and other valuable information.

2.4.3.1.3. Results: VR System

Like in the AR case, we tested our VR framework by implementing the (robot-executed) pick-and-place operation including the already mentioned intermediate (human-executed) assembly task, which leads to the sequence: pick-assembly-place (Fig. 2.11).

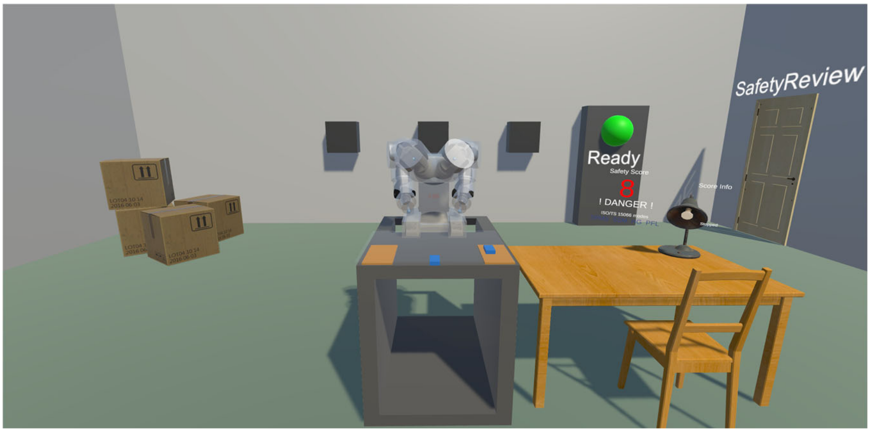


Fig. 2.11. Screen capture of the VR environment for the reported use-case.

Our VR application monitors the movements of the robot and of the operator while they are executing the collaborative task. The tool, on a continuous time basis, is accurately identifying and reporting all the human-machine collisions (that occur in the virtual world), so that the necessary measures be taken to avoid such collisions in a real-life situation (e.g., redesign of the workspace, redesign the sequence of the manipulations, etc.).

Hand-tracking and eye-tracking information is also monitored and reported to be used with motion prediction modules in order to increase safety. These modules are use-case dependent and can be easily added to the system at a later stage. The operator's virtual body is displayed to give a real-life feeling to the user.

The VR system is tightly coupled with our modular Robot Components System. The latter is used to define an ABB Robot which is displayed to the user and can be animated using data from real robots or simulators or even by using the replay feature of the connectors which permits to get data from a previously recorded use case.

The VR system has two phases. a) The Record Phase, which alike our AR framework, allows a user to test and to record a human-robot collaboration scenario in a fully virtual environment. And second, b) The Replay Phase, which is used to get an insight about the recorded scenario.

Furthermore, the implemented solution offers analysis tools which allow the user to examine and better understand the occurred events during the Record Phase. The tools also provide additional useful information such as the minimal distance between the operator and the robot at any time of the record. This, as explained above, in order to make the necessary adjustments (workspace layout redesign, re-adapt the manipulations, etc.) in order to guarantee a collision-free human-robot collaboration.

Safety features such as the danger zone or the safety score are also displayed to keep consistency and mutual reference with the AR environment (Fig. 2.11).

2.4.3.1.4. Usability and User Testing

The reported frameworks deal, among the described topics, with human interaction, requiring the design of human-robot interaction (HRI) [27]. Within the frameworks, we have used a structured method focused on user experience (UX) to study and map out the requirements for quality interaction. The System Usability Scale (SUS) [28] provides a method for measuring the usability of a system or a tool. It consists of a 10-item questionnaire with five response options for respondents: from Strongly agree to Strongly disagree.

In order to test the degree of usability of our platforms from a *new user* perspective, we deployed a System Usability Scale (SUS) test. This allowed us to get an insight into the usability of our applications and to gather ideas for future improvements. Given the asymmetry of development status between our AR and VR systems, we decided to implement the SUS tests only on the AR platform.

We selected three different people among the staff of our laboratory who were unaware of the details of the developed system. To each of them, we briefly introduced the platform and then asked them use the AR application using the Microsoft Hololens 2 and the ABB Yumi. The test consisted of executing the above-described pick-assembly-and-place sequence and moving the robot's arms using our interface. At the end of the executed tasks, each user had to complete the 10-item SUS questionnaire.

The mean score of our application was 60.83. Typically, a figure of 68 is considered an *above average* SUS score, which means that our AR application is below average in terms of usability, but all the same it is still at an acceptable level. This result was quite expected, since our first development efforts of the platforms were rather oriented to achieve full functionality (when used by members of the development team). The result of the SUS tests has however indicated that an improvement of the user interface would be part of the future work list to expand the tool to a broader users base.

2.5. Conclusions

Collaborative robots represent an effective option for promoting human-machine interaction in view of increasing productivity in the manufacturing room. Cobots relieve humans of repetitive, dangerous, non-ergonomic or heavy-load tasks. Cobots and humans would collaborate side-by-side sharing the same workspace.

Given the multiplicity of use cases that this collaboration makes possible, the design and deployment of a cobot still faces a certain number of challenges before it is fully deployed in the assembly room. One of these challenges remains guaranteeing the human safety. This is the topic we addressed in the chapter.

More specifically, we reported extended and complementary research work based on our previous results on "Learning from Demonstration" robot programming for safety collaboration. As described in the initial sections of the chapter, by means of a series of sensors attached to a human (the programmer), his/her movements were accurately captured and translated into programming instructions. The latter were then transmitted to the robot, which was then able to reproduce exactly the human movements.

The content of the chapter then evolved into the description of two developed environments oriented to human-robot safe collaboration. One consisted in the development of a virtual-reality environment to improve, for example, workspace layout design, and the second consisted in the development of an augmented-reality environment that increases safety during real-time operation. Both VR and AR environments are intended to assure a maximum-safety human-robot collaboration. Our system also features an open and scalable architecture. It can easily accommodate further developments and/or the integration of new modules (for example, a motion prediction module).

It is important to highlight that for the real-time robot operation (AR system), the goal of our approach is not to fully replace other traditionally used security measures such as safety switches and low-power light beams. The purpose is rather to add an effective security layer for maximum operator safety and to reduce the number of events in which production should the stopped to handle safety issues.

The developed system also includes the following features: 1) Hazardous areas are shown directly on the scene, as can be seen in Fig. 2.10. This should reduce the risks of the operator walking into a protected area. 2) A *safety score* (Figs. 2.10 and 2.11), which is based on the combination of multiple inputs, such as: a) the operator's distance to the robot, b) the safety level of the current robot's task, c) the operator's attention (e.g., tracking the operator's eyes) relative to the robot, and d) the operator's (next/imminent) movement prediction. The value of the *safety score* is used to warn the operator by using audio and visuals cues (when the score is too low). 3) The system can also be equipped with a signal to communicate with the robot to slow it down or stop it when the operator gets too close. We consider that our solution may lead to increased productivity, as it provides a common User Interface (UI) directly in front of the robot, which allows the operator to manage tasks and eventually stop the robot. Additionally, when the robot is stopped, the operator can jog the robot from a safe distance, without touching it, simply by moving 3D spheres around in augmented reality.

References

[1]. International Federation of Robotics, https://www.worldrobotics.org

[2]. A. Rizzotti-Kaddouri, L. Jeanneret, B. Studer, J. Bracamonte, N. Ouerhani, M. Kunze, Learning from Demonstration and Safe Cobotics Using Digital Twins, *Sensors & Transducers,* Vol. 261, Issue 2, 2023, pp. 25-32.

[3]. J. Lee, A survey of robot learning from demonstrations for human-robot collaboration, *arXiv Preprint*, 2017, arXiv:1710.08789.

[4]. Z. Zhu, H. Hu, Robot Learning from Demonstration in Robotic Assembly: A Survey, *Robotics,* Vol. 7, Issue 2, 2018, 17.

[5]. J.-T. Masse, Capture de mouvements humains par capteurs RGB-D, PhD Thesis, *Université Paul Sabatier-Toulouse III,* 2015.

[6]. K. Bernardin, K. Ogawara, K. Ikeuchi, A sensor fusion approach for recognizing continuous human grasping sequences using hidden Markov models, *IEEE Transactions on Robotics,* Vol. 21, Issue 1, 2005, pp. 47-57.

[7]. Y. Lan, J. Li, Z. Ju, Data fusion-based real-time hand gesture recognition with Kinect V2, in *Proceedings of the 9th International Conference on Human System Interactions,* Portsmouth, United Kingdom, 2016, pp. 307-310.

[8]. Z. Zhu, H. Hu, D. Gu, Robot Performing Peg-in-Hole Operations by Learning from Human Demonstration, in *Proceedings of the 10th Computer Science and Electronic Engineering Conference (CEEC'18),* 2018, pp. 30-35.

[9]. M. Koskinopoulou, S. Piperakis, P. Trahanias, Learning from Demonstration facilitates Human-Robot Collaborative task execution, in *Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI'16),* 2016, pp. 59-66.

[10]. D. Xu, S. Nair, Y. Zhu, Neural task programming: Learning to generalize across hierarchical tasks, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'18)*, 2018, pp. 3795-3802.

[11]. M. Stenmark, E. A. Topp, From demonstrations to skills for high-level programming of industrial robots, in *Proceedings of the AAAI Fall Symposium*, 2016, pp. 75-78.

[12]. ISO/DIS 10218-1 Robotics – Safety Requirements, https://www.iso.org/standard/73933.html

[13]. ISO/TR 20218-1:2018 Robotics – Safety Design for Industrial Robot Systems, https://www.iso.org/fr/standard/69488.html

[14]. ISO/TS 15066:2016 Robots and Robotic Devices – Collaborative Robots, https://www.iso.org/standard/62996.html

[15]. F. Vicentini, Terminology in safety of collaborative robotics, *Robotics and Computer-Integrated Manufacturing*, Vol. 63, 2020, 101921.

[16]. R.-J. Halme, et al., Review of vision-based safety systems for human-robot collaboration, *Procedia CIRP,* Vol. 72, 2018, pp. 111-116.

[17]. H. Ravichandar, et al., Recent advances in robot learning from demonstration, *Annual Review of Control, Robotics, and Autonomous Systems*, Vol. 3, 2020, pp. 297-330.

[18]. S. E. Ovur, et al., Novel Adaptive Sensor Fusion Methodology for Hand Pose Estimation With Multileap Motion, *IEEE Transactions on Instrumentation and Measurement*, Vol. 70, 2021, pp. 1-8.

[19]. HTC, Vive Tracker Sensor, https://www.vive.com/us/accessory/tracker3/

[20]. I. Soares, et al., Accuracy and repeatability tests on HoloLens 2 and HTC Vive, *Multimodal Technologies and Interaction*, Vol. 5, Issue 8, 2021, pp. 47-62.

[21]. Automatic Planning and Scheduling, https://en.wikipedia.org/wiki/Automated_planning_and_scheduling#Classical_planning

[22]. STRIPS, https://en.wikipedia.org/wiki/Stanford_Research_Institute_Problem_Solver

[23]. M. Faroni, et al., A layered control approach to human aware task and motion planning for human-robot collaboration, in *Proceedings of the 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN'20),* 2020, pp. 1204-1210.

[24]. Microsoft HoloLens 2, https://en.wikipedia.org/wiki/HoloLens_2

[25]. HTC, Vive Pro Eye, https://www.vive.com

[26]. Unity Technologies, https://unity.com

[27]. E. Prati, et al., How to include User eXperience in the design of Human-Robot Interaction, *Robotics and Computer-Integrated Manufacturing*, Vol. 68, 2021, 102072.

[28]. System Usability Scale (SUS), https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html

Chapter 3

Robust Controller Design for Nonlinear Hemispherical Tank System

P. Madhavasarma, P. Veeraragavan and M. Sridevi

3.1. Introduction

Recent times in process control industry and biochemical industry most of the process are nonlinear in nature so in this industry the design of control system depending on the process model. Control system design for most of the real processes is challenging task because processes are non-linear and the non-linear models are linearised so that they can be used in linear design methodologies. Even if a linear model is accurate, the parameters of the model could be time varying and could change due to change in operating conditions. Thus, the linear models used in the controller design techniques are normally not very accurate and have uncertainties. Controller design performance will be based on nonlinear region is linearised and design the controller. So due to change in time the model parameters are varying in nature and have uncertainty. Normally nonlinear model cannot identify easily but due to shape and size the model may be predicted as a nonlinear model. In the real time applications such a type of nonlinear model are hemispherical, spherical, conical in shape the afro mentioned vessels may be have a dead time and process delay and bye pass flows [1-3]. In conventional method of the design of controller process to be consider as a nominal operating point in that situation the inlet flow rate and volume of the system assumed to be constant but the assumption is not always possible for all condition because external disturbance, fluid flow rate, production rate through the system inlet is not always constant. Hence ARMAX method of identification is proposed. The problem then is to determine a way to

P. Madhavasarma
SASTRA Deemed University Thanjavur, 613401, Tamilnadu, India

describe these uncertainties, so that they may be taken into account during control design and analysis. This is done by imposing:

- Bounds on the parameters of linear models;
- Bounds in the frequency response.

Of these two approaches, the second approach in which the uncertainties are modeled by defining the *bounds on frequency response* has received favourable response in the control studies. In this approach, the dynamic behaviour of the plant is modeled by only two parameters namely '*amplitude ratio*' and '*phase shift*'. Thus, given a nominal description (model) of a process, bounds on amplitude ratio and phase shift can be cited at each frequency over a range of frequencies. [4] Recent times there has been more demand for H-infinity controller design. In H_2 procedure the stability and performance of the controller cannot be guaranteed in the presence of model uncertainties.

For design of control system using H^{∞} method gives advanced version with compare to conventional control methods [5] For find a stabilizing controller F(S) for a given augmented plant P(s) that minimizes the norm of the cost function to be lesser than unity. The principal advantages of H^{∞} control strategy include (i) it provides robust stability to structural uncertainties. (ii) it achieves performance requirements efficiently. (iii) it not only works on SISO (Single Input Single Output)systems but also for MIMO(Multi Input Multi Output) systems and takes care of disturbance. Therefore, frequency response criterion can be easily shaped to desired specifications. Some of the authors are identified using graphical response for obtaining the FOPDT and SOPDT model parameters [6] and the parameters of higher order and delayed models. Lubyan et al. [7] proposed a new method, where a LS method is applied to give a high-order dead-time free transfer function from the recorded process input and output. From this high-order model, a FOPDT/SOPDT model is obtained with model reduction techniques. However, such a method may result in instability. The concept of multiple model approach is mainly used in nonlinear systems modeling and control. This method is used to estimate a number of process model for different time delay. The model parameter chosen minimize a cost function that depends on the difference between the process model and output. Parket et al [8] have designed a multiple model strategy for robust damping of low-frequency electromechanical oscillation in an interconnected power system. Rao and Siva Kumar [9-14] suggested the iterative shift algorithm to choose the time delay. In their algorithm process parameters are estimated for a set of time delays in a certain range and a predefined cost function is calculated for every set of estimated parameters. Madhavasarma et al have designed the controller for nonlinear systems using soft controllers as well as H-infinity controller [15-18]. The aim of this work is to experimentally study by open loop analysis a nonlinear process represented by a hemispherical tank. Level is the parameters on which attention is focused. Based on open loop analysis models are to be generated and H infinity controllers were designed by simulation and its performance evaluated.

3.2. Modeling of Hemispherical System

In hemispherical tank level should be maintained constant [8-20] For better result inlet flow rate to be controlled so controlled variable for this system is level of the tank and manipulated variable is the inlet flow to system. The schematic diagram of the system is as shown in Fig. 3.1.

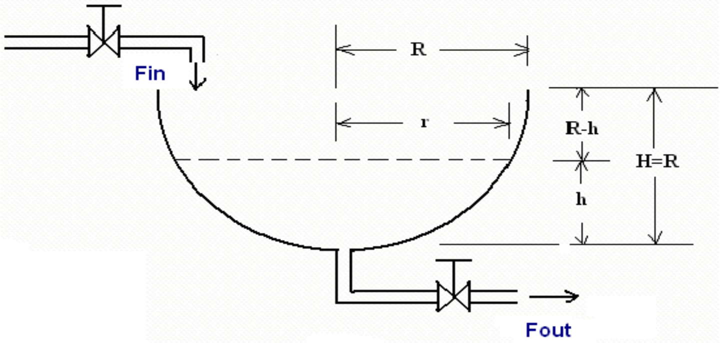


Fig. 3.1. Level control in hemispherical tank.

Let q_{out} and q_{in} be the changes in outflow rate and inflow rate in cm³/sec.

Let R be the top radius of the tank in cm. Let H be the total length of the tank in cm.

Let r be the radius at nominal height h in cm. using the law of conservation of mass given in Eq. (3.1)

$$q_{in} - q_{out} = \frac{dv}{dt}, \tag{3.1}$$

where V is the Volume of the hemi-spherical tank = $\frac{1}{6}\pi h(3r^2 + h^2)$.

From the Pythagoras theorem,

$$R^2 = r^2 + (r - h)^2 \tag{3.2}$$

Now putting the value of r in equation (3.2), then $V = \pi[Rh^2 - \frac{1}{3}h^3]$.

The outflow rate is proportional to the square root of height of the liquid.

$$q_{out} = ch^{(1/2)},$$

$$\frac{d\pi[Rh^2 - \frac{1}{3}h^3]}{dt} = q_{in} - ch^{1/2}, \tag{3.3}$$

$\frac{\pi[2Rh - h^2]dh}{dt} = q_{in} - ch^{1/2}$, Let us take $q_{in} = Q_{in}$,

$$\frac{dh}{dt} = \frac{-c\sqrt{h}}{[\pi(2Rh - h^2)]} + \frac{Q_{in}}{[\pi(2Rh - h^2)]},$$

$$\frac{dh}{dt} = f(h, Q_{in})$$

Using a truncated Taylor series Expansion (3.4)

$$\frac{dh}{dt} = f(h_s, Q_s) + \frac{\partial f}{\partial h} \mid (h_s, Q_1)(h - h_s + \frac{\partial f}{\partial Q} \mid (h_s - Q_s)(Q - Q_s) +$$

$+\frac{1}{2}\frac{\partial^2 f}{\partial h^2} \mid (h_s, Q_s)(h - h_s)^2 + \ldots$. Neglecting the higher terms, where h_s and Q_s are steady state height of level and input flow.

$$\frac{dh}{dt}=\frac{-c\sqrt{h}}{[\pi(2Rh-h^2)]}+\frac{Q_{in}}{[\pi(2Rh-h^2)]}\text{-+}$$
$$+\frac{(Q-Q_1)}{\{\pi(2Rh_s-h^2)\}}-\frac{Q_s(2R-2h)(h-h_s)}{\pi(2Rh_s-h_s)^2}$$

In the above equation, the first term of the right hand side term will be zero, since the linearization is going to be done at steady-state point, hence we will get

$$\frac{d(h_s-h)}{dt}=+\frac{Q_{in}}{[\pi(2Rh-h^2)]}\text{-+}$$
$$+\frac{\{(h-h_s)[\{(2Rh_s-h_s{}^2)2C\sqrt{h_s}\}-2C\sqrt{h}(R-h)]\}}{(2Rh_s-h_s{}^2)^2}-\frac{Q_s(2R-2h)(h-h_s)}{\pi(2Rh_s-h_s)^2}$$

$$\frac{dh}{dt}=-h_a+bQ_{in},\qquad(3.5)$$

where a = $\acute{c}/\{2\sqrt{h_s}(2R-h_s)^2\}+Q(R-h_s)/\{\pi(2Rh_s-h_s^2)^2\}$ and b = $1/\{\pi(2Rh_s-h_s^2)$.

After taking Laplace Transformation of equation (3.5), we get

$$Sh(s)=-h(s)a+bQ(s),\qquad(3.6)$$

$$h(s)(s+a)=bQ(s)hence,$$

$$\frac{h(s)}{Q(s)}=\frac{b}{s+a},$$

$\dfrac{h(s)}{Q(s)}$ = h(s)/Q(s) = $[2h_s^{(5/2)}/\{(2Rh_s-h_s^2)((\pi\acute{c}+4Q_s(R-h_s)h_s^{(5/2)}\}/$

$/[s\{2\pi h/(\pi\acute{c}+4Q_s(R-h_s)\,h^{(5/2)}\}+1],$ (3.7)

where $K=\frac{1}{c^{'}}, T=\frac{D}{c'}, C'=\frac{C}{2\sqrt{h}}$, C = co-efficient of valve output (0.5 to 1), D = $2X3.14Rh-3.14h^2$.

3.2.1. Experimental

The experimental setup for determining the process model is shown in the Fig. 3.2. A fresh water are fed to a 40 liter hemispherical spherical tank through two Gallen Kamp rotameters. The level was monitored using online Honeywell level sensor. The hemi spherical tank process is divided into three levels (L1, L2, L3) longitudinally and various transportation lag is realized in each region using valves. The level sensor output is interfaced to a PC using real time data acquisition card from M/S AD Instruments. The card can be connected directly to the USB port of the computer. It has in built anti-aliasing filter. The card supports 16 ADC and DAC channels with voltage range of ±15 volt. The conversion speed of the card is 200000 samples per second with 16 bit resolution. Dynamic behavior of the system was studied by using a computer system with suitable interface was connected to measure the level of the hemispherical process.

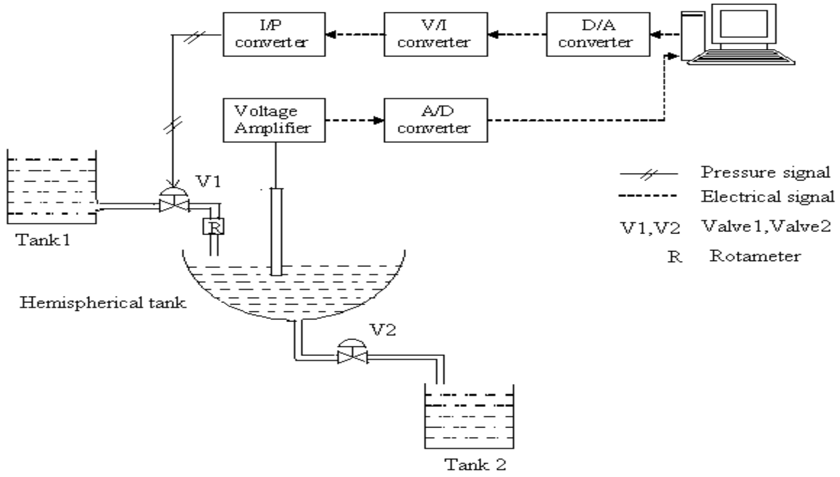


Fig. 3.2. Experimental setup for the nonlinear hemispherical level process.

3.2.2. System Identification Using ARMAX Method

Model of the system was developed based on the data obtained from the open loop response of the hemispherical process. Model was Identified using MATLAB software. Model Identification of the system was done using Auto Regressive Moving Average Extended input (ARMAX) technique [21]. The identified model output was verified with actual process output. The actual output and predicted out is shown in Fig. 3.3. The identified transfer function is given in the Eq. (3.8).

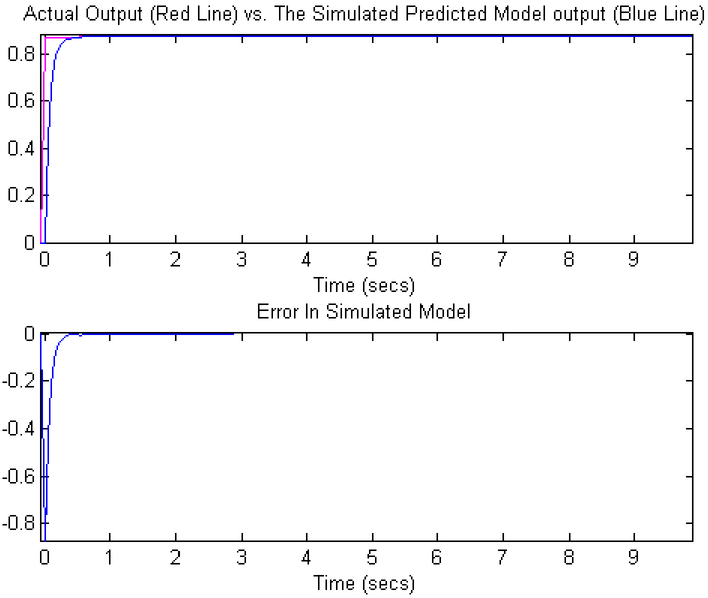


Fig. 3.3. Hemispherical system identification.

The Transfer function of the system is given in Eq. (3.8).

69.31 s^2 + 1.355e005 s + 7.308e007, (3.8)

s^3 + 2025 s^2 + 1.19e006 s + 7.307e007

3.3. PI Controller Design

Nonlinear hemispherical system represented as plant and controller proportional gain is marked as K_p it has been depicted in the form of block diagram which is shown in Fig. 3.4. The PI controller tuning settings values obtained using ZN tuning rules. The designed controller was run for servo problem in the MATLAB software. The response of the PI controller is shown in Fig. 3.5. The controller settings is given in Table 3.1.

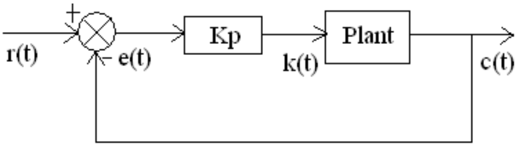


Fig. 3.4. Schematic block diagram for PI controller.

Table 3.1. PI controller settings For hemispherical system.

slo	Kc	Ki
1	0.485	17.2

The controller output for the PI controller is shown in Fig. 3.5. The set point value and output controller response not matched for the set point 20 cm.

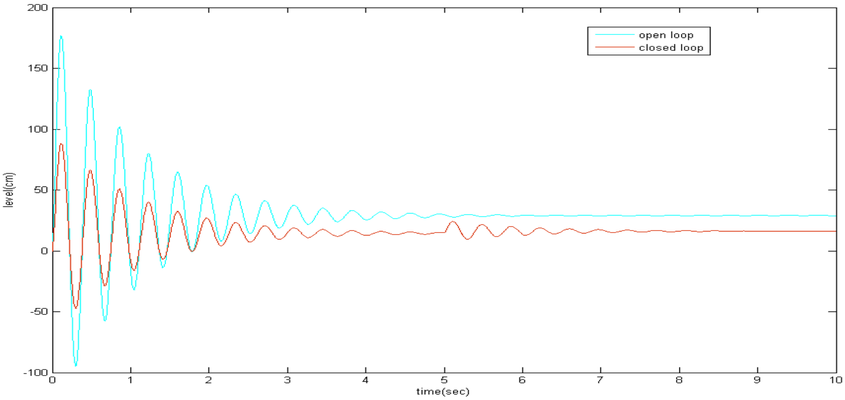


Fig. 3.5. PI controller output response for nonlinear hemispherical system.

3.3.1. Intelligent Control

3.3.1.1. Fuzzy PI Controller

Conventional fuzzy controllers always introduce error and error difference as two inputs without regard to the integral aspect. In order to eliminate steady state errors and improve dynamic control results integral action is added to the conventional fuzzy control system, the fuzzy PI controller combines the advantages of fuzzy control while maintaining the simplicity and robustness of conventional PI controller. The block diagram of fuzzy PI controller is shown in Fig. 3.6. The fuzzy logic controller uses two inputs namely, E (Error), CE (Change in Error) and a single output based on the rule given in Table 3.2 and the member ship function shown in Fig. 3.7(a) to Fig. 3.7(c). The fuzzy PI controller uses the output of the ordinary fuzzy controller to process the proportional and integral operations. The controller is fine-tuned by adjusting controller gains. The tuning parameters are given in Table 3.3. The choice of the parameters is to produce optimal results in simulation.

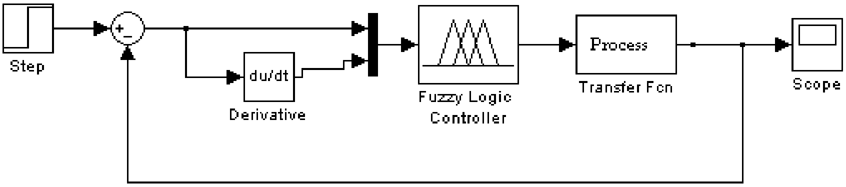


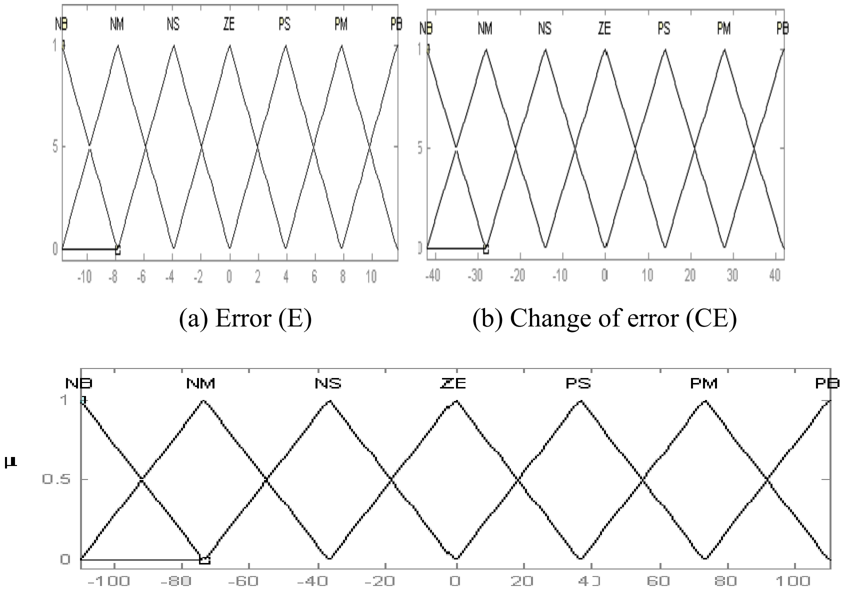
Fig. 3.6. Fuzzy PI control block diagram.

Table 3.2. Rule base of the level controller.

E/CE	NB	NH	NS	ZE	PS	PH	PB
NB	NB	NB	NB	NB	NH	NS	ZE
NH	NB	NB	NB	NH	NS	ZE	PS
NS	NB	NB	NH	NS	NE	PS	PH
ZE	NB	NH	NS	ZE	PS	PH	PH
PS	NH	NS	ZE	PS	PH	PH	PH
PH	NS	ZE	PS	PH	PH	PH	PH
PB	ZE	PS	PH	PB	PH	PH	PH

Table 3.3. Tuning parameters for hemispherical tank process.

S. No	**Kc**	τ_i
1	8	0.33



(a) Error (E)

(b) Change of error (CE)

(c) Controller output (CO)

Fig. 3.7. (a)-(c) Membership functions for error, change of error, controller output.

3.3.1.2. Servo Regulatory Response Analysis

To evaluate the closed loop performance of the fuzzy PI controllers for the hemispherical process, step change in set point and disturbance are introduced.

Servo response to set point change for a step input of magnitude 6 units for the hemispherical process response is shown in Figs. 3.8 - 3.10 and the comparison of controller performance are given in Table 3.4.

3.3.2. Neural Networks Based Controller Design

The structure of neural model and neural model predictive controller is shown in Fig. 3.11. When the network is trained, it can be used in the recall mode where the network weights are fixed, and it is tested with a single presentation of data set. This testing or validation of a network is very important step in the development cycle of non-linear neural model. Identification of the process data was performed using neural network algorithm. The neural model network process consists of three operational steps: prediction, correction and control move determination. In this work water flow rate was input and our put variable and level was the controlled variable. A sampling time of 15 seconds was used for the simulation. For training the neural model step response data of the process was taken. A total of 4000 data were taken continuously and it was saved in file. By training the input output data the NN model of the nonlinear process was obtained. The neural network used for training consists of 2 neurons in the input layer, 1 neuron in the output layer and 9 neurons in the hidden layer. The back propagation through time (BPTT) algorithm was used for training the recurrent network. Neural model was designed for the prediction horizon 2 and control horizon 3 using trained input-output data. Fig. 3.12 shows the training performance of the NN and Fig. 3.13 shows the validation response for optimum alpha value of 0.05 for the hemispherical process. For the network training and validation, the Levenberg-Marquardt back propagation algorithm was used. The convergence criterion was selected as 10^{-3}, and this was achieved in 18 and 65 epochs.

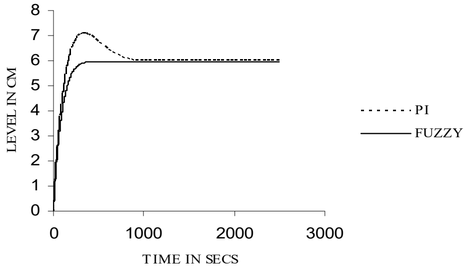


Fig. 3.8. Comparison of servo responses of the controllers to set point change of 6 units.

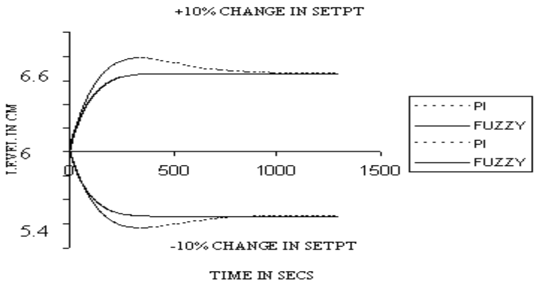


Fig. 3.9. Comparison of level response for a ± 10 % variation in set point.

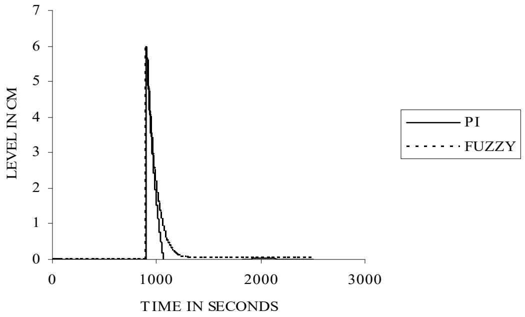


Fig. 3.10. Comparison of regulatory response of the controllers to a load disturbance of 6 units.

Table 3.4. Comparison between PI and Fuzzy controller.

Tuning Method	**Peak Overshoot**	**Rise Time**	**Settling Time**	**Load ISE**	**Servo ISE**	**Decay ratio**
PI	35	110	300	14750	15832	0.1225
FUZZY	0	30	270	2500	4079.1	0

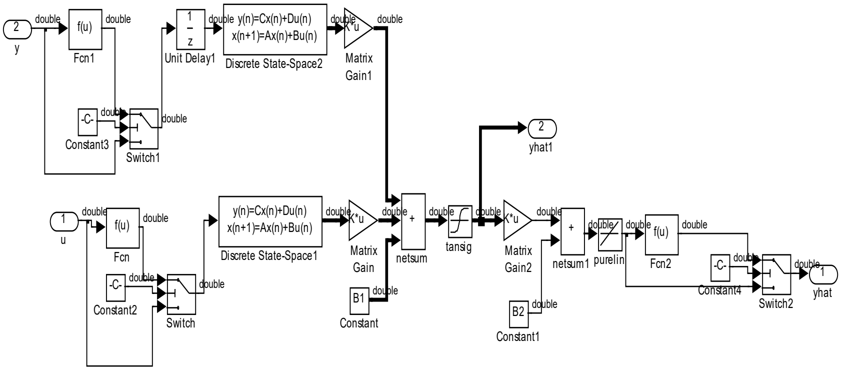


Fig. 3.11. Structure of NN Model.

3.3.3. Robust Control Design

Robust control design is to ensure that the controller will *stabilize* the plant and will achieve the *desired performance* in the presence of plant-model mismatch (uncertainties). Robust stability and robust performance properties can be verified once process-model is match has been quantified [22-24].

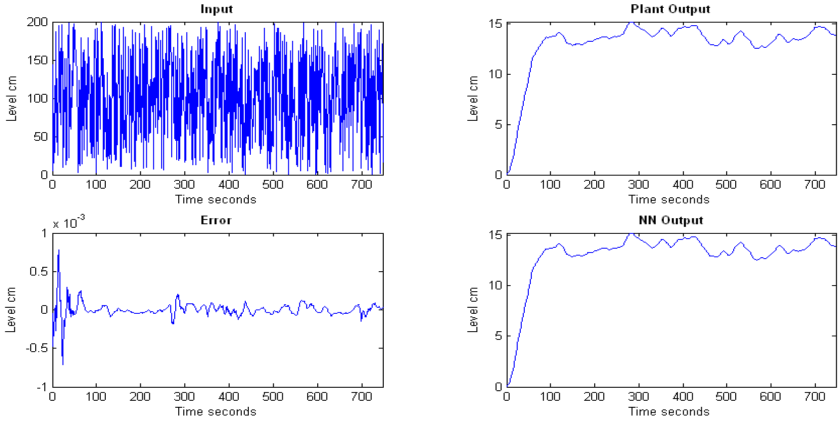


Fig. 3.12. Training of the NN model with process for optimum alpha value 0.05 for hemispherical process.

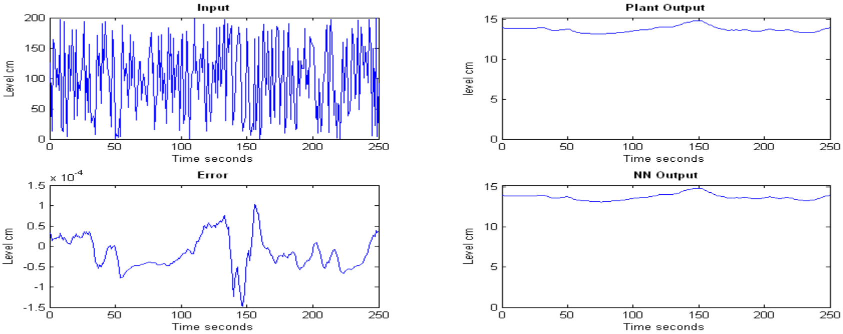


Fig. 3.13. The validation response for optimum alpha value of 0.05 for the hemispherical process.

3.3.3.1. Uncertainty Disk

In any system the nature of the uncertainty area explained by phase shift bounds and amplitude ratio. This will be explained in the form of disk. Assume disk will be only one parameter its radius, $\tilde{l}_a(\omega)$, which is a function of frequency. so, every change frequency value there is a corresponding disk so it directly implies that $\tilde{l}_a(\omega)$ will increase with frequency, it means when process under steady state it will be very easy to characterize.

3.3.3.2. Sensitivity Functions

For Design of controller vital point is the maintain the error value as small as possible. The plant run as closed loop condition it will be affected by some external signals. For analyzing the performance of the controller this external signal should be quantity than only controller gives correct error one such a quantifying measure is the sensitivity function. Sensitivity related to outcome of disturbance on the process output and complementary sensitivity function related to consequence of set point on the process output.

3.3.3.3. Robust Stability

Assume that all plants in the set

$$\Im = \left\{ G_p : \frac{\left| G_p(j\omega) - \tilde{G}_p(j\omega) \right|}{\left| \tilde{G}_p(j\omega) \right|} \le \tilde{l}_m(j\omega) \right\}$$

have the same number of RHP poles, and that the controller $G_C(s)$ stabilizes the model $\tilde{G}_p(j\omega)$. Then $G_C(s)$ will provide robustly stable control if, and only if, the nominal complementary sensitivity function $\tilde{\eta}$ satisfies:

$$\left\| \tilde{\eta}\tilde{l}_m \right\| \equiv \overset{Sup}{\underset{\omega}{}} \left| \tilde{\eta}(j\omega)\tilde{l}_m(j\omega) \right| < 1$$

This implies that

$$\left| G_C \tilde{G}_P \right| < \frac{1}{\tilde{l}_m(\omega)} \text{ and } \left| G_C \right| < \frac{1}{\tilde{l}_a(\omega)}$$

This means that the amplitude ratio of the controller at high frequencies must be constrained by the above inequalities in order to maintain robust stability conditions and these constraints are governed by the magnitude of the uncertainty bounds.

3.3.3.4. Robust Performance

Robust stability is the minimum requirement of any practical control system. However, even though a closed loop is robustly stable, it will not be of any use if it does not deliver the required performance. In frequency domain, the $H-\infty$ control problem is stated as:

$$\overset{\min}{G_C}\ \overset{\max}{G_P} \in \Im \underset{\omega}{\overset{\sup}{}} \left|\varepsilon(j\omega)q(j\omega)\right| < 1$$

The nomenclature $\overset{\max}{G_P} \in \Im$ denotes the worst case plant in the set $\Im$.

Re-arrangement of the $H-\infty$ control performance equation gives:

$$\left|\tilde{\eta}(j\omega)\right|\tilde{l}_m(\omega) + \left|\tilde{\varepsilon}(j\omega)q(j\omega)\right| < 1 \ \ \forall\,\omega$$

Control makes use of the $H-\infty$ and norm to quantify both performance stability, thus allowing the tradeoff between performance and stability to be explicitly stated. In this sense, $H-\infty$ control can be considered to be a better framework for the design of robust control systems.

3.3.3.5. Design of H-infinity Controller

The H^{∞} controller using Mixed Sensitivity approach is to be used for the control formulation is such that the controller parameters are tuned to account for plant changes or varying performance requirements. Here the controller is developed for the plant augmented with weighing functions. Choice of weighing functions is a critical problem as no general guidelines exist for the proper selection of weighing functions. The H-inf controller designed [22] and implemented for the hemispherical system model ensures robustness and also guarantees stability and performance.

3.4. Result and Discussions

For Design of H infinity controller initial step is selection of weighting function. The selected weighting function must obey the singular loop specifications and stability margin. Various trial and error methods to be involved for selection of this function. The selection of the weighting function is the important accept in H-infinity controller design. The representation of this function for various signal as such as first one is the error signal that will be denoted as w$_{1,}$ The second one is the control signal it is marked as w2. The last one is output signal that will be denoted as, w3 respectively. The weighting function w$_1$ $_{\text{is}}$ used to reshape the frequency response characteristics and is chosen in such a way that w$_1$ act as a low pass filter with 10 rad/sec as cut off frequency. There is no weightage given to the control signal. The weighting function w$_3$ is decided based on the multiplicative uncertainty present in the plant. The weighting functions for the Plant is given in Eq. (3.9), (3.10) and (3.11)

$$w_1 = \frac{16.47s + 1266}{s^2 + 612.3s + 7774}, \tag{3.9}$$

$$w_2 = 1, \tag{3.10}$$

$$w_3 = \frac{S^4 + 23.5s^3 + 12.8S^2 + 1250}{s^4 + 345s^3 + 22s^2 + 6456} \tag{3.11}$$

The H-inf controller for the identified hemispherical system is obtained and it is given in the Eq. (3.12)

$$F(s) = \begin{bmatrix} -3.1760 & 239.0253 & 0.0552 & -0.7667 & 0.0011 \\ -160.4203 & -3.1763 & -0.2562 & -0.0046 & -0.002 \\ 3.5205 & -2.6571 & -599.28 & 191.59 & -5.2468 \\ -0.2384 & 0.1797 & 0.1295 & -13.01 & 0.3564 \\ -0.0091 & -0.0014 & 0.0112 & -0.3121 & 0 \end{bmatrix} \tag{3.12}$$

The H-inf controller obtained after performing model reduction is given in Eq. (3.13)

$$F(s) = \begin{bmatrix} -12.98 & 0.28 & -0.0518 & 0.3313 \\ -0.2801 & -6.2403 & 195.7742 & 0.0036 \\ -0.0518 & -195.7742 & -0.2165 & 0.0007 \\ -0.3313 & 0.0036 & 0.0007 & 0 \end{bmatrix} \tag{3.13}$$

The H infinity controller output for different set point 1 cm and 25 cm respectively is shown in Fig. 3.14 and Fig. 3.15. From that it was infers that H infinity controller reaches the set point in faster manner and there is no sluggish and uncertainty in the process. Fig. 3.16 shows the closed loop comparison response for the hemispherical system.

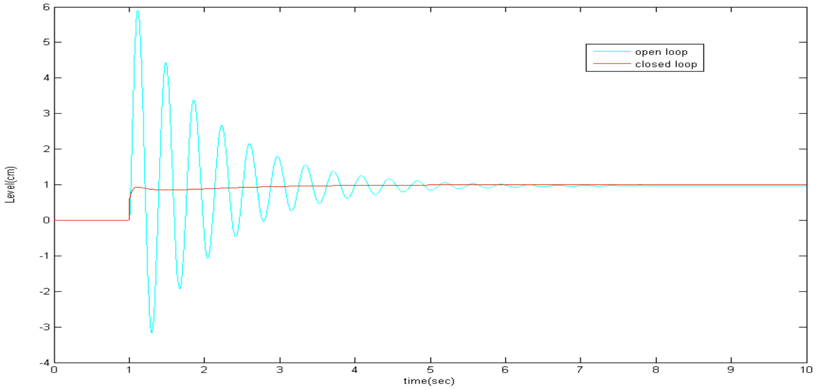


Fig. 3.14. H infinity controller response for set point 1 cm.

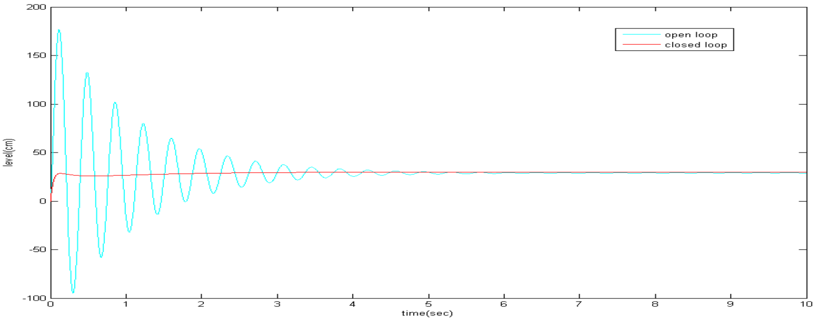


Fig. 3.15. H infinity controller response for set point 25 cm.

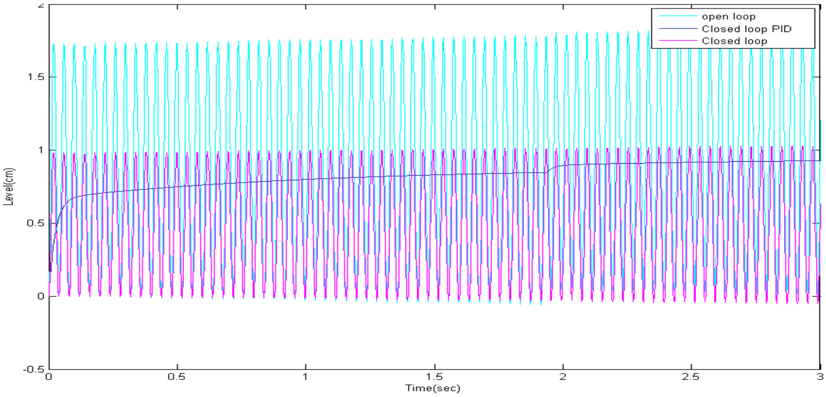


Fig. 3.16. Comparison between PI and H infinity controller Closed loop responses.

The closed loop responses in frequency domain for reduced order model is shown in Fig. 3.17.

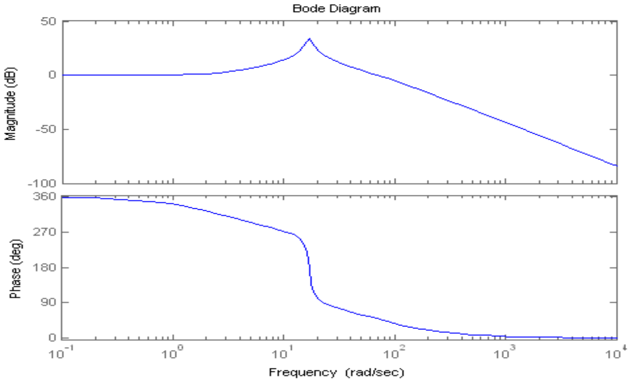


Fig. 3.17. The closed loop response for the frequency domain for reduced order model for the hemispherical system.

3.4.1. Discussions

The results presented in previous section are summarized and relevant conclusions drawn. Models studied for the nonlinear process are summarized and listed in the Eq. (3.14). Model parameters and the average absolute percentage error between the model and the experimental data are also given in the Fig. 3.3. It can be concluded that model represents the processes studied with reasonable accuracy.

Four controllers: PI, Fuzzy, Neural and H infinity controller were simulated using MATLAB for their closed loop performance using MATLAB for the processes indicated in Eq. (3.14) based on the models generated for both servo and regulator problem by a positive and negative step change in flow rate. The parameters assessed for the four controllers are rise time, settling time, overshoot, ISE, IAE (5 parameters) was critically analyzed. and the performance of the system were studied for the four controllers and was ranked as follows based on the above 5 parameters given in the Table 3.4 and Table 3.5: If all the five parameters namely, rise time, settling time, overshoot, ISE, IAE was minimum it is ranked as number 1. In this work for controller design using H-infinity method was considered for the study. Model was identified using ARMAX method based on the obtained data using

MATLAB software. The following transfer function was obtained. It was given in the Eq. (3.14)

$$\begin{array}{c} 69.31 \text{ s^2} + 1.355\text{e}005 \text{ s} + 7.308\text{e}007, \\ \text{s^3} + 2025 \text{ s^2} + 1.19\text{e}006 \text{ s} + 7.307\text{e}007 \end{array} \qquad (3.14)$$

For the identified model H- infinity controller was designed using step response method. The brief discussion about how to choose the singular value loop specifications for the controller was discussed below. In recent times there have been significant advance in modern control theory with H_2 and H^{∞} methods gaining widespread recognition and application. In H_2 procedure the stability and performance of the controller cannot be guaranteed in the presence of model uncertainties. The H^{∞} control strategy as compared to classical control techniques provides an advanced methods and perspective for designing control systems. This is accomplished by shaping the frequency response characteristics of the plant according to prespecified performance specifications in the form of frequency dependent weighting functions. The goal of robust analysis is to find a system was Multivariable Stability Margin (MSM) seen by the uncertainties using a proper nonconservative and analytical tool. The H^{∞} theory provides a direct, reliable procedure for synthesizing a controller which optimally satisfies singular value loop shaping specifications. Here we find a stabilizing controller F(S) for a given augmented plant P(s) that minimizes the norm of the cost function to be lesser than unity. The principal advantages of H^{∞} control strategy include: (i) It provides robust stability to structural uncertainties; (ii) It achieves performance requirements efficiently; (iii) It not only works on SISO (Single Input Single Output)systems but also for MIMO(Multi Input Multi Output) systems and takes care of disturbance. Therefore, frequency response criterion can be easily shaped to desired specifications. The system was tested using step response method it can be noticed form the Fig. 3.14 and Fig. 3.15. The overshoot is very minimum compared to other approaches [25-27]. Similarly, the steady state error is less compared to others methods [25-27]. From the Fig. 3.16 error value of controller is compared with another controller PI is less. The bode plot for the stability of the system is depicted in that Fig. 3.17. It was inferred from the figure due to adding of poles and zeros by weighting filters for shape the system the performance was improved and stability was obtained in good condition. The importance novelty of this work is the H infinity controller design for nonlinear system and selection of the weighting function for control the nonlinear parameters by adjusting the pole zero It is seen from the Table 3.5 Integral square error ISE for our proposed methods and others who work in this area was compared. From that proposed control methods provides the optimal performance.

Table 3.5. ISE Values comparison between proposed methods to others.

slo	Strategy	ISE
1	Diaz et al. [26]	38.4393
2	Da Silva et al. [25]	36.1603
3	Rakesh et al. [27]	85234
4	Proposed method	2342

3.5. Conclusions

The second order model was identified for the hemispherical system using ARMAX methods. H-Infinity controller was designed for the hemispherical system. The closed loop response for the designed controller was obtained using MATLAB software. The results for H-Infinity controller emphasize that the controller shows a minimum dynamic response time than the conventional PI controllers. While a conventional PI controller reaches the set point with small oscillation and reaches smoothly. H-infinity controllers possess a sharp curve like responses. This response is attributed to the non-linearity of the system. Based on the results given in Table 3.4 and Table 3.5 H-infinity controller was best suited for this system.

References

[1]. P. Madhavasarma, S. Sundaram, Model Based Tuning of Controller for Non-Linear Hemispherical Tank Processes, *Instrum. Sci. & Technol.*, Vol. 35, 2007, pp. 681-689.

[2]. P. Madhavasarma, S. Sundaram, Model Based Tuning For a Non-Linear Spherical Tank With Time Delay Process, *Instrum. Sci. & Technol.*, Vol. 36, 2008, pp. 420-431.

[3]. P. Madhavasarma, M. Sridevi, P. Prakasam, Model identification of nonlinear system using soft computing technique, in *Proceedings of the IEEE International Advance Computing Conference (IACC’14)*, 2014, pp. 1174-1178.

[4]. J. C. Doyle, B. A. Francis, A. R. Tannenbaum, Feedback Control Theory, *Macmillan*, New York, 1992.

[5]. W. Levine; M. Athans, On the determination of the optimal constant output feedback gains for linear multivariable systems, *IEEE Transactions on Automatic Control*, Vol. 15, Issue 1, 1970, pp. 44-48.

[6]. D. E. Seborg, T. F. Edgar, D. A. Mellichamp, Process Dynamics and Control, *John Wiley and Sons*, 1989.

[7]. W. L. Luyben, Process Modeling, Simulation and Control for Chemical Engineers, *McGraw-Hill*, New York, 1983.

[8]. S. Park, Y. Lee, PID controller tuning to obtain closed loop response for cascade control systems, *Ind. Engg. Chem.*, Vol. 37, 1998, pp. 1859-1865.

[9]. P. Suganda, P. R. Krishnaswamy, G. P. Rangaish, On-Line Process Identification from Closed-Loop Tests Under PI Control, *Chemical Engineering Research and Design*, Vol. 76, 1998, pp. 451-457.

[10]. S. W. Sung, J. H. Park, I. Lee, Modified relay feedback method, *Ind. Eng. Chem. Res.*, Vol. 34, 1995, pp. 4133-4135.

[11]. W. K. Ho, C. C. Hang, L. S. Cao, Tuning of PID controllers based on gain and phase margin specifications, *Automatica*, Vol. 31, 1995, pp. 497-502.

[12]. B. Chaudhuri, R. Majumder, B. C. Pal, Application of multiple-model adaptive control strategy for robust damping of interarea oscillations in power system, *IEEE Trans. Control System Technology*, Vol. 12, 2004, pp. 727-735.

[13]. G. Rao, L. Sivakumar, Identification of deterministic time-lag systems, *IEEE Trans. Automat. Contr.*, Vol. 21, 1976, pp. 527-529.

[14]. E. Gabay, S. J. Merhav, Identification of linear systems with time-delay operating in a closed loop in the presence of noise, *IEEE Trans. Automat. Contr.*, Vol. 21, 1976, pp. 711-716.

[15]. P. Madhava Sarma, M. Sridevi, P. Prakasam, Model Identification and Agitator Speed Optimization for Bioreactor Process Using Soft Computing Techniques, *Applied Mathematical Sciences*, Vol. 9, Issue 45, 2015, pp. 2197-2210.

[16]. P. Madhavasarma, M. Sridevi, S. Kumaravel, P. Veeraragavan, An electrical stimulation data based model to predict the healing period of fractured limb, Mathematical and Computer Modeling of Dynamical Systems, Vol. 25, Issue 4, 201, pp. 354-375.

[17]. P. Madhavasarma, M. Sridevi, P. Prakasam, Model Identification and control of nonlinear spherical tank process using soft computing method, *Sensors & Transducers Journal*, Vol. 145, Issue 10, 2012, pp. 41-49.

[18]. P. Madhavasarma, S Arul Kumar, P Veeraragavan, Design and Implementation of the Monitoring and Control System for Unified Power Quality Conditioner Using Soft Computing Method, in *Proceedings of the Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM'17)*, 2017, pp. 234-238.

[19]. M. Chidambaram, R. Anandanatarajan, T. Jayasingh, Controller design for non-linear process with dead time variable transformation, in *Proceedings*

of the International Symposium on Process System Engineering and Control, 2003, pp. 223-228.

[20]. P. Madhavasarma, Model identification and control of linear and nonlinear process with transportation lag, PhD Thesis, *Sastra University*, 2009, pp. 152-154.

[21]. P. Madhavasarma, M. Sridevi, P. Veeraragavan, ARMAX Identification and Robust control of Hybrid Non-Linear Systems using Soft Computing Method, *Research Journal of Pharmaceutical, Biological and Chemical Sciences*, Vol. 8, Issue 3, 2017, pp. 1895-1900.

[22]. M. Sridevi, P. Madhavasarma, Model identification and Smart structural vibration Control using H∞ controller, *International Journal on Smart Sensing and Intelligent Systems*, Vol. 3, Issue 4, December 2010, pp. 655-671.

[23]. M. Sridevi, P. Madhavasarma, P. Prakasam, Modeling and Control of Smart Structure Using Soft Computing Method, *Sensors & Transducers Journal*, Vol. 145, Issue 10, October 2012, pp. 10-18.

[24]. P. Madhavasarma, M. Sridevi, P. Veeraragavan, Robust and Artificial Neural Network Controller for Piezoelectric Bonded Beam, *International Journal of Advanced Information Science and Technology*, Vol. 4, Issue 9, 2015, pp. 46-52.

[25]. L. R. da Silva, R. C. C. Flesch, J. E. Normey-Rico, Controlling industrial dead-time systems: When to use a PID or an advanced controller, *ISA Trans.*, Vol. 99, 2020, pp. 339-350.

[26]. J. M. Díaz, S. Dormido, R. Costa-Castelló, The use of interactivity in the controller design: Loop shaping versus closed-loop shaping, *IFAC-PapersOnLine*, Vol. 51, 2018, pp. 334-339.

[27]. S. Matumathi, S. Rakesh Kumar, R. Srimathy, G. Kanimozhi, Design of Robust Controller for Hemi-Spherical Tank System Using Volumetric Observer, *International Journal of Applied Engineering Research*, Vol. 12, Issue 17, 2017, pp. 6477-6481.

Chapter 4

Vehicle Localization Based on MEMS Sensor Data

Takayoshi Yokota

4.1. Introduction

The Global Navigation Satellite System (GNSS) is widely used, but its accuracy may be degraded depending on the reception status of the radio waves from the satellite. Urban areas with high-rise buildings have long been a problem in this regard, but in recent years, although rare, radio interference caused by solar flares or by intentional jamming have occurred in some areas. This has created the need for a new technology that can compensate for the inherent disadvantages of the vehicle position estimation technology, which is highly dependent on GNSS and requires huge maintenance costs. In the light of this background, we developed a localization algorithm for a traveling vehicle based on the characteristics of the road obtained from MEMS sensors including acceleration, angular velocity, and magnetization for x, y, and z axes relying only minimally on GNSS.

4.2. Vehicle Localization Algorithm by MEMS Sensor

Terrain-based localization is an interesting research field that has emerged in recent years [1-12]. We have been working on a vehicle localization algorithm based on MEMS sensors for the past several years [1-6], where accurate location information obtained by high-precision GNSS including RTK-GNSS (u-blox ZED-F9P) [13] as well as acceleration, angular velocity, and geomagnetism information (InvenSense 9250) [14] are collected in advance on an actual road for use as reference data. A test vehicle for evaluating the performance of

Takayoshi Yokota
Tokyo Information Design Professional University, Japan

the localization algorithm is then run to acquire sensor data, which are compared with the above reference data to obtain an accurate location. The outline of our algorithm is shown in Fig. 4.1. Pattern matching is performed on the basis of the normalized cross- correlation function defined as

$$C(i,j) = \frac{\sum_{k=0}^{N-1}(f_{i+j+k}-\bar{f}_{i+j})(g_{i+k}-\bar{g}_i)}{\sqrt{\sum_{k=0}^{N-1}(f_{i+j+k}-\bar{f}_{i+j})^2}\sqrt{\sum_{k=0}^{N-1}(g_{i+k}-\bar{g}_i)^2}}, \qquad (4.1)$$

$$\bar{f}_{i+j} = \frac{1}{N}\sum_{k=0}^{N-1} f_{i+j+k}, \qquad (4.2)$$

$$\bar{g}_i = \frac{1}{N}\sum_{k=0}^{N-1} g_{i+k}, \qquad (4.3)$$

where *i* is the index of evaluation for the vehicle's travel time and *j* is the index of the time lag between two vehicles (reference vehicle and evaluation vehicle). If we can determine this time lag accurately, the location of the evaluation vehicle can be substituted by the location (latitude and longitude) of the reference vehicle at the matched time. The pattern matching can then be achieved by finding the maximum value of similarity of the sensor data with a cross- correlation function calculated over a specific time window (e.g., one second).

4.3. Experiments

4.3.1. Experimental Setup

We evaluated our algorithm on a flat road section (Fig. 4.2) with very little change in the road attributes, where the localization error had been as large as 100 m or more when we tested an earlier version of our method [1, 2]. Moreover, even when we had used an accelerometer and gyro sensor in addition to the barometric sensor, there was no improvement of the position accuracy at the processing interval of 0.1 s. This was due to the influence of aliasing noise caused by the sampling interval being too large.

Therefore, our aim here was to reduce the influence of aliasing noise by performing normalized correlation at 0.02 s (50 Hz). The details of the experimental setup are listed in Table 4.1. The MEMS sensors (Invensense Mpu9250) [14] and Bosch BME280s [15] were installed in the rear seat of the vehicle as shown in Fig. 4.3. We processed the resultant data with the resolution of 50 Hz for the Mpu9250 data. The atmospheric data was used only to identify the flatness of the test course. Fig. 4.4 shows the altitude along the test course.

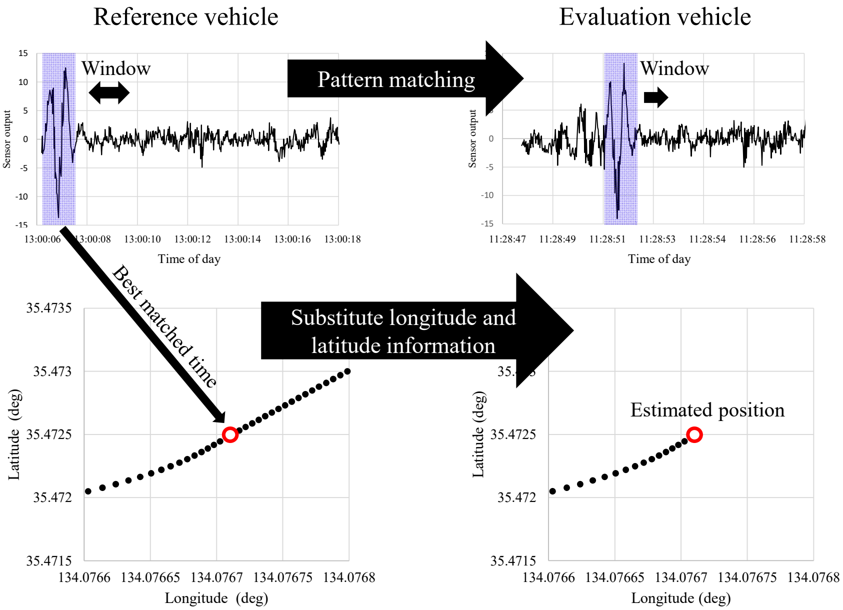


Fig. 4.1. Outline of developed algorithm. The reference vehicle stores sensor data together with RTK-GNSS position data for later reference.

Table 4.1. Experimental setup.

Date	**Dec 5（Sun）,2021, Mar. 6(Sun), 2022**
Vehicle	Daihatsu Mira Gino(Small Car)
MEMS sensor	1. InvenSense MPU-9250 50Hz data acquisition. acceleration, angular velocity, geomagnetism 2. Bosche BME280 20Hz data acquisition atmospheric pressure
RTK-GNSS	u-blox ZED-F9P, 5Hz data acquisition
Test course	1.36km, Shikano-cho, Tottori-shi,Japan

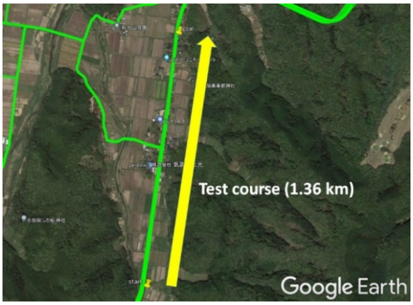


Fig. 4.2. Flat and straight 1.36-km test course along a Grade 2 river.

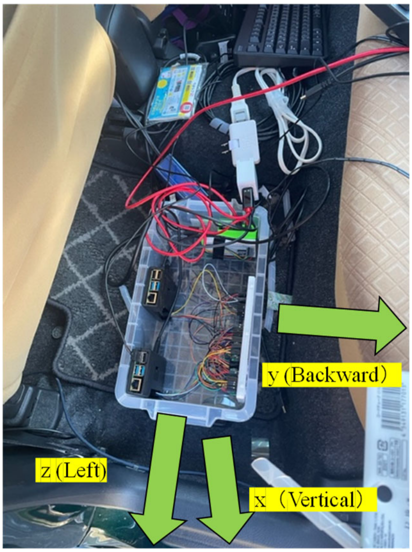
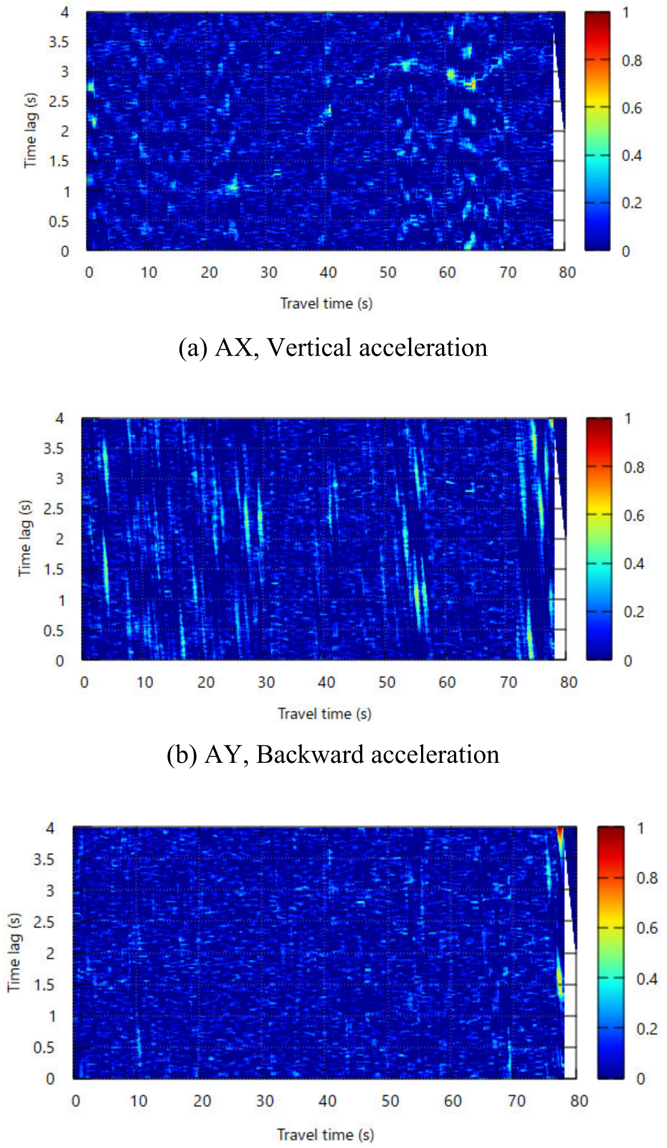


Fig. 4.3. Sensor axes of MPU-9250 are indicated by arrows. Three Raspberry Pi 4Bs are used for data acquisition: one for an MPU-9250 and two for 16 modules of the BME280s. The 16 outputs of BME280s are averaged to reduce random noise.

4.3.2. Pattern Matching by Normalized Cross-correlation Function

Nine normalized cross-correlation functions were obtained for each sensor data (acceleration, angular velocity, and magnetization for x, y, and z axes), as shown in Figs. 4.4, 4.5, and 4.6. Each sensor data of the evaluation vehicle on Mar. 6, 2022 was correlated with the reference vehicle data of Dec. 5, 2021.

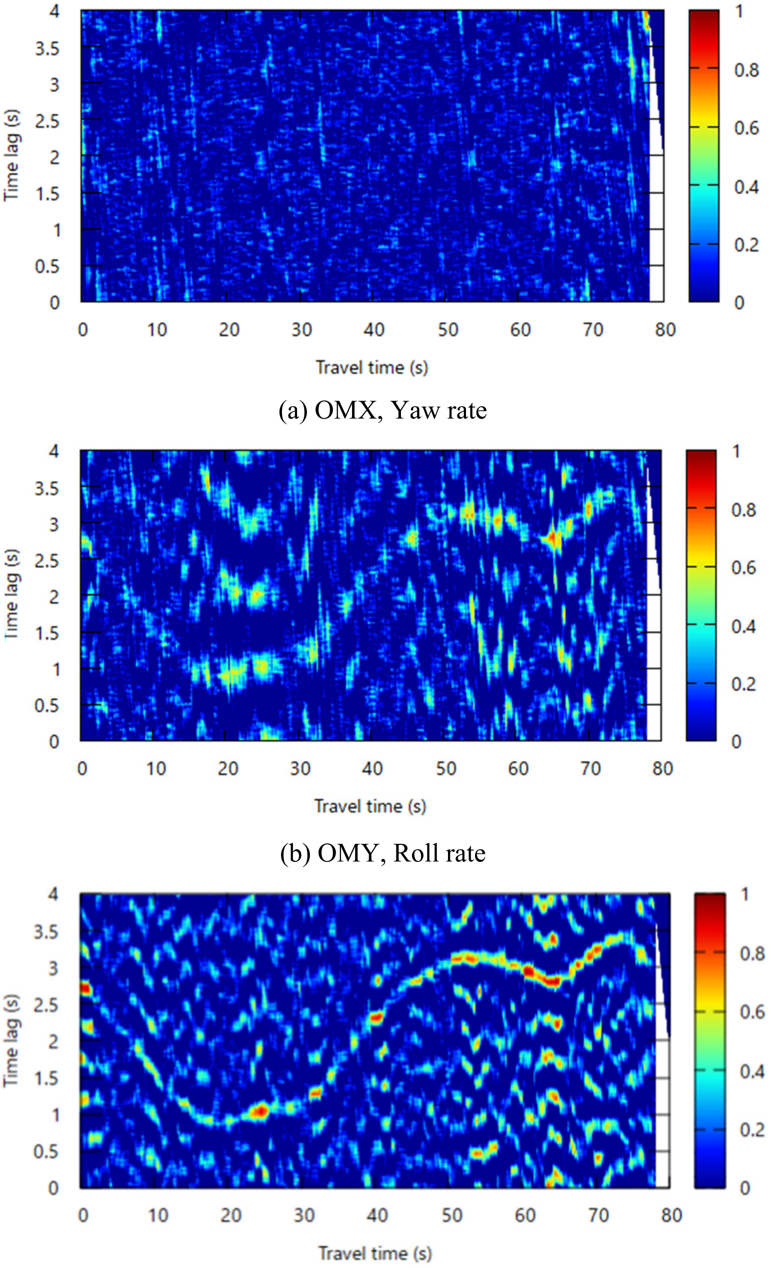


(a) AX, Vertical acceleration

(b) AY, Backward acceleration

(c) AZ, Leftward acceleration

Fig. 4.4. Normalized cross-correlation of acceleration sensor data. Acceleration data toward x, y, and z axes are denoted as ax, ay, and az, respectively.

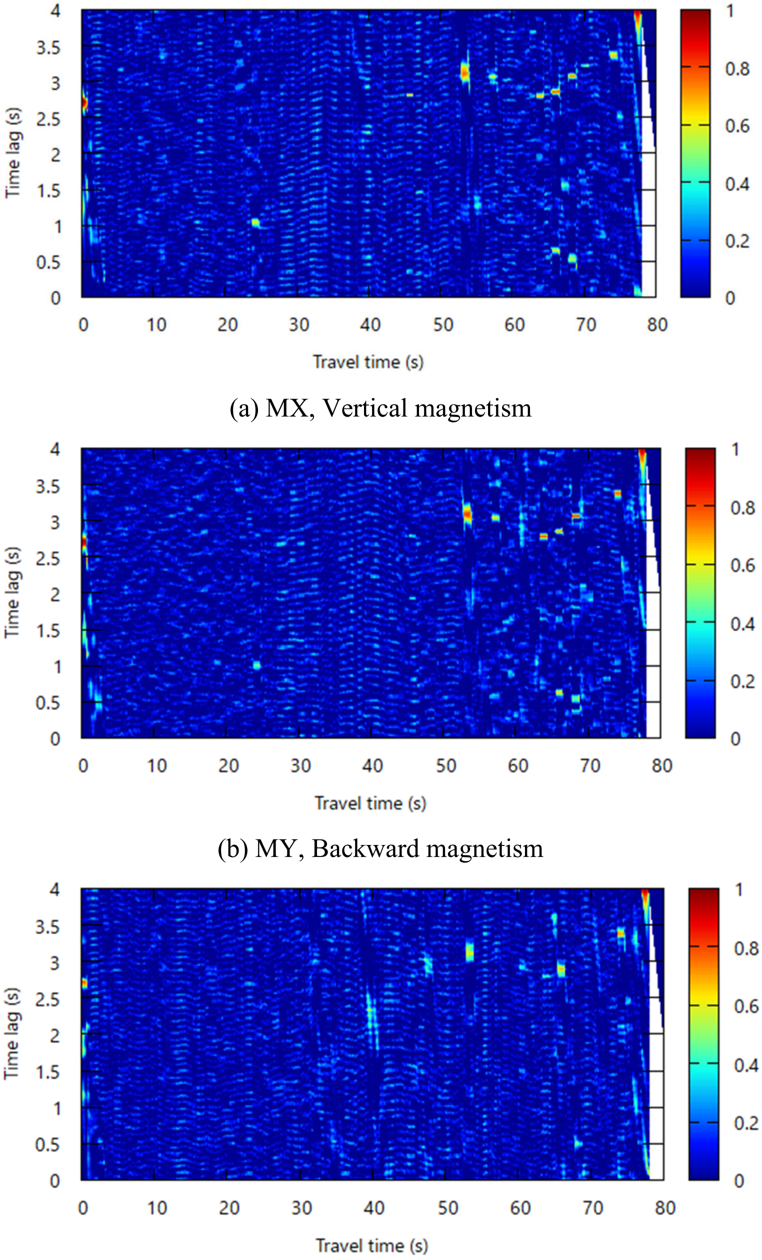


(a) OMX, Yaw rate

(b) OMY, Roll rate

(c) OMZ, Pitch rate

Fig. 4.5. Normalized cross-correlation of gyroscope data. Gyroscope data around x, y, and z axes are denoted as omx, omy and omz, respectively.



(a) MX, Vertical magnetism

(b) MY, Backward magnetism

(c) MZ, Leftward magnetism

Fig. 4.6. Normalized cross-correlation of geomagnetism data. Geomagnetism data along x, y, and z axes are denoted as mx, my, and mz, respectively.

4.3.2.1. Normalized Cross-correlation of Acceleration Sensor Data

Fig. 4.4 shows the results of the normalized cross- correlation of the acceleration data. Among them, a time lag profile can be seen slightly for ax(acceleration for vertical direction).

4.3.2.2. Normalized Cross-correlation of Gyroscope Data

Fig. 4.5 shows the results of the normalized cross-correlation of the gyroscope data. Among them, omz clearly shows a time lag profile. Omy also shows a time lag profile, although its resolution is lower. Gyroscope data around the z axis represents pitch rate and around the y axis represents roll rate. Yaw rate, that is, the rotation rate around the x axis, does not effectively capture the features of roads, and the time lag profile cannot be seen. This is because the test road is an almost completely straight road and thus very little change exists in yaw rate.

4.3.2.3. Normalized Cross-correlation of Geomagnetism Data

Fig. 4.6 shows the results of the normalized cross-correlation of the geomagnetism data. Some portions of the time lag profile are visible here, especially for the latter part from 40 to 80 s of the evaluation vehicle travel time. The resolution is rather high.

4.3.3. Optimum Weight for Normalized Cross-correlation Functions

In the previous sections we demonstrated that we were able to obtain nine normalized cross-correlation functions, some of which clearly show the time lag profile and others that do not. In this section, we explain how to find an optimally weighted average these functions to form an optimum normalized cross-correlation function (Eq. (4.4)). The optimum weights were calculated by a simulated annealing method so that the measure function *F* (defined in Eq. (4.7)) was maximized. *F* measures the signal-to-noise ratio (SNR) of the optimum normalized cross-correlation function of Eq. (4.4). The underlying concept of function *F* comes from the idea that the ratio of the maximum value of correlation at each travel time (x-axis) and the negative minimum value of the cross-correlation at the same travel time (x-axis) must be highest. The minimum value of the cross-correlation is almost always negative, so its absolute value is used as the denominator in Eq. (4.7).

$$R_{opt}(i,j) = \sum_{n=0}^{M-1} w_n R_n(i,j) \tag{4.4}$$

$$w_n \geq 0, n = 0,1,\ldots,8, \tag{4.5}$$

$$\sum_{n=0}^{8} w_n = 1, \tag{4.6}$$

$$F = \frac{1}{M}\sum_{i=0}^{M-1} \frac{max_j R_{opt}(i,j)}{|min_j R_{opt}(i,j)|} \tag{4.7}$$

Unfortunately, the function F is nonlinear, and a simple algebraic solution does not exist. To overcome this, we applied a simulated annealing algorithm in which candidates for the solutions to $w_n \geq 0, n = 0,1,\ldots,8$ are randomly generated iteratively over the course of iterations. Solutions which improves the value of F is favored to survive during the iterations with the randomness. The randomness is called temperature, was gradually decreased. This is similar to the annealing utilized in metal finishing. When the temperature has dropped sufficiently, we can expect to obtain a near optimum solution for the weights $w_n \geq 0, n = 0,1,\ldots,8$. Fig. 4.7 shows the profiles of each value of the weight for the cross-correlation functions.

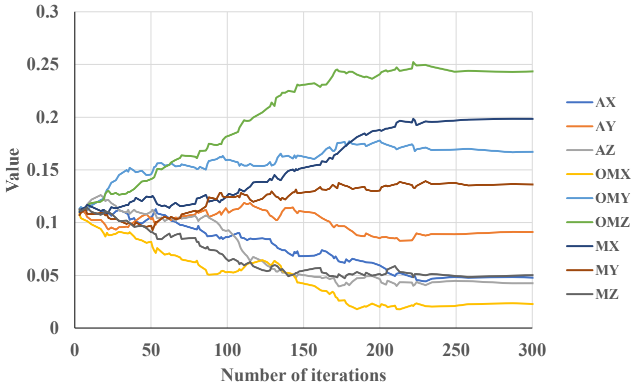


Fig. 4.7. Convergence of optimal weight for nine cross- correlation functions over the course of simulated annealing.

After 300 iterations of the simulated annealing, the weights converged. The range of travel time for evaluating the function F is 20 seconds (M is 1,000 in Eq. (4.7)) of sensor data from a flat road section of the test course (shown in Fig. 4.2). The weights rapidly moved toward convergence values by 250 iterations and at 300, they were almost completely fixed. There were no apparent ambiguities, and each weigh had a solid value depending on its contribution to improving the function *F*. In this case, atmospheric pressure data was not included in the measurements because the test course was almost flat and the altitude data obtained from atmospheric pressure was not considered effective.

Fig. 4.8 shows the optimum weights we derived. As we can see, the pitch rate, vertical magnetism, roll rate, and backward magnetism all play significant roles (74.7 %) in vehicle localization in the proposed algorithm. Fig. 4.9 shows the optimally weighted cross-correlation function, where the time lag profile is clearly recognizable. Fig. 4.10 shows the estimation results of the time lag between the two vehicles. We calculated the ground truth of the time lag by comparing the RTK-GNSS trace of each vehicle.

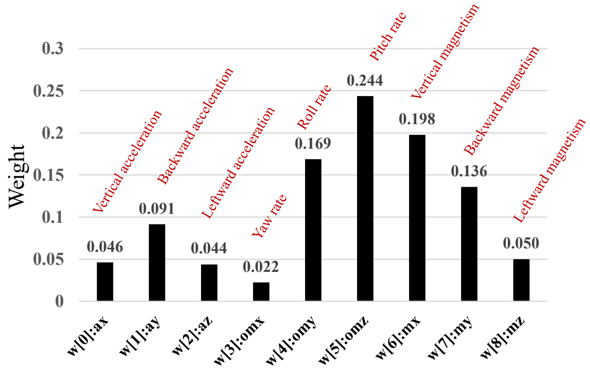


Fig. 4.8. Optimal weight for nine cross- correlation functions.

RTK-GNSS was operating at 5 Hz and the traces were over-sampled at 50 Hz by introducing a linear interpolation of 5 Hz position data. It is reasonable to assume that a vehicle's velocity cannot be changed so rapidly within 0.2 seconds. From the optimally weighted cross-correlation function, we calculated the maximum correlation value at each travel time, as shown in Fig. 4.10. We can clearly recognize the time lag profile here, even though some noise exists. In estimating the optimum time lag, we introduced a particle filter so as to reduce the influence of the noise that still exists from the optimally averaged cross-correlation function (Fig. 4.10). The estimated time lag profile is shown in Fig. 4.11 with the ground-truth value that was obtained by RTK-GNSS. The RMS error of the time lag estimation was

0.091 seconds, which is equivalent to about 1.5 m in position error (assumed vehicle speed was 16.7 m/s) and is far more precise than the results with our earlier method [1, 2]. At the travel time of 33 s, the time lag estimate reached the maximum error of about 0.3 s, which is almost equivalent to 5 m in position error (again, vehicle speed was assumed to be 16.7 m/s). Next, we precisely estimated the position error by the proposed algorithm following the data flow in Fig. 4.12. The results are shown in Fig. 4.13. Recall that the rough estimate from the time lag estimate result (Fig. 4.11) was rather overestimated assuming the vehicle speed is 60 km/h. To obtain the position of the evaluation vehicle, further steps to convert the time lag estimate into a position estimate are required (see algorithm in Fig. 4.12). For our evaluation of the position estimation error, both the reference vehicle and the evaluation vehicle are equipped with RTK-GNSSs, which have precision on the cm order. The steps for the estimation of the position of the evaluation vehicle is shown in the right-hand flow of the algorithm shown in Fig. 4.12.

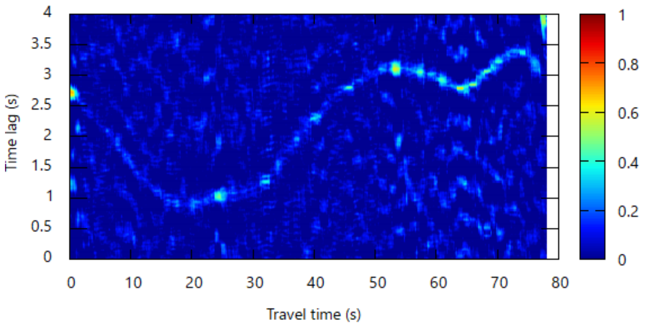


Fig. 4.9. Nine optimally weighted cross-correlation functions. The time lag profile appears clearly.

After the time lag between the reference vehicle and evaluation vehicle is estimated by the optimally weighted cross-correlation function, we calculate the corresponding date and time the reference vehicle was running for each time of evaluation vehicle's running time. Then, by retrieving the RTK-GNSS position (longitude and latitude) data of the reference vehicle, we can estimate the position of the evaluation vehicle. The details of the position error results along the evaluation travel time are provided in Fig. 4.13. We can see here that even without RTK-GNSS, the position of the evaluation vehicle was successfully estimated with an error that was far lower than the results of our previous studies.

Specifically, the RMS error was 0.38 m and the maxim error was 1.74 m.

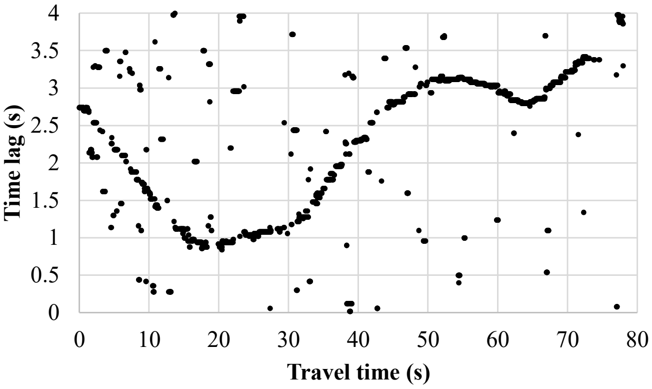


Fig. 4.10. Plots of maximum value of cross-correlation function at each travel time. Time lag profile is clearly recognizable, although some noise exists.

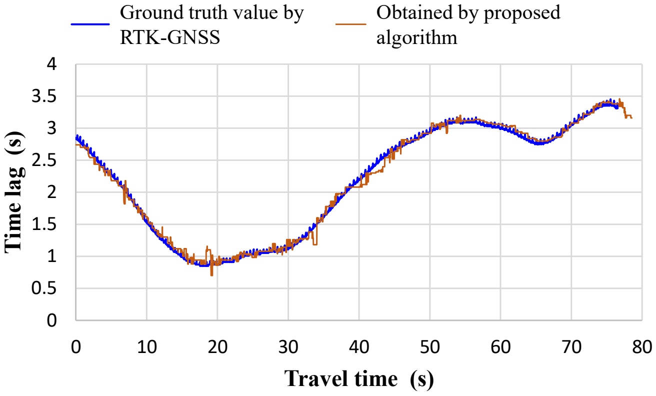


Fig. 4.11. Estimation results of time lag between the two vehicles.

The maximum error occurred at the travel time of 33 s, which is also where the time lag estimate error is apparent in Fig. 4.10. Throughout this work, all software were written in C++ and Python for both Windows PC and Raspberry Pi 4B.

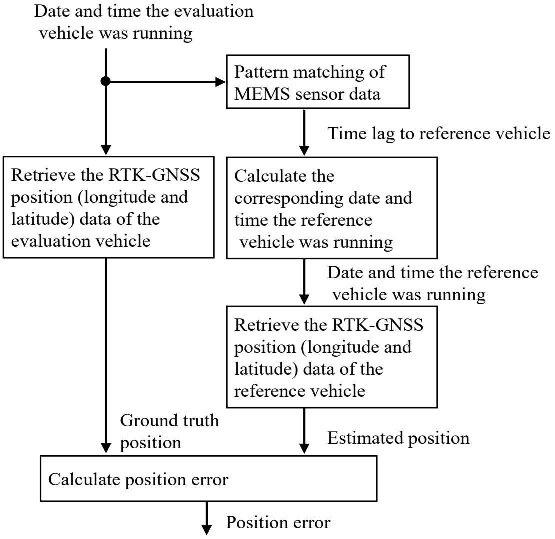

Date and time the evaluation vehicle was running
Pattern matching of MEMS sensor data
Time lag to reference vehicle
Retrieve the RTK-GNSS position (longitude and latitude) data of the evaluation vehicle
Calculate the corresponding date and time the reference vehicle was running
Date and time the reference vehicle was running
Retrieve the RTK-GNSS position (longitude and latitude) data of the reference vehicle
Ground truth position
Estimated position
Calculate position error
Position error

Fig. 4.12. Algorithm for position estimation and its accuracy evaluation in data flow representation.

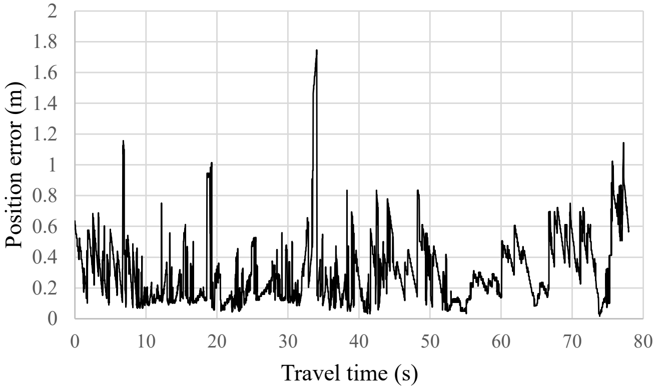


Fig. 4.13. Detailed position error along the evaluation travel time. RMS error was 0.38 m and maximum error was 1.74 m (at travel time of about 33 s).

4.4. Conclusions

We presented a vehicle localization algorithm based on MEMS sensor data and showed through evaluation tests that it achieved the highly precise position estimation. Our findings are summarized as flows.

1. We set up MEMS sensor data acquisition equipment operating at 50 Hz to calculate cross-correlation functions with a sufficient time resolution.

2. Among the nine cross-correlation functions we examined, the pitch rate, vertical magnetism, roll rate, and backward magnetism played significant roles (74.7 %) in vehicle localization in our algorithm.

3. Thanks to optimally weighting and averaging the 9 cross-correlation functions and using a particle filter, the time lag estimation worked quite well: its estimation RMS error was 0.091 seconds, which is equivalent to about 1.5 m in localization error (vehicle speed was assumed to be 16.7 m/s)

4. Our detailed position error analysis showed that the position RMS error was 0.38 m and its maximum error was 1.74 m, both of which are far more accurate than our previous results [1-6].

Overall, these findings on a real road test demonstrate that our vehicle localization algorithm based on MEMS sensor data can achieve highly precise position estimation. Future work will include combining our algorithm with a dead reckoning algorithm to stabilize the estimation and extending the algorithm to more general road networks and different vehicles and drivers.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP20K04723 and by the Japan Digital Road Map Association.

References

[1]. T. Yamagiwa, T. Yokota, Vehicle localization method based on MEMS sensor data comprising pressure, acceleration and angular velocity, in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA'22)*, 2022, pp. 786-791.

[2]. T. Yokota, Vehicle localization by altitude data matching in spatial domain and its fusion with dead reckoning, *International Journal of Mechatronics and Automation*, Vol. 8, Issue 4, Jan.2021, pp. 208-216.

[3]. T. Yokota, Vehicle localization by dynamic programming from altitude and yaw rate time series acquired by MEMS sensor, *SICE Journal of*

Control, Measurement, and System Integration, Vol. 14, Issue 1, April 2021, pp. 78-88.

[4]. T. Yokota, Vehicle localization based on MEMS sensor data, in *Proceedings of the 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE'21)*, Sept. 2021, pp. 1094-1100.

[5]. T. Yokota, Localization Algorithm Based on Altitude Time Series in GNSS-Denied Environment, in *Proceedings of the 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE'20)*, Sept. 23-26, 2020, pp. 952-957.

[6]. T. Yokota, M. Okude, T. Sakamoto, R. Kitahara, Fast and robust map-matching algorithm based on a global measure and dynamic programming for sparse probe data, *IET Intell. Transp. Syst.*, Vol. 13, Issue 11, Nov. 2019, pp. 1613-1623.

[7]. J. Tsurushiro, T. Nagaosa, Vehicle localization using its vibration caused by road surface roughness, in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, Nov. 2015, Yokohama, Japan, pp. 164-169.

[8]. A. J. Dean, R. D. Martini, S. N. Brennan, Terrain-based road vehicle localisation using particle filters, *International Journal of Vehicle Mechanics and Mobility*, Vol. 49, Issue 8, 2011, pp. 1209-1223.

[9]. E. Laftchiev, C. Lagoa, S. Brennan, Terrain-based vehicle localization from real-time data using dynamical models, in *Proceedings of the 51st IEEE Conference on Decision and Control (CDC'12)*, 2012, pp. 3366-3371.

[10]. J. Gim, C. Ahn, Ground feature-based vehicle positioning, , in *Proceedings of the 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE'20)*, 2020, pp. 983-984.

[11]. J. Gim, C. Ahn, IMU-based virtual road profile sensor for vehicle localization, *Sensors*, Vol. 18, Issue 10, 2018, pp. 33 -44.

[12]. X. Qu, B. Soheilian, N. Paparoditis, Landmark based localization in urban environment, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 140, June 2019, pp. 90-103.

[13]. ZED-F9P module, u-blox F9 high precision GNSS module, https://www.u-blox.com/en/product/zed-f9p-module

[14]. TDK, MPU-9250, Nine-Axis (Gyro + Accelerometer + Compass) MEMS MotionTracking™ Device, https://invensense.tdk.com/products/motion-tracking/9-axis/%20mpu-9250

[15]. Bosch, Humidity sensor BME280, https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/

IFSA

Advances in Robotics and Automatic Control Volume 3

Sergey Y. Yurish, Editor

Building upon the strong foundation laid by the first two volumes of *'Advances in Robotics and Automatic Control'* open access Book Series published in 2018 and 2021, the third volume continues to explore the cutting-edge developments and research in the rapidly evolving fields of robotics and automatic control.

This book covers a broad spectrum of topics, including advanced control algorithms, industrial autonomous vehicles, human-robot collaboration, sensor integration, AI applications in robotics, and emerging trends in automation. Each chapter offers deep insights into both theoretical advancements and practical applications, underscoring the symbiotic relationship between academic research and industrial implementation.

A notable highlight of this volume is the emphasis on real-world applications and case studies. These contributions demonstrate the tangible impact of research innovations on industries such as manufacturing, transportation, and beyond. The detailed case studies not only showcase the practical utility of theoretical models but also inspire further exploration and experimentation.

We hope that the book will be a vital resource for researchers, students, and professionals engaged in the field of robotics and automatic control.

ISBN 978-84-09-57872-6

9 788409 578726