TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Mohammad M. Aref

Vision-Based Macro-Micro Mobile Manipulation

Julkaisu 1568 • Publication 1568

Tampere 2018

Tampereen teknillinen yliopisto. Julkaisu 1568
Tampere University of Technology. Publication 1568

Mohammad M. Aref

Vision-Based Macro-Micro Mobile Manipulation

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Festia Building, Festia Pieni Sali 1, at Tampere University of Technology, on the 19th of October 2018, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2018

Doctoral candidate: Mohammad Mohammadi Aref
Laboratory of Automation and Hydraulic Engineering
Faculty of Engineering Sciences
Tampere University of Technology
Finland

Supervisors: Professor Jouni Mattila
Laboratory of Automation and Hydraulic Engineering
Faculty of Engineering Sciences
Tampere University of Technology
Finland

Associate Professor Reza Ghabcheloo
Laboratory of Automation and Hydraulic Engineering
Faculty of Engineering Sciences
Tampere University of Technology
Finland

Pre-examiners: Associate Professor Alexandre Bernardino
Institute for Systems and Robotics
Department of Electrical and Computer Engineering
Technical University of Lisbon
Portugal

Associate Professor Henrik Andreasson
AASS Research Centre
Department of Computer Science
Örebro University
Sweden

Opponents: Docent Tapio Heikkilä
Principal Scientist, VTT Technical Research Centre of Finland
Docent of Systems Engineering, University of Oulu
Oulu, Finland

Associate Professor Alexandre Bernardino
Institute for Systems and Robotics
Department of Electrical and Computer Engineering
Technical University of Lisbon
Portugal

Custos: Associate Professor Reza Ghabcheloo
Laboratory of Automation and Hydraulic Engineering
Faculty of Engineering Sciences
Tampere University of Technology
Finland

ISBN 978-952-15-4190-2 (printed)
ISBN 978-952-15-4243-5 (PDF)
ISSN 1459-2045

Abstract

Increasing demand for robots with broader applications has created many challenges for the commercialization of recent achievements in the fields of robotics and perception. This research focuses on formulating a solution for these challenges and to integrate vision-based information in the real-time control loop of nonholonomic Mobile Manipulators (MMs) to fulfil requirements for modularity and reusability of control subsystems. The proposed solution answers two research questions: how can visual camera feedback be improved during motion and how can dissimilar subsystems of mobile platforms and manipulators be coordinated?

Seven publications have been developed to address these issues from various perspectives. It is necessary to carefully address changes in camera outputs, discontinuities, variable latencies, and quantization errors and uncertainties. In particular, there are challenging issues about output of numerical iterative methods used for classic controllers with assumptions of well-behaved and continuous setpoints and feedbacks. Due to the multidisciplinary nature of this problem, the primary objective is to ensure that benefits are realized by lessens learned from the everyday motions of humans as a smart agent. Thus, the camera should act as the eyes, the mobile platform (i.e., the macro robot) as the body trunk, and the manipulator (i.e., the micro robot) as the arm.

The thesis analyzes two main research questions considering practical issues caused by the position feedback of several object detection methods. It is shown that a complementary M-estimator and accurate synchronization of signals can significantly smoothen a visual feedback for the control purposes. Moreover, another focus of this research is to formulate coordination of macro robot and micro robot based on the performance of the visual feedback and vehicle capabilities.

The solutions addressed in this thesis suggest the use of online evaluations of sensor feedback to determine its quality. This evaluation plays a key role in both macro−micro coordination and integration of visual feedback into robots' motion controllers.

Preface

The author is grateful for the help, guidance, and feedback received from supervisors, examiners, and colleagues at the Laboratory of Automation and Hydraulics Engineering (AUT), Tampere University of Technology during the research presented here.

Contents

List of Figures

Acronyms

4WS	Four-Wheel Steerable
AFS	Articulated Frame Steering
AGV	Automated Guided Vehicle
ALARA	As Low As Reasonably Achievable
ALARP	As Low As Reasonably Practicable
APP	Autonomous Pallet Picking
AUT	Laboratory of Automation and Hydraulic Engineering
CAN	Controller Area Network
CAD	Computer-Aided Design
CIM	Computer-Integrated Manufacturing
DOF	Degrees of Freedom
EE	End Effector
FOV	Field of View
FPS	Frames per Second
GigE	Gigabit Ethernet Communication Protocol
GIM	Generic Intelligent Machines
GNSS	Global Navigation Satellite System
GUI	Graphical User Interface
IBVS	Image-Based Visual Servoing
ICR	Instantaneous Center of Rotation

IHA Intelligent Hydraulic and Automation (Currently AUT)

iMoro IHA Mobile Robot

IMU Inertial Measurement Unit

LAN Local Area Network on the target pc

M4 Macro–Micro Controller for Mobile Manipulation

M5 Macro–Micro Multistage Controller for Mobile Manipulation

MM Mobile Manipulator

ODT Object Detection Toolbox

PBVS Position-Based Visual Servoing

POSIX Portable Operating System Interface

PURESAFE Preventing hUman intervention for incrREased SAfety in inFrastructures Emitting ionizing radiation

RFID Radio-frequency Identification

ROS Robot Operating System

RP Research Problem

SIL Safety Integrity Level

SLAM Simultaneous Localization and Mapping

TCP Transmission Control Protocol

TOF Time-of-Flight Camera

TUT Tampere University of Technology

UAV Unmanned Aerial Vehicle

UDP User Datagram Protocol

VS Visual Servoing

VSM Vehicle State Machine

List of Publications

P-I **M. Aref, Mohammad**, Ghabcheloo, R., Kolu, A., Hyvönen, M., Huhtala, K., and Mattila, J., "Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics, Automation and Mechatronics (RAM), Manila, Philippine*, pp. 218 – 224, 2013.
http://doi.org/10.1109/RAM.2013.6758587

P-II **M. Aref, Mohammad**, Ghabcheloo, R., and Mattila, J., "A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong*, pp. 6816 – 6822, 2014.
http://doi.org/10.1109/ICRA.2014.6907865

P-III **M. Aref, Mohammad**, Oftadeh, R., Ghabcheloo, R., and Mattila, J., "Fault tolerant control architecture design for mobile manipulation in scientific facilities," *International Journal of Advanced Robotic Systems*, vol. 12, p. 4, SAGE Publications, London, England, 2015.
http://doi.org/10.5772/60038

P-IV **M. Aref, Mohammad**, Ghabcheloo, R., Kolu, A., and Mattila, J., "A multistage controller with smooth switching for autonomous pallet picking," *IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden*, 2016.
http://doi.org/10.1109/ICRA.2016.7487408

P-V **M. Aref, Mohammad**, Oftadeh, R., Ghabcheloo, R., and Mattila, J., "Real-time vision-based navigation for nonholonomic mobile robots," *IEEE International Conference on Automation Science and Engineering (CASE)*, Texas, US, 2016.
http://doi.org/10.1109/COASE.2016.7743449

P-VI **M. Aref, Mohammad**, Ghabcheloo, R., Kolu, A., and Mattila, J., "Vision-guided autonomous forklift," in *Advances in Robot Design and Intelligent Control, RAAD 2016*, Advances in Intelligent Systems and Computing, vol. 540, pp. 338 – 346, Springer, Cham, 2016. *[Award Winner for the Best Robotic Application Paper]*
http://doi.org/10.1007/978-3-319-49058-8_37

P-VII **M. Aref, Mohammad**, Vihonen, J., Ghabcheloo, R., and Mattila, J., "On Latencies and Noise Effects in Vision-Based Control of Mobile Robots,"in *Advances in Service and Industrial Robotics. RAAD 2017, Mechanism and Machine Science,* vol. 49, pp 191 – 199, Springer, Cham, 2017.
http://doi.org/10.1007/978-3-319-61276-8_22

1 Introduction

Mobile Manipulators (MMs) have multidisciplinary applications and are applied to many aspects of robotics that share the challenges of high-dimensionality, uncertainty, and task variability [10]. Since the 1980s, robotic arms have provided comparable performance to human arms and hands and have been substituted for human labor to perform repetitive tasks, especially in automotive industries. As manipulator technology has evolved, robots have begun to appear outside the limited environments of Computer-Integrated Manufacturing (CIM) cells and production lines. By 2022, the service robotics market is expected reach a volume of $23 billion per year at a compound annual growth rate of 15% [1].

Common mechanisms for service robots are MMs. For example, mobile platforms carry one or two manipulators to widen workspace [36], in which service robots target unstructured environments while sharing the environment with humans [60], unlike CIM cells. Broad application areas and the variety of tasks that these robots perform highlight demands for vision-based methods to provide both indoor service robots and outdoor field robots with suitable environmental feedback. Thus, this thesis developed within two projects and test cases aligned to these demands.

1.1 PURESAFE

Project PURESAFE[2] (i.e. Preventing hUman intervention for incrREased SAfety in inFrastructures Emitting ionizing radiation), was a project partially targeting engineering of complex robotic systems for scientific facilities, such as CERN[3] and GSI Helmhotz FAIR [4], as shown in Fig. 1.1. It was funded under the European Commission's Seventh Framework Programme and the Marie Curie Actions. Tampere University of Technology (TUT) acted as the leading organization for this project. Design and manufacturing of the

[1] Service Robotics Market by Operating Environment, Application, and Geography-Global Forecast to 2022, reported by Markets and Markets, SE 2714, January 2017.

[2] http://puresafe.web.cern.ch

[3] https://home.cern

[4] https://www.gsi.de/en/researchaccelerators/fair.htm

iMoro was also during this project, as the first mobile manipulator addressed in physics hall of fame, illustrated in Figure X.30 of [37].

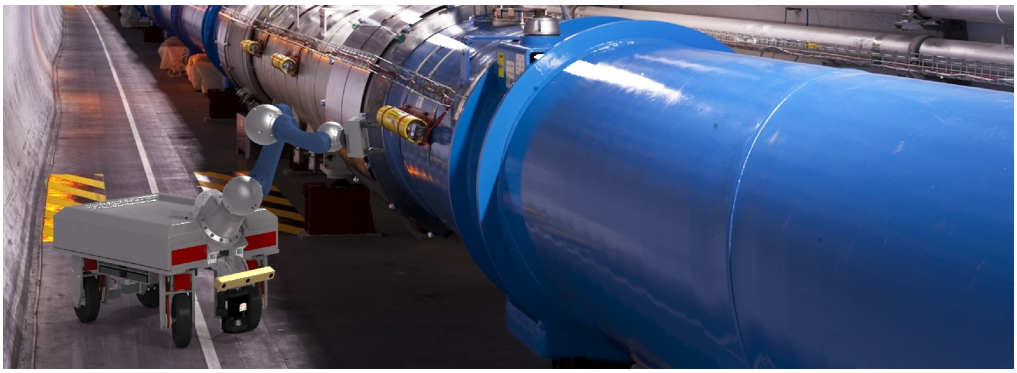


Figure 1.1: Graphical proof of concept proposed by [51] for iMoro inspection applications in CERN LHC tunnels as a modular wheeled mobile manipulator for the PURESAFE project

1.2 GIM

Some algorithms of this research were implemented in autonomous field robots that were developed at the Finnish Center of Excellence in Generic Intelligent Machines (GIM)[5] and partially funded by the Academy of Finland from 2008 to 2013. Research teams from Aalto University and TUT improved technologies for field robotics and MMs. Several autonomous and multipurpose mobile machines and a test track were created for GIM projects, such as one of the test cases in this research, namely the Avant forklift shown in Fig. 1.2.

1.3 Research Problems (RPs)

The multidisciplinary problem of MMs includes many engineering branches and demands a variety of complex technologies to be implemented. As much as possible, pre-existing tools were used for basic robot functionalities, and the primary objective of the research was to find solutions for two research questions (provided below) by considering multidisciplinary approaches.

RP.I **Coordinated Motion For Mobile Manipulation**
How do two subsystems in a mobile manipulation system cooperate to accomplish a manipulation task?

[5] http://gim.aalto.fi/

Figure 1.2: Avant autonomous forklift; AUT's mobile laboratory

RP.II **Analysis and Integration of Visual Feedback in Control Loop of a Mobile Platform**

What are the effects of vision-based feedback in control loops and how can undesirable effects be reduced?

For the MM test cases using autonomous vehicles, maximum utilization of existing hardware and software modules was required to ensure the practical use of output for commercialization.

1.3.1 RP.I: Coordinated Motion for Mobile Manipulation

Introducing manipulators for applications beyond the capabilities of stationary installations within a CIM cell shows that there is demand for mobile platforms. Table 1.1 compares the use of mobile platforms for manipulation tasks, revealing challenges that arise from dissimilarities in many aspects. Integration of these two systems can compromise precision in wider workspaces; thus, this research focused on generating mobile platform motion and on-board manipulators to perform shared tasks. The research also addressed the aspects of a task that each subsystem should share in terms of end effector motion.

Table 1.1: Summarized common properties of macro and micro subsystems.

Subsystem	Motion Mechanics	Workspace	Motion Precision	Response
Macro (mobile platform)	Nonholonomic	Unlimited (but constrained)	Low	Slow
Micro (Manipulator)	Holonomic	Limited Spatial	High	Fast

1.3.2 RP.II: Analysis and Integration of Visual Feedback in the Control Loop of a Mobile Platform

Service robots usually receive visual information extracted from two- (2D) or three-dimensional (3D) camera images. This information can be about robot status and functionalities, environmental perception, or both. Recent developments in machine vision, embedded processors, and image perception in deep convolutional networks now make such data extraction possible. In MM tasks, a common application of visual information is to track an object for manipulation purposes; thus, vision output influences feedback during closed-loop manipulation or of a path-following controller. There are several differences in the fields of computer vision and robotic control that give rise to challenging issues for system integration. One fundamental difference is how success is defined in these fields. For a computer vision algorithm, a 90% success rate in detection is considered extraordinarily good and beyond human capabilities. Relevant examples of this are the winners of the ImageNet [25] competitions in recent years with success rates higher than 93%, for instance by GoogLeNet [70]. Although this level of quality is useful for user-assistive tasks, it causes problems in autonomous systems because a 90% success rate means that the system can fail to recognize correct targets once in ever 10 attempts. In a robotic control system, the autonomous system should be positioned at strictly descending distances from the target object, and system quality is measured based on errors when tracking an object or following a path in terms of repeatability and accuracy. Therefore, to accomplish a vision-based MM task, both RPs should be considered.

In the steerable mechanisms of mobile platforms, nonzero degrees of steerability, as in the Avant forklift or iMoro, mean that oscillations are more obvious if the steering angles directly depend on the instantaneous path curvature. For field robots, this issue is less prevalent due to high inertia and viscous friction in power transmission. However, for accurate and fast update rates in the iMoro, each oscillation in the curvature measurement causes displacement of steering even when the body coordinates have not changed.

Studies focusing on these RPs are included in this thesis. For example, RP.I was considered by P-II, P-IV, and P-VI, while publications by P-I, P-V, and P-VII focused on RP.II. Moreover, P-III and [54] described the architecture of the iMoro.

1.4 Requirements and Scope of the Research

Besides the scientific challenges, dealing with service robotic or field robotic test cases requires certain properties and specifics to develop solutions for RP.I and RP.II. The main concern is to preserve system capabilities for further developments or commercialization

of existing or developing software modules. In this research, availability and reusability of methods for realistic experimental setups were the primary concerns. Test cases for outdoor field robotics or indoor service robotics are commonly built on pre-existing and advanced products and methods; for instance, in the Avant, control of hydraulic actuator valves for setting the vehicle's articulation angle was studied by [29]. For the iMoro, the same level of development was achieved in cooperation with research conducted on the path-following control of four-wheel steerable mobile platforms as described in [57]. Based on the generic architecture for MMs in such systems and applications, selected requirements are listed as follows. They are compatible with the complete requirements of the PURESAFE project, which were developed in conjunction with the needs and requirements addressed in meetings with the Engineering Department of CERN [41].

1. The output performance of system positioning should maintain repeatability for the entire system such that the iMoro could continuously hold a small object (e.g., $3cm \times 3cm$ cube) within its gripper jaw fingers. The same is required for the Avant forklift, which must be able to pick up a standard EUR pallet. Thus, system repeatability must be better than $\pm 7cm$ because this accuracy allows the test to be conducted using commercially available forklift forks and pallets.

2. All targets should be marked and defined using a monochrome single ARTag [26] or have a bundle with detection preferences set to the farthest possible distance.

3. The visual feedback should be available at variable process times of 5 to 10 Frames per Second (FPS). Controller feedback must be available for minor sample times of hard real-time process units. For the Avant test case, this is about 20 milliseconds, and for the iMoro test case, this is about 5 milliseconds. The controller is responsible for driving the lower level actuators at each sample time, and without this feedback, the system is enable to perform emergency soft stops.

4. The wheel odometer of the iMoro should be available at each sample time based on the wheel angular encoders. However, for the Avant, the wheel odometer should also be an event-based sensor that transmits updates for each appearance in the proximity sensor label.

5. Mobile platforms should be based on nonholonomic steerable mechanisms.

6. The steering and driving actuators can have limited velocity and acceleration capabilities.

1.5 Thesis Contribution

The primary contribution of this thesis is an experimentally verified method for integration of eye-in-hand systems for object tracking that considers limitations of system integration and constraints of the nonholonomic mobile platform. The publications analyzed to solve the RPs address several viewpoints and two different mechanisms. For outdoor conditions, such as the test environments in P-II, P-IV, and P-VI, the signals of the Global Navigation Satellite System (GNSS) were commonly available and the vehicles used global coordinates during task performance. However, these coordinates were not sufficiently accurate for pallet picking, and local feedback was necessary to ensure precise motion control for this purpose. The proposed method caused the forklift to deviate from the global path, based on logics provided in P-VI, when it sees the desired target, as in P-II. Commonly available industrial cameras do not provide sufficiently accurate visual feedback at distances greater than three meters. Consideration of special constraints, is necessary to add more information to the system. As shown in P-II, assumptions of constant pose for target objects with respect to an inertial frame can significantly improve system robustness and smoothness.

The same solution was modified for better accuracy and faster responses for the iMoro, and it was shown that adapting a path-following controller for vision-based motion control can significantly improve the repeatability of positioning for mobile platforms of service robots, as in P-V. This solution, which is briefly described in Section 3.2, supported the real-time capabilities of a vision-based MM. Additionally, this solution considered uncertain visual feedback that was dependent on object distance. As proposed in P-VII, using complementary M-estimator to precise consideration of robot footprints, after compensation for variable Object Detection Toolbox (ODT) delays, can significantly improve vision feedback. This improves the smoothness of signals in closed-loop and tolerance to bouncing of visual feedbacks. Thus, pure motion of the robot's mobile base maintained the robot's positioning better than $\pm 3cm$ at End Effector (EE).

This research was conducted in conjunction with research on path following and object detection [52–58, 3, 4].

1.6 The Author's Contribution to the Publications

This section briefly explains the role of the author in each of the listed publications.

Publication P-I: The author developed the presented methods. Professor Reza Ghabcheloo initiated the idea and edited the paper. M.Sc. Antti Kolu helped with the experimental setup, and Dr. Mika Hyvönen

helped with the hardware-in-the-loop simulation setup. Professors Kalevi Huhtala and Jouni Mattila reviewed the paper.

Publication P-II: The author developed the initial idea and the theoretical framework and wrote the paper. As academic supervisors, Professors Reza Ghabcheloo and Jouni Mattila reviewed the paper and made corrections and suggestions.

Publication P-III: The author developed the initial idea and the theoretical framework and wrote the paper. Dr. Reza Oftadeh helped with the experimental setup. As academic supervisors, Professors Reza Ghabcheloo and Jouni Mattila reviewed the paper and made corrections and suggestions.

Publication P-IV: The author wrote the paper and developed the methods presented. M.Sc. Antti Kolu helped with the experimental setup and the hardware-in-the-loop simulation setup. As academic supervisors, Professors Reza Ghabcheloo and Jouni Mattila reviewed the paper and made corrections and suggestions.

Publication P-V: The author wrote the paper and developed the methods presented. Dr. Reza Oftadeh helped with the experimental setup. As academic supervisors, Professors Reza Ghabcheloo and Jouni Mattila reviewed the paper and made corrections and suggestions.

Publication P-VI: The author wrote the paper and developed the methods presented. M.Sc. Antti Kolu helped with the simulation and experimental setup. As academic supervisors, Professors Reza Ghabcheloo and Jouni Mattila reviewed the paper and made corrections and suggestions.

Publication P-VII: The author wrote the paper and developed the methods presented. Dr. Juho Vihonen helped with the experimental setup and review of signal processing methods. As academic supervisors, Professors Reza Ghabcheloo and Jouni Mattila reviewed the paper and made corrections and suggestions.

1.7 Outline of the Thesis

This compendium thesis is comprised of five chapters. The arrangement of chapters and publications is illustrated in Fig. 1.3. After the first chapter, which introduces the RPs, scope of the research, and contributions, subsequent chapters are organized as follows.

Chapter 2 starts with a brief review of fundamental studies and materials for definitions and terminologies in the field. Then, the literature review is presented in four categories, highlighting various research viewpoints. Then, experimental studies of state-of-the-art methods are compared in terms of long-range detection and positioning accuracy of robots.

Chapter 3 addresses the conceptual design basis of the publications for answering the RPs. The second half of this chapter describes the relevant experimental setups and highlights their similarities and differences. This chapter ends with a selection of practical issues among experimental studies about these robotic problems.

Chapter 4 presents the research conclusions based on observations and experiment results that answer the RPs (i.e., visual input and interference of multidisciplinary concepts). Discussions are categorized into four subjects, and the final part of the chapter concludes the discussions and addresses possibilities for further research.

Chapter 5 consists of summaries for each of the seven publications. This chapter explains the connection between the thesis RPs and the publications. This chapter is followed by the publications, which are relevant published papers.

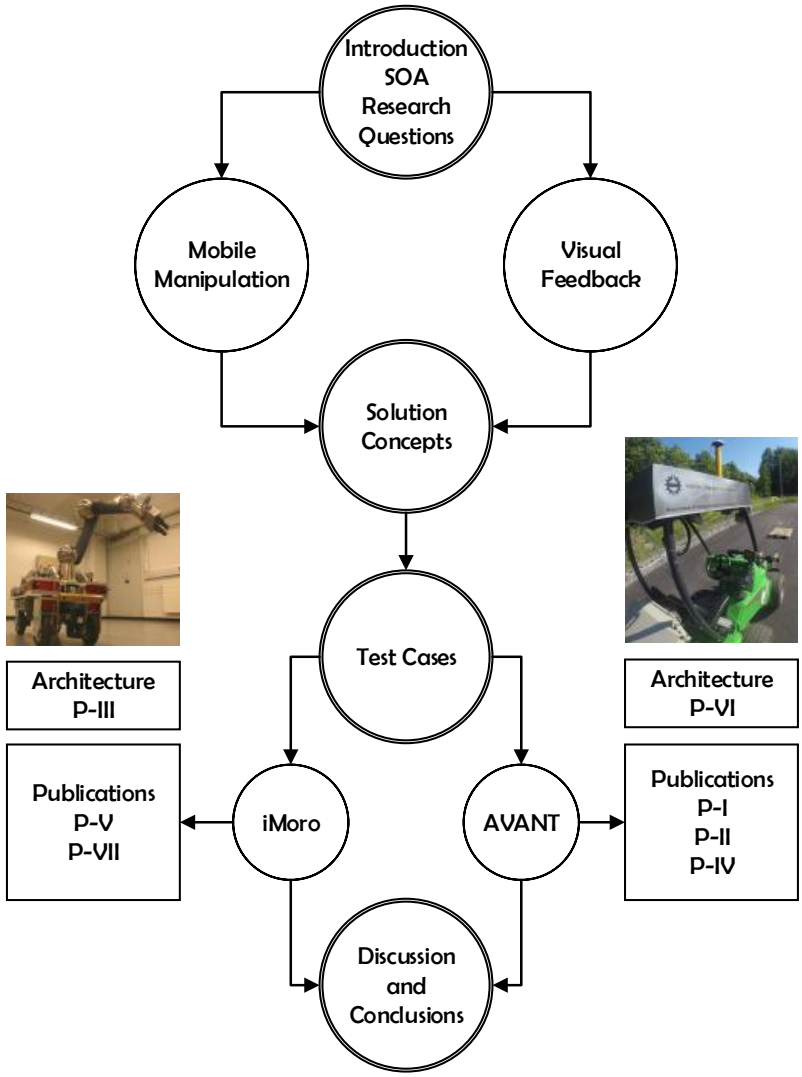


Figure 1.3: Thesis structure.

2 State of the Art

This chapter briefly describes the fundamental studies of control of MM using visual feedback as well as a selection of achievements in the field. Similar approaches are utilized for different applications, and the majority of studies can be categorized under these topics: position-based or image-based visual servoing, visual guidance for mobile platforms, docking and motion control of AGVs, and pallet picking.

2.1 Visual Servoing

Visual Servoing (VS) methods are categorized into three main types: Image-Based Visual Servoing (IBVS) [15], Position-Based Visual Servoing (PBVS) [79], and a hybrid of both [49]. IBVS is a visual target-tracking method that is most widely used in the literature to compare a desired set of features to the features of current images to produce necessary commands for robot motion generation [2]. Iterative feature matching and transformation extraction methods incur high costs but are presented in [39], and due to limited features, these methods are compatible with real-time systems but are best suited to 2D problems, with initial knowledge of depth or scale [34], without large rotational movements [14]. IBVS is mostly used for short distances and eye-in-hand applications [17], while PBVS is based on the estimation of the target pose using image features. Because pose estimations (e.g., in Cartesian space) are provided in PBVS, vision output can be readily integrated with other sensor modalities. However, the calibration of the extrinsic parameters of the camera with respect to the robot body plays a significant role in the success of manipulation [81]. A hybrid method, such as $2^1/_2$ D VS [49], combines both IBVS and PBVS to extract the translational movement of a camera (using the PBVS method) and its rotation in a 2D image coordinate system [13, 14]. Several studies have been reported in this regard [13, 14] [33].

This thesis work is comparable to works on motion control of mobile systems based on exteroceptive sensor data. Although the problems are similar, these works use different terminologies and applications; however, the common goal is to position a mobile (or a MM) robot in accordance with an object within its environment. This becomes more

challenging when the robot must remain in contact with the environment as well. Related works in the literature can be loosely categorized as those pertaining to pallet picking or docking, IBVS, and PBVS. They use sensors or driving mechanisms not necessarily similar to the current research. The intersection point of all these studies is the control architecture of mobile platforms based on exteroceptive sensor data. For AGV docking problems, the environment is more structured, and robustness is the main concern. In contrast, the aim of IBVS research is to ensure more flexibility, while PBVS research combines these aims; this is the category in which the current thesis resides. The following sections describe each research category in more detail.

2.2 Motion Control of Mobile Manipulators

A variety of approaches have been implemented to control or distribute tasks among the joints of a MMs. In some cases, the approach depends on the specific capabilities of the mechanical design, such as the capabilities of the fourth generation of the Care-O-Bot [28]. Early works considered the entire system of the MM as a redundant robot and, therefore, proposed extensions of redundancy resolution methods for MMs, without considering the effects of visual feedback in control loops. Comprehensive studies by Khatib [35] and Seraji [67] are good examples of these works, as well as methods later addressed in the Springer Handbook of Robotics [10]. These solutions are more useful when system response is not limited by nonlinear changes in the object – MM pose estimation. In the close neighborhood of the object, these approaches are applicable because of better visual feedback and limited demand for heading change from the mobile platform [17]. However, far distances, especially when feedback changes can demand large rotational displacements, are not the best cases for these methods. The majority of complexities in MMs are due to the steering of mobile platforms and include nonholonomic constraints and singularities, which have been more extensively studied in relation to path following for combinations of active and passive wheels and steering mechanisms [56, 60].

Despite limitations in efficiency and approach angle of its mobile platform, similar to KUKA's omniMove and youBot [8], the nonholonomic dynamics of MMs are avoided with the use of *omni-directional wheels* (e.g. Swedish or Mecanum wheels). Combinations of traction forces generated by these wheels partially cancel the force or torque of other wheels such that the resultant planar force or torque is similar to a planar holonomic joint. Since cancellations are based on wheel – ground contact friction, mobile platform become shaky unless the application is limited to flat and even surfaces and flexible wheels. Studies similar to [32] treated *Castor wheels* with excentricity as holonomic mechanisms because of their capability to generate force by steering. However, these approaches are solely dependent on very specific structures. The vast variety of recently developed

commercially available MM platforms, such as PR2, Care-O-bot, Tiago, Baxter, and Fetch [61], as well as autonomous field robots, are not suitable for use with these methods.

The proposed method in P-VII is comparable to recent work in [16], in which a simulated mobile robot tracked a target using vision feedback. This method considered piecewise path generation for following a given target. However, it overlooked some issues and did not address low-frequency nonlinear changes in optical systems and deviations from pin-hole camera models. Moreover, the switching method was used only within a predefined coordinate frame. In P-II and P-IV, it was shown that for long distances and for integration with systems in real-world applications, it was necessary to have switching capabilities and the ability to change from global to local coordinates to enable independent control loops and to coordinate complicated tasks and synchronization for disturbance rejection, which is similar to solutions addressed in [73].

Compared to other contemporary methods provided in Table 2.1, the proposed method achieves outstanding performance in positioning of the mobile platform by avoiding compromised performance for smoothness, compared to other relevant works that have lowered control-loop frequency when visual feedback is lacking. For instance, [18] updated inputs directly after each ODT data update, meaning that the entire system had variable sample times due to visual perception, which is not acceptable for large-scale automation systems dealing with hard real-time requirements. The table also showed that, in the proposed methods (P-IV, P-VII) for better system integration, positioning of the mobile platform does not lead to loss of accuracy.

2.3 Docking and Motion Control of Vehicles and AGVs Based on Exteroceptive Sensor Data

For AGVs with flexible tasks, the docking process must be able to handle different stations and loads. A comprehensive study reviewed research methods and patents for

Table 2.1: Research on Accurate Mobile Robot Navigation

Reference	Local Update	Global Update	Max Positioning Accuracy	Sensor	Robot
[69]	35 Hz	1.5 Hz	4 cm	Range	omniRob
[62]			6 cm	2xLaser	omniRob
[18]			8 cm	Vision	Pioneer3AT
P-IV	50 Hz	5-7 Hz	10 cm	Vision &IMU	Avant Forklift
P-VII	200 Hz	7-10 Hz	3 cm	Vision	iMoro

* The addressed papers cover a wide range of functionalities. Only the parts comparable to the proposed methods are considered in the table. The applications and environments are not exactly similar. Empty cells indicate lack of information.

this process [68]. Applications of AGVs are more dedicatedin structured environments; for instance, some AGVs navigate by following wires or painted lines or by monitoring through Radio-frequency Identification (RFID). These options are more suitable for workshops or production lines with repetitive tasks rather than high-maneuver motion-task demands. Thus, such research is better for precision and repeatability that compromises flexibility and integration.

For outdoor vehicles, common sensor input are GNSS data, which are globally available; however, even good control methods can generate $0.5m$ lateral errors for high maneuvers [50]. Lateral errors are a key parameter for placement of a forklift's fork into the proper position during picking, and errors must be smaller than $10cm$. Usually field robots, similar to Avant, do not carry a dexterous manipulator, and only the mobile platform is capable of affecting this error. Since this thesis discusses a nonholonomic mechanism for mobile platforms, this error must be minimized during a platforms' longitudinal motion; therefore, vehicles cannot be stopped to improve estimation uncertainties using high-order filtering methods.

A comprehensive review of applicable motion control methods are presented in [74]. This paper states that the control methods based on path following, so called vector pursuit controllers such as [80], have good functionalities for controlling heading and curvature. However, it considers the necessity of the look-ahead distance for these methods as a negative point. As shown in P-IV, by choosing an appropriate strategy for multistage inclusion of visual information on path segments, the look-ahead distance cannot be a limiting factor for pallet-picking tasks. Additionally, methods based on multiple trajectories for control of autonomous vehicles was studied by [27].

Generally speaking, the papers targeting AGV docking are more suited for industrial workshop environments, in which workspace can be adapted to vehicle capabilities. This is not possible in outdoor environments, such as construction sites. Thus, the approach chosen in this thesis was to keep all functionalities of autonomous vehicles, as long as the desired target was identified. After consistent recognition of the target, the robot began to act based on the target's pose and its imaginary desired path of manipulation toward it. There are also branches of research on task planning and scheduling of AGVs [19], although such research is out of the scopt of this thesis. However, they highlight a level of autonomy that is expected from the MMs in similar environments. This topic includes less legal and safety issues in industrial environments, and therefore, the architectures can be simpler compared to service robots.

2.4 Vision-based Control of MMs as a Visual Servoing Problem

Research addressing IBVS focus on simple control architecture that allows control of an entire control loop using a single VS controller. This is not possible in complex architecture, such as those presented in P-VI or P-III. [21] looked at the integrity of vision in the control loop together with adaptation of gains when considering robot dynamics. However, this work is not concerned about timing and data fusion in different coordinates because the camera was stationary, and therefore, the coordinate frames did not change. The delay effects on position errors, addressed in P-V and P-VII, were not considered because such errors depend on speed values. Since this method learns the dynamics of entire systems, it can include synchronization errors, and when moving coordinate frames, this method cannot be applied.

VS for manipulation or navigation was also addressed by [77] and [78], who kept the actuator saturated and used the maximum available actuation but only at short distances to observe error reduction as a controlled and damped step response. Another study highlighted the importance of VS of nonholonomic mobile platforms [83] and provided a solution for this problem based on minimizing errors among matched features in the camera-image frame in a differentially driven mobile platform application. Here, changes in camera image depending on range, can affect the behavior of the vision-based estimated values, which can cause a robot's path to deviate or increases errors in estimations of the relative pose. In [84], this change was assumed to be a disturbance. However, this could have been caused by changes in the focus of the optics. This issue is considerable if the robot must be used for long-range applications [40]. A similar problem, which has relevance for long-distance detection, is the Unmanned Aerial Vehicle (UAV) landing problem, especially for moving targets [1]. Sometimes, detection is based on utilization of deep learning in comparison to Google Street View [5]; these works have inspired broader applications for this thesis' work as addressed in Section 4.5.

2.5 Vision-based Control for Autonomous Pallet Picking

For vision-based controllers in industrial applications, target detection inconsistencies together with the limitations of the mechanical system are both common sources of error [82]. A proper controller should be able to operate with both error sources while improving in overall performance. In the literature, some studies emphasized one error source more than the other. Since Autonomous Pallet Picking (APP) involves more large-scale industrial MMs test cases, industrial solutions and system integration are considered more often than VS. A group of studies has considered pallet detection and integration of the visual sensor in the control topology, such as [31], whereas work such as that of [7]

aimed to correct data while camera is approaching the pallet. A majority of these works employ at least two cameras or monocular vision systems with defined Computer-Aided Design (CAD) models [11] or incorporate both laser scanners and cameras [6].

Typically, the maximal range of the vision detection methods is three to four meters in front of the vehicle for known targets. For instance, the reported maximal detection ranges are:

- $3m$ for the CAD-based method of [11],
- $4m$ for a double-sensor architecture in [6],
- $3.7m$ for the MCSMwith stereo vision [64, 12],
- $3.5m$ for the Macro – Micro Controller for Mobile Manipulation (M4) method P-II, and
- $2m$ for variable sized pallets using a laser scanner [38].

Only research benefiting multiple-view laser scanners provide longer distances for forklift configuration space up to $6m$ [76, 75]. For large articulated frame-steering machines with reduced mobility, the approach angle must be corrected from longer distances to avoid extra back-and-forth maneuvers. The proposed method for APP is capable of detecting a pallet (with markers) from $6m$ away by employing a monocular vision system. The Macro – Micro Multistage Controller for Mobile Manipulation (M5) method in P-IV improved the M4 method of P-II by incorporating step-by-step switching among the control modes and path segments to include marker-detection data to the extent of its validity. Its immediate result was a wider configuration space with a higher impact on performance and pallet picking feasibility and more application areas. The method does not impose limitations on detection methods; therefore, it is capable of integration with any vision, Time-of-Flight Camera (TOF), or range sensors that are suitable for object detection based on exteroceptive sensor data.

There are multiple controllers proposed for object manipulation and pallet picking in the literature. Some navigate the robot in the image space or camera coordinate, such as the mobile camera-space manipulation method (MCSM) [66, 12, 63, 65]. The intrinsic complexity, nonlinear dynamics, and nonholonomic constraints of such machines make the motion control problem very challenging to overcome [20],[56] because it requires both path planning and following algorithms for APP [71],[75, 72]. Docking problems face similar challenges [74], and one of the fundamental issues for these categories of APP studies is the definition of the operational coordinate that becomes a reference for the desired path, errors, and the target pose [76]. The operational coordinates for APP must

be global or local, and selecting either has its own advantages or disadvantages. For global coordinates, integration with normal autonomous navigation modules of a vehicle is easier compared to that of local coordinates. However, any update in the desired path requires re-execution of the global path planner since the improvements in the target pose estimation occur when the robot approaches the target. In contrast, operations using local coordinates provide high frequency location and detection feedback but change the navigation control architecture, which may lead to undesirable drifts in the position estimation.

The proposed methods, M4 and M5, take advantage of both operational coordinates. During the time that the MM follows a desired path and detects the object for the first time, the proposed method plans a smooth path before switching to a local operation coordinate. Therefore, the robot motion avoids jumps and drifts during navigation in both global and local coordinates as well as during switching intervals. Another benefit of this method is the ability to start visual servoing from farther distances, which provides more space and time for the MM and, therefore, smoother steering of the nonholonomic mobile base. This feature, in most of the scenarios, eliminates the need to align the forks by driving backward and re-planning the path.

3 Solutions and Results

3.1 Inspiration

The majority of challenges faced in service robotics arise from tasking robots together with (or on-behalf of) humans in their environments. Considering real-world solutions to such challenges can provide smart solutions rather than numerically optimal solutions. This is also true for isomorphism between human arm and robotic manipulators. For example, manipulators in the automotive industry work in the same way that the human arm works because the entire design of production lines was initially based on human abilities. Improvements in mechatronic design and cyber–physical systems were introduced as a revolutionary movement, as in Industrie 4.0 [30]. However, the entire process is evolutionary and based on improvements in integration of multidisciplinary methods and products during the third and fourth industrial revolutions. A noticeable example is the difference between the utilization of AGVs and autonomous vehicles. As a result of being in a simpler environment, AGVs are now functional in industrial and structured environments; however, autonomous passenger cars are still under development due to the complexities of interactions with nonindustrial and unstructured environments. Since these devices are designed for humans, robot's actions are considered to be intelligent or smart if they are acceptable and similar to human actions or justifiable by skillful operators. The words smart or intelligent might be vague from this viewpoint and not necessarily refer to optimality in a specific domain, but they do mean better integration of process power and control in a mechatronic system. The majority of the literature addressed in this thesis embrace a multidisciplinary perspective as a way of finding inspiration from manipulation of objects in everyday life, as shown in Fig. 3.1.

If a person is told to pick up a box from a shelf, the following steps are carried out subconsciously.

1. Understand the task.

2. Consider the way toward the assumed location of the shelf from the current location.

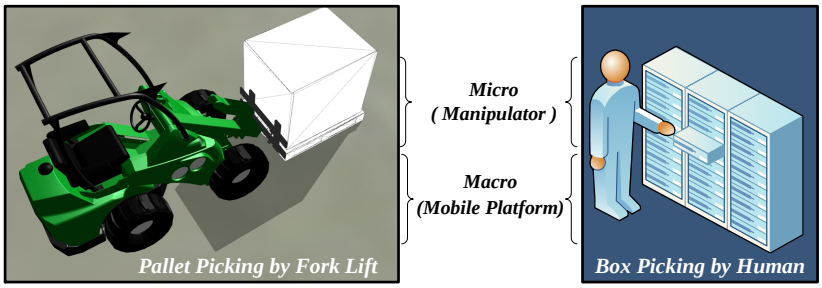


Figure 3.1: An inspirational concept for MMs in everyday life

3. Move toward the shelf location.
4. Update shelf pose assumption, as soon as it is visible.
5. Seek the box as soon as the shelf is in sight.
6. Check that the box label is the same as given.
7. Pose the body such that the arm has enough freedom (dexterity) to move.
8. Grasp the box using arm and hand motions.
9. Pick up the box.

Our muscles, brain, and nerves work smoothly through these steps such that the outcome is a smooth motion, adjusted for the task and the capabilities of the person; it takes into consideration environmental effects and is reactive to changes. These conditions are further highlighted by considering different situations. For example, someone with a shorter arm stops closer to the shelf, while someone who moves quickly may not stop or pause during steps. On a slippery floor, or as a result of illness, fine adjustments of the trunk might be ignored and the arm moved more. Studies on human motor control and perception can partially confirm these observations [23]. The same issues must be corrected for vehicles and mobile platforms that use MMs. Based on these analogies, the MM uses a macro–micro architecture, while the mobile platform acts as the body, and the manipulator acts as a human arm. In this way, robot cognition acts similarly to accomplish the aforementioned steps for object manipulation.

3.2 Solution Concept for Integration

If inspirational concepts are compiled into motions and measurements of coordinate frames, their relationships are described by homogeneous transformation matrices as in SE(3)[1], which is denoted by $\boldsymbol{T}$. For example, the following tensor comprised of an orthonormal rotational matrix and position vector describes a body frame $\{\mathcal{B}\}$ in an inertial frame $\{\mathcal{I}\}$ represented in $\{\mathcal{I}\}$.

$$ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(x,y,z,\varphi,\theta,\psi) = \left[\begin{array}{ccc|c} & {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}(\varphi,\theta,\psi) & & {}^{\mathcal{I}}\boldsymbol{P}_{\mathcal{B}}(x,y,z) \\ \hline 0 & 0 & 0 & 1 \end{array}\right]_{4\times 4} \tag{3.1} $$

If the coordinate frames are assumed, as illustrated in Fig. 3.2, the coordinate frame $\{\mathcal{W}\}$ is stationary in $\{\mathcal{I}\}$. Therefore, the transformations between these coordinate frames are ${}^{\{\mathcal{W}\}}\boldsymbol{T}_{\{\mathcal{I}\}}$ or ${}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{W}\}}$, and therefore ${}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{O}\}}$ can be assumed as a constant matrix, which is equal to

$$ {}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{O}\}} = {}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{B}\}}\ {}^{\{\mathcal{B}\}}\boldsymbol{T}_{\{\mathcal{C}\}}\ {}^{\{\mathcal{C}\}}\boldsymbol{T}_{\{\mathcal{W}\}}\ {}^{\{\mathcal{W}\}}\boldsymbol{T}_{\{\mathcal{O}\}}, \tag{3.2} $$

for which parameters are described in Table 3.1. Based on this assumption, it is possible to have redundant measurements for relative coordinates.

The same type of loop closure can be written to include a manipulator.

$$ {}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{O}\}} = {}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{B}\}}\ {}^{\{\mathcal{B}\}}\boldsymbol{T}_{\{\mathcal{M}\}}\ {}^{\{\mathcal{M}\}}\boldsymbol{T}_{\{\mathcal{M}^*\}}\ {}^{\{\mathcal{M}^*\}}\boldsymbol{T}_{\{\mathcal{O}\}} \tag{3.3} $$

Based on these loop-closures, the body pose $\{\mathcal{B}\}$ (as in P-V and P-VII) or manipulator pose $\{\mathcal{M}\}$ (as in P-I) can be extracted or the desired values (as in P-II) can be extracted

[1] special Euclidean group in 3D space

Table 3.1: Transformations between different coordinate frames illustrated in Fig. 3.2

Matrix	Description	Determination
${}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{O}\}}$	Object's pose in the inertial frame	matrix composition
${}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{B}\}}$	Mobile platform's pose in inertial frame	odometer or GNSS
${}^{\{\mathcal{B}\}}\boldsymbol{T}_{\{\mathcal{C}\}}$	Camera coordinates in the body frame	extrinsic calibration
${}^{\{\mathcal{C}\}}\boldsymbol{T}_{\{\mathcal{W}\}}$	Work piece (or marker) in the camera image frame	vision system
${}^{\{\mathcal{W}\}}\boldsymbol{T}_{\{\mathcal{O}\}}$	Grasp pose with respect to the work piece	by definition (grasp planning)
${}^{\{\mathcal{B}\}}\boldsymbol{T}_{\{\mathcal{M}\}}$	Manipulator pose with respect to the body	manipulator joint values

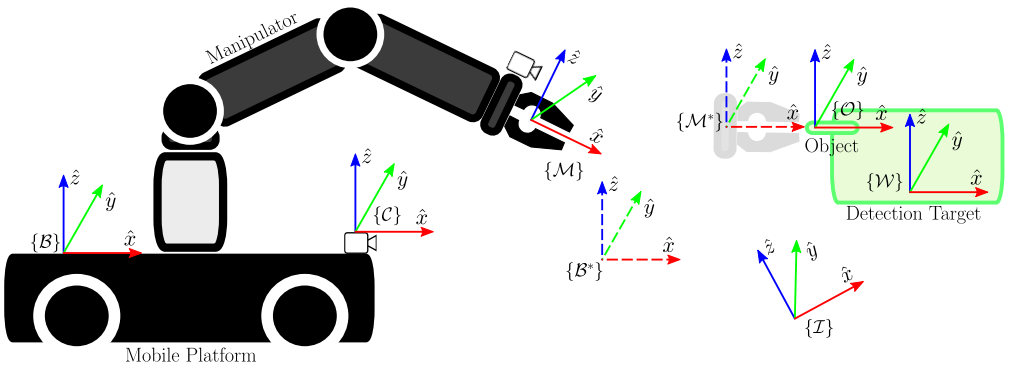


Figure 3.2: Defined coordinate frames of MMs, camera, and object.

in accordance with the object based on two sources. If practical issues are ignored and assuming that ${}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{O}\}}$ is constant in time, multiple measurements can be obtained during a time interval independent of the robot's motions. This is helpful in different situations because changes in estimated values for ${}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{O}\}}$ can be a measure of performance for sensor output, as shown in P-VII. The entire chain of transformations can be mapped in the visible coordinate frame $\{\mathcal{O}\}$, such that the changes remain in the same coordinate, theoretically. In P-V, it was shown that outliers can be rejected to significantly improve the oscillations of ODT output by comparing ${}^{\{\mathcal{O}\}}\boldsymbol{T}_{\{\mathcal{B}\}}$ from (3.3) and from (3.2) after delay compensation. Consistency of these two appearances can be used as an index for the soundness of the outcome of the entire system. Moreover, wheel slippage or view occlusion of the camera for short periods of time are not a problem for this system. Compared to the example presented in Section 3.1, the pose of the box remains the same on the shelf no matter how the person moves, and as long as the person moves toward the shelf, the displacement appears in both the driven distance of the person and in the visual appearance of the box from the person's viewpoint. If the person closes their eyes for a few seconds, they are still able to guess their hand's position because of knowledge of arm motion.

In utilizing such an idea, there are benefits from some assumptions that do not limit applications similar to pallet picking or MMs for grasping. Because the forks of forklifts and grippers jaws of MMs are usually parallel bars, it is more likely to have the last part of a planned path be a straight line. This helps avoid nonlinearity in mapping between the coordinates because the majority of the coordinates have parallel rotation matrices. This happens at the same time as acquisition of better images because the robot is in the neighborhood of the object. These two effects are helpful for improving the performance of both the Avant forklift and mobile platform of the iMoro throughout the majority of a path in the neighborhood of an object. For a limited time interval, ODT output can correct sensor drift effects in ${}^{\{\mathcal{I}\}}\boldsymbol{T}_{\{\mathcal{B}\}}$ caused by Inertial Measurement Unit (IMU) or

wheel odometer. However, these two sensors can compensate lack of real-time ODT by generation of absolute pose feedback for hard real-time control loops using confidence measures.

3.3 Concept of Macro–Micro Multistage Controller for Mobile Manipulation (M5)

As discussed in the previous sections, the main idea behind the M4 was to consider separate roles for each part of the MM by taking into consideration their dissimilarities, as briefly addressed in Table 1.1. At the same time, the structure of existing controllers should be maintained, as shown by the blocks in Fig. 3.6 or Fig. 3.8. This simplified concept is illustrated in Fig. 3.3 and briefly described here. Theoretical details are addressed in more detail in publications P-I, P-II, and P-IV. Programming logic for implementation of these methods are addressed in P-VI.

In the M4, when the robot receives a message for a preplanned path that has a VS flag toward the object, in P-II or P-I this object was a pallet, it starts seeking for corresponding visual cues while driving on the global path. As soon as the estimation of the object pose becomes accurate enough, the M4 algorithms interfere with the input and output of the higher and lower level subsystems by mapping their feedback and set points into the local coordinates to define a local path. When the robot finishes the path, the pallet-picking task is accomplished. Participation of the manipulator in this task depends on error improvement. By default, the M4 does not allow the manipulator to track the target, and it remains in control of the global path planner as long as large error variations exist. As the robot arrives in the vicinity of the pallet with an appropriate heading, the manipulator and mobile base begin to coordinate. The macro robot stops coordination as soon as the motion requires better accuracy than it can provide.

M5 is more suitable for larger motion scales and high inertia test cases, for which the acceptable estimation of object pose is not convenient compared to the time or space needed for error correction of the mobile platform. One solution is to pause the M4 algorithm and request re-planning of the global path based on the updated pose of the target. However, as shown in P-IV, an intermediate phase can improve this situation. Since the concept described in Section 3.2 can be used to determine estimation quality, it was used to determine object visibility. The criteria for judgment were modified to keep only the position of the object detection tensor in the equations. Thus, mapping can be done only in the camera (or body) coordinates and not in global or local coordinates. The two main results of the intermediate phase are to enable VS methods for more heavy-duty autonomous vehicles and to avoid challenging the linearization assumptions of lower level controllers by smoothing the steering command. For Articulated Frame

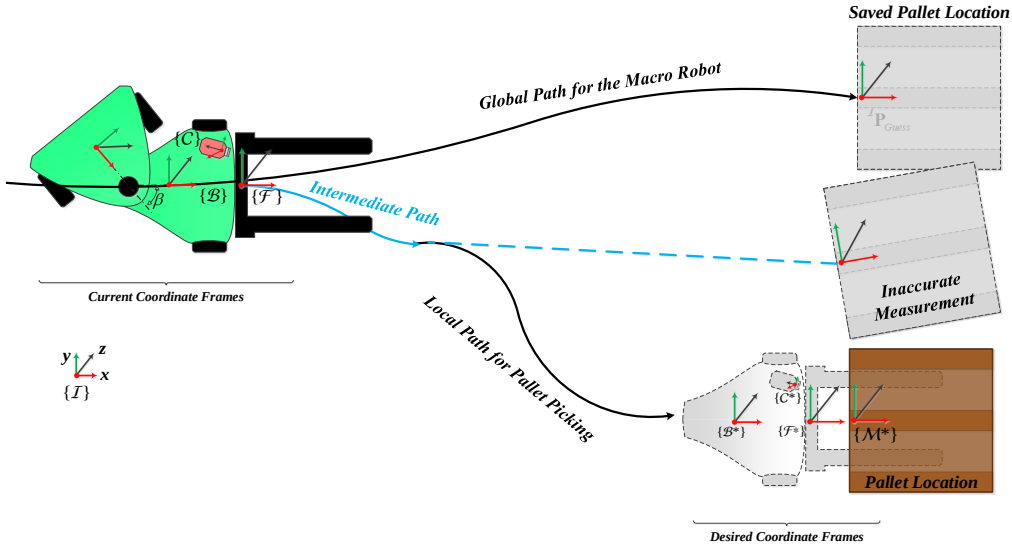


Figure 3.3: Three types of paths for pallet-picking experiments: the Macro–Micro Multistage Controller for Mobile Manipulation (M5) uses global, intermediate, and local paths, as addressed in P-IV; the Macro–Micro Controller for Mobile Manipulation (M4) only uses global and local paths as addressed in P-II.

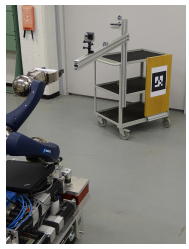
Steering (AFS) vehicles, speed of steering determines deviation in the Instantaneous Center of Rotation (ICR) for the front and rear parts of the vehicle, which can create instability in the basis of the system model.

3.4 Test Cases

The robots' hardware architecture is illustrated in Fig. 3.7 and Fig. 3.5 and includes two main networks: Controller Area Network (CAN) and Local Area Network (LAN). Communication between the time-critical components and hard real-time functionalities uses the CAN bus. For Avant, this network carries commands and feedback for hydraulic control valves and sensors. For the iMoro, it communicates with servo motor drives. Other real-time hardware are also on the same network, including sensors and manipulator joints. Other hardware components are connected through LAN in the physical layer. These network components are:

- the PC that hosts the remote development environment and Graphical User Interface (GUI),
- the Gigabit Ethernet Communication Protocol (GigE) camera,
- the image-processing PC,

- the real-time target, and
- the access point for wireless communications.



(a) iMoro, IHA mobile robot

(b) AVANT Autonomous forklift (GIM)

Figure 3.4: Test Cases

For generic toolboxes, such as image grabbing, marker detection, planning, and calibration, packages from the Robot Operating System (ROS) were used, as described in P-VI in more detail.

3.4.1 Autonomous GIM Machine (Avant Forklift)

In P-I, P-II, P-VI, and P-IV, the Avant autonomous forklift was used for experimentation, which shares its architecture and software modules with all the robots that use GIM. Avant has four wheels on an AFS mobile platform driven by hydrostatic power transmission and a 3-Degrees of Freedom (DOF) manipulator, as shown in Fig. 3.4b.

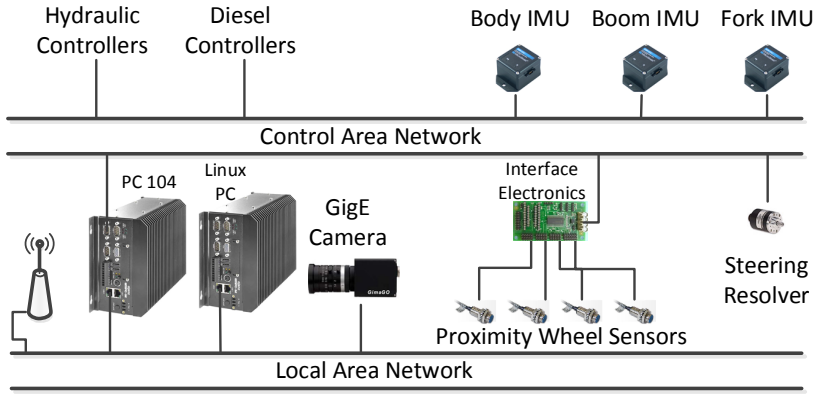


Figure 3.5: Hardware architecture for the iMoro as a common architecture for a MMs

3.4.2 iMoro Mobile Platform

As illustrated in Fig. 1.3, the test cases in P-V, P-III, and P-VII used the iMoro. The robot and its fundamental functionalities are described in P-III and [54]. These studies

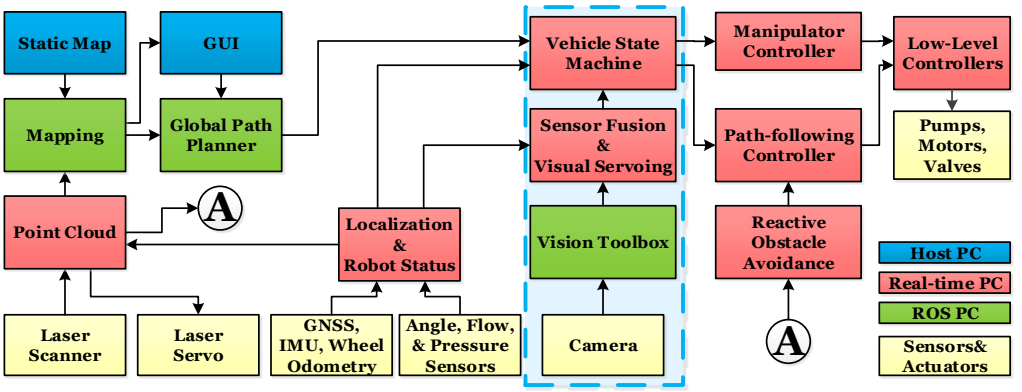


Figure 3.6: Software architecture for an outdoor hydraulic MM GIM. The focus area of this research is highlighted.

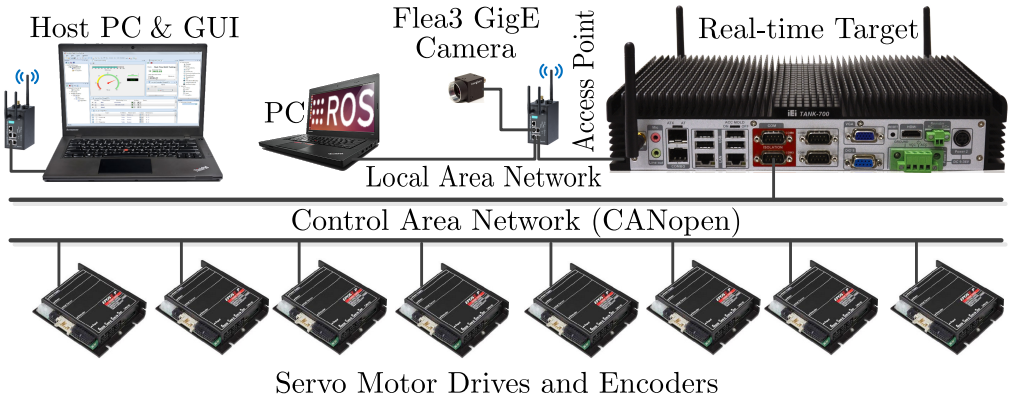


Figure 3.7: Common hardware architecture for a MMs

focused on integration of visual information into the control loop of the mobile platform's local coordinates.

Remote programming was done in the Simulink Real-time (formerly xPC-Target) environment or through MATLAB Code Generation using a Portable Operating System Interface (POSIX) implemented in a real-time Linux operating system, Xenomai, as explained in [58]. In both cases, host–target communication was based on Transmission Control Protocol (TCP). Other communications, such as updating the robot's status or image information were done through User Datagram Protocol (UDP) signals.

3.4.3 Practical Issues

iMoro, a service robot, and a GIM (Avant), an autonomous forklift vehicle, both use a nonholonomic steerable mobile platform, GigE camera, and wheel odometer sensors. However, there are significant differences; the details of each system have already been

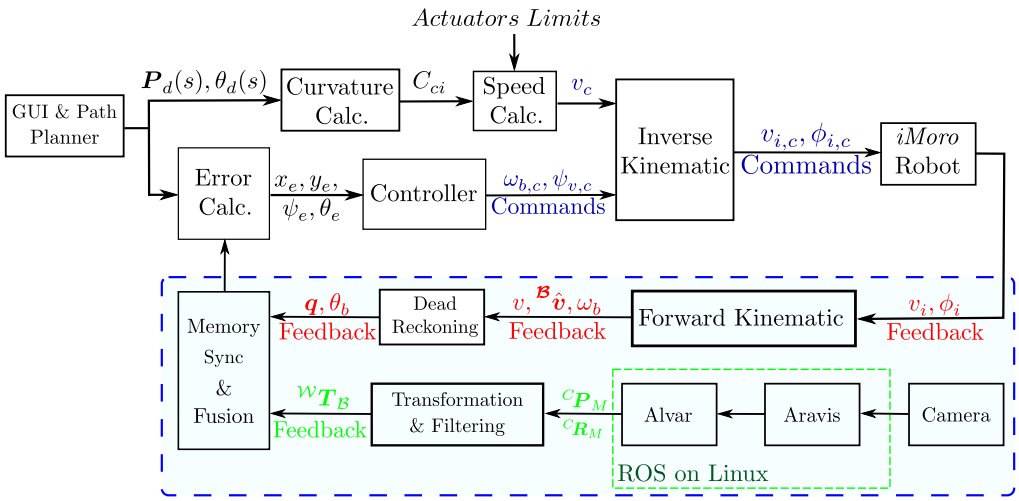


Figure 3.8: Data flow for visual guidance of indoor MMs, *i*Moro using ROS packages. The focus area of this research is highlighted.

explained in their relevant research papers, as referred in illustration of Fig. 1.3. The $k-$mean solution was sufficient for mapping between the robot's coordinates and the object coordinate in all test scenarios for the Avant (e.g., P-II,P-IV). To some extent, in near neighborhoods, the same solution was useful for the iMoro as presented in P-V. However, for a GNSS-denied environment, such as the iMoro's, the robot localized to ensure required accuracy. The robot's actuation benefited from brushless servo motors with significantly better steering bandwidth and accuracy compared to the actuators and wheel odometer in the Avant. The hydraulic machine handled both dry and Coulomb friction and high-inertia mechanical parts, while the indoor robot had a lower moving mass and better maintained gearboxes and driving system. Therefore, the common oscillations addressed in P-V or P-VII were dampened by system nonlinearities; thus, short-time jumps were small. However, in the iMoro, the control and actuation systems had faster responses and were more sensitive to process delays in images, errors, and discontinuities.

4 Discussion and Conclusion

This chapter explains the research conclusions, discussed the research limitations, and explains the research outcomes in comparison to relevant publications, divided by topic.

Due to the quantized nature of camera sensors and nonlinearities of optics, discontinuities occur frequently in the output of any vision system. High-resolution and expensive cameras can decrease the size of these jumps, but they do not resolve the issue.

4.1 Smoothness and Accuracy Contradictions

The simplest solution for oscillation of feedback for closed-loop systems is to lower the controller gains or utilize high-order filters. However, in both of the test cases, these solutions reduced system performance. The motion of the mobile platform must be accurate to ensure few positioning errors to maintain manipulator dexterity. Therefore, lowering system sensitivity is not a suitable option. Thus, smoothness becomes contradictory to accuracy for the overall outcome of the MM because the modules for image-grabbing, visual perception algorithms, and communication delay the ODT output and increase the raise time of the system by over-dampening the control loop, which is unacceptable due to possibility of significant changes in the system states during the delay time.

4.2 Macro–micro Coordination of Motion for a Vision-based MM

Although distribution of errors between macro and micro parts can be an independent problem, ODT properties can be considered to avoid undesirable motions. Generally speaking, micro robots are quick and accurate, as shown in Table 1.1, making them more prone to sensor noise than macro robots. Additionally, sudden changes in the direction of motion for nonholonomic mobile platforms cause steering errors because of low inertia motion, without significantly changing the center of mass. However, when the micro part the entire manipulator moves toward the command, which can cause oscillations in high inertia parts and rapid changes in the inertial forces of the center of mass of the

robot. These changes challenge the capabilities of the embedded power, especially for hydraulically-driven forklifts with diesel engines. Based on these issues, system integrity is better if only high-quality visual feedback is permitted to reach the micro robot. This underlines the importance of online assessment of measurement performance by comparing loop closures, as discussed in Section 3.2.

4.3 Demand for Long-range Object Detection

Quality of object detection by camera, necessary in time-of-flight cameras and range finders, depends on the distance between the object and the camera due to the limited angular resolution of sensors. If sensors can provide feedback with acceptable accuracy in a limited range, certain long-range applications for object detection can be achieved. For outdoor autonomous vehicles, it is common to have low degrees of maneuverability and steerability, which means that the vehicles require some driving distance to correct heading or side errors. For the Avant, this distance was about 3.5 m for a repeatability better than $\pm 7cm$. Thus, if the robot detected an object nearer than this distance, it was less likely to be able to pick it up from an appropriate angle. For indoor service robots, such as the iMoro, maneuverability and steerability are less challenging because these robots are designed for tight areas. However, they are usually used in GNSS-denied environments; therefore, cameras or other exteroceptive sensors are needed to localize the robot. This is more important if the task includes manipulation, obstacle avoidance, or fine adjustments of the mobile platform with respect to external objects. For the iMoro test case, the minimum distance was 2 m to accomplish positioning with repeatability better than $\pm 3cm$.

4.4 A Vision-based MM for Service Robotics and Field Robotics

Vision feedback provides considerable environmental data and, with internal sensors, it can increase the flexibility and functionality of an MM. This research showed how visual information could be integrated among existing subsystems in two autonomous mobile platforms. In addition to experimental challenges, there are a few interdisciplinary issues that should be mentioned. Exploiting vision or localization methods based on local observations can make the entire system's work range-dependent, partially because of the sensor structures, which are comparable to mapping using polar coordinates to Cartesian coordinates with finite amounts of pixels. Moreover, with respect to the camera coordinates shown in Fig. 3.2, position measurements along the $\hat{y}$ and $\hat{z}$, which are generally more accurate than the $\hat{x}$ (i.e., the depth of the camera image). However,

for orientation estimation, only rotations around the $\hat{x}$-axis benefit from these position measurements, while other rotations are prone to errors in depth estimation because estimation of rotation between two points (or features) needs comparison in at least two dimensions. This dependency is also affected by the size of objects or distances between matched features, which can improve or worsen the effect of depth estimation. All these dependencies highlight the sensitivity of orientation estimation of the depth of target objects, which is also made clear by comparison of covariance ellipsoids that become larger at farther distances.

Additionally, mapping between coordinates can cause propagation of uncertainties. For instance, to localize the robot's manipulator with respect to the object, the inverse of the transformations previously introduced, such as ${}^{\mathcal{M}}\boldsymbol{T}_{\mathcal{O}}$, are inversed and become ${}^{\mathcal{O}}\boldsymbol{T}_{\mathcal{M}}$, which is equal to

$$ {}^{\mathcal{M}}\boldsymbol{T}_{\mathcal{O}}^{-1}(x,y,z,\varphi,\theta,\psi) = \left[\begin{array}{ccc|c} & {}^{\mathcal{M}}\boldsymbol{R}_{\mathcal{O}}^{T}(\varphi,\theta,\psi) & & -\,{}^{\mathcal{M}}\boldsymbol{R}_{\mathcal{O}}^{T}(\varphi,\theta,\psi)\;{}^{\mathcal{M}}\boldsymbol{P}_{\mathcal{O}}(x,y,z) \\ \hline 0 & 0 & 0 & 1 \end{array}\right]_{4\times 4} . \tag{4.1} $$

This shows how the position data are more prone to orientation detection when providing feedback to the controller. If measurements in the $y-z$ plane were better in accuracy, this multiplication negatively affects spatial representations of motions.

4.5 Conclusion and Remaining Challenges

For simplified assumptions of constant uncertainty, a system can become too conservative unless another measure to reassess uncertainties is used, such as in P-VII. In P-IV, this issue was solved by proposing the M5, which introduced an intermediate stage that only used the position elements of the tensor to move the robot toward the object. Because the ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{O}}$ estimates were consistent, as addressed in 3.2, allowing them to be used as quality measures, the local coordinates in the $\{\mathcal{W}\}$ were used after reliable feedback was received for ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{O}}$. Both of these methods maintained the quality of visual feedback, as shown in P-V (for the iMoro) and P-II (for the Avant), while the robot and camera were moving at far distances.

Integration of vision into an MM was also analyzed and implemented, and integration of information from the MM can be incorporated in perception tools. This process can be applied using the same method as that used for detailed image features and Simultaneous Localization and Mapping (SLAM) outputs. The filter introduced in P-VII was utilized for each axis and extension of that with vector implementations of the proposed algorithm together with deep learning methods, such as [22], or transfer learning can provide the

network with motion primitives to be trained together with the image features. Further analysis and formulation of uncertainties should be conducted to include better vector representations of spatial motion estimations. Then, analytical formulations of the entire system could be developed to define certain switching and error distributions in geometric, frequency, and time domains with parameterization of the uncertainties. The idea can be representation of nonholonomic constraints as delay that steering can cause to steer toward the desired velocity vector. This is an important direction for future research to support system integration for nonholonomic dynamics.

5 Summary of Publications

This chapter summarizes each publication of this thesis, and answering the research problems were the primary aim of these publications. As previously addressed in Figure 1.3, there were two variations of test cases. The hardware descriptions in P-I and P-VI addressed the architecture of the Avant autonomous forklift, and P-III and [54] addressed the iMoro mobile platform. The solutions provided in P-I, P-II, P-VI, and P-IV were more suitable for field robots, while solutions for challenges highlighted in P-III, P-V, and P-VII were more suitable for service robots.

5.1 Summary of P-I: Position-based Visual Servoing for Pallet Picking by an Articulated Frame-steering Hydraulic Mobile Machine

This paper discussed the first implementation of the visual guidance method developed for the Avant, which resulted in an articulated frame-steering forklift driven by a hydrostatic circuit (Fig. 3.4b). The kinematics and hardware architecture of the autonomous system were described, focusing on sensory information and specifications as well as actuation accuracy for driving the mobile platform. Due to limitations caused by outdoor conditions, wheel encoders consisted of two hall sensors, for which output was invalid for low-speed motions; therefore, the experiment was the first answer for challenges in facing the use of mobile platforms and manipulators together. Such differences were then considered to develop a solution to the control problem by dividing APP into two subspaces that shared an x-axis. This initial solution addressed both research questions, RP.I and RP.II, and was specific to real-world conditions. Based on common practices in grasp planning, a parametric curve was proposed to convert updates with discontinuities into one smooth path for pallet picking.

5.2 Summary of P-II: A Macro–Micro Controller for Pallet Picking Using an Articulated Frame-steering Hydraulic Mobile Machine

This paper proposed an architecture to define a MM, in this case a forklift, as a macro–micro system, namely M4. Despite a challenging application, by formulating an M4, significant differences between the mobile platform and manipulation mechanism were addressed. Practically, given the coordinate frames shown in Fig. 3.2 and Table 3.1, this method provided the autonomous forklift with an APP range of 3–4 m pallet-picking distance in the x-direction and a $\pm 0.5m$ distance in the y-direction. Additionally, this paper addressed the problem of visual servoing for pallet picking and the need to configure multiplication for a homogeneous transformation matrix by reconstructing equations similar to 3.3. Doing this, the problem can be represented as an M4 problem because the macro part carries the micro part of the MM, for which, general properties are addressed as presented in Table 1.1.

Based on this representation, visual features are artificially generated for the macro part based on imposition of idealistic motion for the micro part and vice versa. Thus, the simplification and limitations of the problem only influence the reachable directions of the macro part and prevent cyclic bouncing that can delay hydraulic actuators and create imprecise measurements. Additionally, the manipulator was prevented from performing unnecessary movements because of inadequate detection quality, possibility of camera occlusion. This method is robust for single jumps in vision data and short-term occlusions.

5.3 Summary of P-IV: A Multistage Controller with Smooth Switching for Autonomous Pallet Picking

The main contribution of this paper is an improvement of M4 pallet-picking method by introducing Macro–Micro Multistage Controller for Mobile Manipulation (M5). In addition to cooperation between the macro and micro parts, the field robot's macro–micro behavior was formulated based on changes in visual feedback uncertainty to allow the pallet-picking process to begin from farther distances from the pallet (e.g., 4–5 m) using imperfect visual information.

An analysis of experimental errors presented in P-I and P-II highlights two contradictory issues. To a certain extent, vision output improves when a robot is close to a target; however, an AFS vehicle is not capable of fast corrections for lateral deviations from the path or of following high-curvature paths. Generally speaking, car-like or AFS vehicles are limited by a small turning radius, unlike differentially driven mobile platforms. This limitation increases when a robot carries a camera and must perform sharp steering,

which is more likely to occur when a pallet (i.e., a target) is out of the camera's Field of View (FOV). To overcome these issues, this papers proposed an initial position estimation as soon as vision feedback provides consistent output, which allows the robot to incorporate both orientation and position information. Practically, the intermediate stage only works at distances of $0.5\check{}1m$; however, this does increase robot functionality and robustness such that it can tolerate one meter more than the robot in P-II in each direction. This considerable heading change improves critical curvatures on the planned path and target visibility.

5.4 Summary of P-III: Fault Tolerant Control Architecture Design for Mobile Manipulation in Scientific Facilities

This paper introduced the hardware and software architecture for the iMoro and described the needs, requirements, and conceptual designs for MMs and remote handling in scientific facilities, based on [54]. It described how to break down a mission into different tasks, and which subsystems are necessary for the autonomous functionalities of a service robot for inspection and manipulation in restricted areas. Scientific facilities, such as CERN , have high-tech and expensive materials and test areas, in which human intervention can be costly and must be As Low As Reasonably Achievable (ALARA) [9] or As Low As Reasonably Practicable (ALARP) [24].

Based on the common conditions of scientific facilities, if a robot is designed to reduce human intervention, it must guarantee minimum functionality; thus, a service robot, including the iMoro, should guarantee that it can exit the test bed, in this case CERN tunnels, in case of mission failure. Because remaining as an external object in a test environment is unacceptable due to the physical or environmental limitations of scientific tests. Thus, this paper proposed a parallel logic and architecture, namely a safety anchor, that guarantees minimum functionality. Since the proposed method has limited states and conditions, it can achieve the third Safety Integrity Level (SIL)[59] certification because it implements safety components.

The paper also provided an analysis of the kinematic formulation of the robot to show simplified actuation of the mobile platform without a path-following controller and to use the fault-tolerant method during mission failure to extract body inclination data from steering and driving wheel states in one robot leg to prevent corruption of inverse kinematics. Since the test case mobile platform is Four-Wheel Steerable (4WS), it has at least one redundant leg; therefore, the experimental results showed reliable outcomes even if one of the legs does not perform well during synchronized motion.

5.5 Summary of P-V: Real-Time Vision-Based Navigation for Nonholonomic Mobile Robots

The solutions of RP.I and RP.II were briefly explained in previous papers for the Avant test case. This paper experimentally verified these ideas on an indoor robot test case using the iMoro to extend these ideas to RP.II to gain better integration of visual feedback in the control loop. It was shown that an eye-in-hand setup for a mobile platform can be prone to synchronization errors even when common issues in visual feedback are ignored. Thus, if the real-time PC receives ${}^{\{\mathcal{C}\}}\boldsymbol{T}_{\{\mathcal{W}\}}$ (as explained in Table 3.1) at time t, data belong to the camera frame at time $t - t_d$, where t_d stands for variable delays caused by processes, image grabbing, buffering, and communication in order of their role. This means that the transformation matrix represents an object's pose with respect to an imaginary camera behind the robot t_d seconds in the past. This effect proportionally depends on a robot's velocity history for the past t_d seconds, which is converted from a timing error into a positioning error.

The paper compensated for this error and proposed improvements for positioning accuracy based on the concept represented in Eq. (3.2). The results showed the importance of synchronization for high-speed motions of mobile bases. Thus, the solution made it possible for the robot to grab an object with tolerances better than $3cm$ solely using the motion control of the mobile platform based on camera and wheel feedback.

5.6 Summary of P-VI: Vision-guided Autonomous Forklift

This paper targeted the architectural design and system structure of the Avant, which is shown in Fig. 3.6. As an autonomous forklift, Avant's motion control subsystem is commanded by a GUI and a global path planner. The control system receives feedback from estimation and localization data based on a GNSS, IMUs, and wheel odometer, as well as command actuator-level controllers. Safety or obstacle avoidance systems can override the actions of this control system; thus, this paper proposed a Vehicle State Machine (VSM) to manage these operations and, at the same time, exploit these systems to convert the machine into a flexible AGV. Based on certain rules, an event-based work flow design was proposed to utilize pre-existing systems as a visual servoing controller.

The proposed multistate VSM used a three-step process of preprocessing, command processing, and post processing. The paper proposed 3 – 4 rules for each step to prevent interference of roles in each subsystem. This VSM connected the vehicle subsystems to the higher level planner and monitoring modules. The interface flagged higher level modules, which was used for long-distance communication consisting of

1. **Busy** the component is running a command or not available.
2. **Free** ($\neq$ Busy) the component is available for the next task.

Internal flags can use more accurate descriptions for better coordination of actions among lower level modules. Such flags include the following.

1. **Active** The module can receive a command according to its internal state and synchronization criteria with other modules.
2. **Done** The module's assigned task is accomplished, or it is no longer valid.
3. **Release** Accomplishment can be declared. The system is no longer busy with the task.
4. **Wait** The module cannot be tasked due to its condition or because of interference with other modules.

Based on these flags, the VSM runs the planned actions in order; thus, if a pallet-picking task is implemented in the system, the vision toolbox will be activated, and driving will continue in a saved pose. As soon as the object is detected in the neighborhood, the localization feedback will be overridden by the vision feedback causing driving and manipulation modules to coordinated actions.

5.7 Summary of P-VII: Latencies and Noise Effects in Vision-Based Control of Mobile Robots

Based on the previous experiences explained in P-II, P-IV, and P-V, the efficacy and workflow for system decomposition was presented before this study. Using the same architecture and formulation, it was possible to analyze measurement effects more accurately. Mobile platforms are prone to shakiness due to imperfect localizations P-VI. Any assumptions of continuity and well-behaved feedback in the MM control design are questioned due to jumps of localization in indoor (e.g., P-V) and outdoor (e.g., P-I) conditions. The methods addressed in the previous publications were more suited to outdoor conditions, in which robustness imposes challenges for positioning precision. To extend the findings of P-IV to indoor service robots (e.g., the iMoro), the fusion method proposed in this paper provides significant improvements. Commonly, service robots have better interceptive information and lack of GNSS and the absolute positions **BOEX!** (**BOEX!**). Therefore, a method was proposed to integrate visual feedback from long distances that were smoothened by wheel odometer data to provide the control system with hard real-time absolute position and orientation feedback. As shown in Figure 3.4a, the iMoro was capable of repeatedly

gripping a gauge below 3^{cm} using only mobile platform motion. While the ODT camera was working at a rate of 8 FPS, this method was capable of generating feedback for a 200 Hz control system. By accurately rejecting outliers and considering range-dependent noise, the mobile robot was capable of localizing itself with respect to a detected object within 5^{m}, with negligible oscillating behavior.

Bibliography

[1] Acevedo, J. J., García, M., Viguria, A., Ramón, P., Arrue, B. C., and Ollero, A., "Autonomous landing of a multicopter on a moving platform based on vision techniques," *ROBOT 2017: Third Iberian Robotics Conference: Volume 2*, pp. 272 – 282, 2018.

[2] Allibert, G., Courtial, E., and Chaumette, F., "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933 – 939, 2010.

[3] Astola, P., Helin, P., M. Aref, M., Mattila, J., Astola, J., and Tabus, I., "Precise outline matching criteria for target pose estimation and odometry from stereo video," *39th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 724 – 730, 2016.

[4] Astola, P., M. Aref, M., Vihonen, J., Mattila, J., and Tabus, I., "Object detection in robotic applications for real-time localization using semi-unknown objects," *40th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 682 – 687, 2017.

[5] Bag, S., Venkatachalapathy, V., and Ptucha, R. W., "Motion estimation using visual odometry and deep learning localization," *Electronic Imaging*, vol. 2017, no. 19, pp. 62 – 69, 2017.

[6] Baglivo, L., Biasi, N., Biral, F., Bellomo, N., Bertolazzi, E., Da Lio, M., and De Cecco, M., "Autonomous pallet localization and picking for industrial forklifts: a robust range and look method," *Measurement Science and Technology*, vol. 22, no. 8, p. 085502, 2011.

[7] Ban, K., Warashina, F., Yamada, M., and Namiki, Y., "Robot system comprising visual sensor," Dec. 4, 2012, US Patent 8,326,460.

[8] Bischoff, R., Huggenberger, U., and Prassler, E., "Kuka youBot – a mobile manipulator for research and education," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1 – 4, 2011.

[9] Bonnal, P., Baudin, M. O., and Ruiz, J.-M., "Systems engineering and safety issues in scientific facilities subject to ionizing radiations," *International Journal of Advanced Robotic Systems*, vol. 10, no. 10, p. 366, 2013.

[10] Brock, O., Park, J., and Toussaint, M., "Mobility and manipulation," *Springer Handbook of Robotics*, 2016.

[11] Byun, S. and Kim, M., "Real-time positioning and orienting of pallets based on monocular vision," *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, vol. 2, pp. 505 – 508, 2008.

[12] Cárdenas, A., Goodwine, B., Skaar, S., and Seelinger, M., "Vision-based control of a mobile base and on-board arm," *International Journal of Robotics Research*, vol. 22, no. 9, pp. 677 – 698, 2003.

[13] Chaumette, F. and Hutchinson, S., "Visual servo control. I. basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82 – 90, 2006.

[14] – – –, "Visual servo control. II. advanced approaches [tutorial]," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109 – 118, 2007.

[15] Chaumette, F., "Potential problems of stability and convergence in image-based and position-based visual servoing," pp. 66 – 78, 1998.

[16] Chen, H.-Y., Bell, Z., Licitra, R., and Dixon, W., "A switched systems approach to vision-based tracking control of wheeled mobile robots," *IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 4902 – 4907, 2017.

[17] Chesi, G. and Vicino, A., "Visual servoing for large camera displacements," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 724 – 735, 2004.

[18] De Cristóforis, P., Nitsche, M., Krajník, T., Pire, T., and Mejail, M., "Hybrid vision-based navigation for mobile robots in mixed indoor/outdoor environments," *Pattern Recognition Letters*, vol. 53, pp. 118 – 128, 2015.

[19] Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D., and Djemaï, M., "Navigation scheme with priority-based scheduling of mobile agents: application to AGV-based flexible manufacturing system," *Journal of Intelligent & Robotic Systems*, vol. 82, no. 3, pp. 495 – 512, 2016.

[20] Donaire, A., Romero, J. G., Perez, T., and Ortega, R., "Smooth stabilisation of nonholonomic robots subject to disturbances," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4385 – 4390, 2015.

[21] Dönmez, E., Kocamaz, A. F., and Dirik, M., "A vision-based real-time mobile robot controller design based on gaussian function for indoor environment," *Arabian Journal for Science and Engineering*, pp. 1 – 16, 2017.

[22] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T., "Flownet: Learning optical flow with convolutional networks," *IEEE International Conference on Computer Vision*, pp. 2758 – 2766, 2015.

[23] Dounskaia, N., "Control of human limb movements: the leading joint hypothesis and its practical applications," *Exercise and sport sciences reviews*, vol. 38, no. 4, p. 201, 2010.

[24] Fabry, T., Vanherpe, L., Feral, B., and Braesch, C., "Developing an interactive intervention planner - a systems engineering perspective," *International Journal of Advanced Robotic Systems*, vol. 10, no. 9, p. 335, 2013.

[25] Fei-Fei, L., "ImageNet: crowdsourcing, benchmarking & other cool things," *CMU VASC Seminar*, vol. 16, pp. 18 – 25, 2010.

[26] Fiala, M., "ARTag, a fiducial marker system using digital techniques," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 590 – 596, 2005.

[27] Frazzoli, E. and Iagnemma, K., "Facilitating vehicle driving and self-driving," May 9 2017, US Patent 9645577B1.

[28] Froehlich, T. and Reiser, U., "Design and implementation of a spherical joint for mobile manipulators," *Proceedings of ISR 2016: 47st International Symposium on Robotics*, pp. 1 – 8, 2016.

[29] Ghabcheloo, R., Hyvönen, M., Uusisalo, J., Karhu, O., Järä, J., and Huhtala, K., "Autonomous motion control of a wheel loader," *ASME Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control*, Hollywood, CA, US, 2009.

[30] Hermann, M., Pentek, T., and Otto, B., "Design principles for industrie 4.0 scenarios," *49th Hawaii International Conference on System Sciences (HICSS)*, pp. 3928 – 3937, 2016.

[31] Holeva, L. F., Elston, E. R., Seelinger, M. J., and Yoder, J.-D. S., "Identifying and selecting objects that may correspond to pallets in an image scene," May 5, 2015, US Patent 9,025,886.

[32] Holmberg, R. and Khatib, O., "Development and control of a holonomic mobile robot for mobile manipulation tasks," *The International Journal of Robotics Research*, vol. 19, no. 11, pp. 1066 – 1074, 2000.

[33] Hutchinson, S., Hager, G., and Corke, P., "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651 – 670, 1996.

[34] Janabi-Sharifi, F. and Marey, M., "A kalman-filter-based method for pose estimation in visual servoing," *IEEE Transactions Robotics*, vol. 26, no. 5, pp. 939 – 947, 2010.

[35] Khatib, O., Yokoi, K., Chang, K., Ruspini, D., Holmberg, R., and Casal, A., "Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 546 – 553, 1996.

[36] Khatib, O., Yokoi, K., Brock, O., Chang, K., and Casal, A., "Robots in human environments: Basic autonomous capabilities," *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 684 – 696, 1999.

[37] L'Annunziata, M. F., "Chapter 21 - hall of fame: Part x," in *Radioactivity (Second Edition)*. Boston: Elsevier, 2016, pp. 729 – 798.

[38] Lecking, D., Wulf, O., and Wagner, B., "Variable pallet pick-up for automatic guided vehicles in industrial environments," *IEEE Conference onEmerging Technologies and Factory Automation, (ETFA'06)*, pp. 1169 – 1174, 2006.

[39] Lippiello, V., Ruggiero, F., Siciliano, B., and Villani, L., "Visual grasp planning for unknown objects using a multifingered robotic hand," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 1050 – 1059, 2013.

[40] M. Aref, M., Astola, P., Vihonen, J., Tabus, I., Ghabcheloo, R., and Mattila, J., "Adaptive feedback in local coordinates for real-time vision-based motion control over long distances," *International Conference on Robotics and Mechatronics*, vol. 320, no. 1, p. 012009, 2018.

[41] M. Aref, M., Oftadeh, R., Koivumäki, J., Ghabcheloo, R., and Mattila, J., "System requirement document for modular mobile manipulator system," *PURESAFE System Engineering Documents*, vol. WP3, no. RP10-RP14, 2012.

[42] M. Aref, M., Ghabcheloo, R., Kolu, A., Hyvönen, M., Huhtala, K., and Mattila, J., "Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics, Automation and Mechatronics (RAM), Manila, Philippine*, pp. 218 – 224, 2013.

[43] M. Aref, M., Ghabcheloo, R., and Mattila, J., "A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong*, pp. 6816 – 6822, 2014.

[44] M. Aref, M., Oftadeh, R., Ghabcheloo, R., and Mattila, J., "Fault tolerant control architecture design for mobile manipulation in scientific facilities," *International Journal of Advanced Robotic Systems*, vol. 12, p. 4, 2015.

[45] M. Aref, M., Ghabcheloo, R., Kolu, A., and Mattila, J., "A multistage controller with smooth switching for autonomous pallet picking," *IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden*, pp. 2535 – 2542, 2016.

[46] M. Aref, M., Ghabcheloo, R., and Mattila, J., "Real-time vision-based navigation for nonholonomic mobile robots," *IEEE International Conference on Automation Science and Engineering (CASE)*, 2016.

[47] M. Aref, M., Ghabcheloo, R., Kolu, A., and Mattila, J., "Vision guided autonomous forklift," *Advances in Robot Design and Intelligent Control (RAAD 2016)*, pp. 338 – 346, 2017.

[48] M. Aref, M., Vihonen, J., Ghabcheloo, R., and Mattila, J., "On latencies and noise effects in vision-based control of mobile robots," *Advances in Service and Industrial Robotics*, pp. 191 – 199, 2018.

[49] Malis, E., Chaumette, F., and Boudet, S., "$2\frac{1}{2}$D visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238 – 250, 1999.

[50] Menhour, L., D'Andréa-Novel, B., Fliess, M., Gruyer, D., and Mounier, H., "An efficient model-free setting for longitudinal and lateral vehicle control: Validation through the interconnected pro-sivic/rtmaps prototyping platform," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[51] Oftadeh, R., "Universal path following of wheeled mobile robots," Ph.D. dissertation, Tampere University of Technology, 2016.

[52] Oftadeh, R., M. Aref, M., Ghabcheloo, R., and Mattila, J., "Bounded-velocity motion control of four wheel steered mobile robots," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Australia*, pp. 255 – 260, 2013.

[53] ———, "Mechatronic design of a four wheel steering mobile robot with fault-tolerant odometry feedback," *IFAC Mechatronic Systems*, vol. 1, no. 1, pp. 663–669, 2013.

[54] ———, "Real-time system integration for mobile manipulation," *International Journal of Advanced Robotic Systems*, vol. 11, no. 15, 2013.

[55] Oftadeh, R., Ghabcheloo, R., and Mattila, J., "Time optimal path following with bounded velocities and accelerations for mobile robots with independently steerable wheels," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2925–2931, 2014.

[56] ———, "A time-optimal bounded velocity path-following controller for generic wheeled mobile robots," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 676–683, 2015.

[57] ———, "A novel time optimal path following controller with bounded velocities for mobile robots with independently steerable wheels," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4845–4851, Japan, 2013.

[58] Oftadeh, R., M. Aref, M., Ghabcheloo, R., and Mattila, J., "Unified framework for rapid prototyping of Linux based real-time controllers with Matlab and Simulink," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 274–279, Taiwan, 2012.

[59] Redmill, F., "Understanding the use, misuse and abuse of safety integrity levels," *Proceedings of the Eighth Safety-critical Systems Symposium*, pp. 8–10, 2000.

[60] Reiser, U., Connette, C., Fischer, J., Kubacki, J., Bubeck, A., Weisshardt, F., Jacobs, T., Parlitz, C., Hägele, M., and Verl, A., "Care-o-bot® 3-creating a product vision for service robot applications by integrating design and technology," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1992–1998, 2009.

[61] Roa, M. A., Berenson, D., and Huang, W., "Mobile manipulation: toward smart manufacturing," *IEEE Robotics and Automation Magazine*, vol. 22, no. 4, pp. 14–15, 2015.

[62] Röwekämper, J., Sprunk, C., Tipaldi, G. D., Stachniss, C., Pfaff, P., and Burgard, W., "On the position accuracy of mobile robot localization based on particle filters combined with scan matching," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3158–3164, 2012.

[63] Seelinger, M. and Yoder, J.-D., "Automatic pallet engagment by a vision guided forklift," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4068 – 4073, 2005.

[64] – – –, "Automatic visual guidance of a forklift engaging a pallet," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 1026 – 1038, 2006.

[65] Seelinger, M., Yoder, J.-D., Baumgartner, E. T., and Skaar, S. B., "High-precision visual control of mobile manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 6, pp. 957 – 965, 2002.

[66] Seelinger, M. J., Yoder, J.-D. S., and Skaar, S. B., "Mobile camera-space manipulation," Feb. 27, 2001, US Patent 6,194,860.

[67] Seraji, H., "A unified approach to motion control of mobile manipulators," *The International Journal of Robotics Research*, vol. 17, no. 2, pp. 107 – 118, 1998.

[68] Shneier, M. and Bostelman, R., "Review of research for docking automatic guided vehicles and mobile robots," National Institute of Standards and Technology (NIST), US Department of Commerce, Tech. Rep. NISTIR 8140, October 2016.

[69] Sprunk, C., Lau, B., Pfaff, P., and Burgard, W., "An accurate and efficient navigation system for omnidirectional robots in industrial environments," *Autonomous Robots*, pp. 1 – 21, 2016.

[70] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. *et al.*, "Going deeper with convolutions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[71] Tamba, T. A., Hong, B., and Hong, K.-S., "A path following control of an unmanned autonomous forklift," *International Journal of Control, Automation and Systems*, vol. 7, no. 1, pp. 113 – 122, 2009.

[72] Teller, S., Walter, M. R., Antone, M., Correa, A., Davis, R., Fletcher, L., Frazzoli, E., Glass, J., How, J. P., Huang, A. S. *et al.*, "A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 526 – 533, 2010.

[73] Umlauft, J., Sieber, D., and Hirche, S., "Dynamic movement primitives for cooperative manipulation and synchronized motions," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 766 – 771, 2014.

[74] Villagra, J. and Herrero-Pérez, D., "A comparison of control techniques for robust docking maneuvers of an AGV," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 4, pp. 1116 – 1123, 2012.

[75] Walter, M. R., Karaman, S., Frazzoli, E., and Teller, S., "Closed-loop pallet manipulation in unstructured environments," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5119 – 5126, 2010.

[76] Walter, M. R., Antone, M., Chuangsuwanich, E., Correa, A., Davis, R., Fletcher, L., Frazzoli, E., Friedman, Y., Glass, J., How, J. P. *et al.*, "A situationally aware voice-commandable robotic forklift working alongside people in unstructured outdoor environments," *Journal of Field Robotics*, vol. 32, no. 4, pp. 590 – 628, 2015.

[77] Wang, K., Ding, N., and Dai, F., "Visual servoing based pickup of stationary objects with a dynamically controlled manipulator," *IEEE International Conference on Industrial Technology (ICIT)*, pp. 902 – 907, 2017.

[78] Wang, R., Zhang, X., Fang, Y., and Li, B., "Visual servoing of mobile robots with input saturation at kinematic level," *International Conference on Image and Graphics:ICIG, Shanghai, China, Revised Selected Papers, Part I*, pp. 432 – 442, 2017.

[79] Wilson, W. J., Hulls, C. W., and Bell, G. S., "Relative end-effector control using cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684 – 696, 1996.

[80] Wit, J., Crane, C. D., and Armstrong, D., "Autonomous ground vehicle path tracking," *Journal of Field Robotics*, vol. 21, no. 8, pp. 439 – 449, 2004.

[81] Xu, D., Lu, J., Wang, P., Zhang, Z., and Liang, Z., "Partially decoupled image-based visual servoing using different sensitive features," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2233 – 2243, 2017.

[82] Yoder, J.-D., West, J., Baumgartner, E., Perrollaz, M., Seelinger, M., and Robinson, M., "Experiments comparing precision of stereo-vision approaches for control of an industrial manipulator." Springer International Publishing, 2013, pp. 245 – 256.

[83] Zhang, X., Fang, Y., and Liu, X., "Motion-estimation-based visual servoing of nonholonomic mobile robots," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1167 – 1175, 2011.

[84] Zhang, X., Fang, Y., Li, B., and Wang, J., "Visual servoing of nonholonomic mobile robots with uncalibrated camera-to-robot parameters," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 390 – 400, 2017.

Publications

Publication I

M. Aref, M., Ghabcheloo, R., Kolu, A., Hyvönen, M., Huhtala, K., and Mattila, J., "Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics, Automation and Mechatronics (RAM), Manila, Philippine*, pp. 218 – 224, 2013

Position-Based Visual Servoing for Pallet Picking by an Articulated-frame-steering Hydraulic Mobile Machine

Mohammad M. Aref, Reza Ghabcheloo, Antti Kolu, Mika Hyvönen, Kalevi Huhtala, Jouni Mattila

***Abstract*— This paper addresses a visual servoing problem for a mobile manipulator. Specifically, it investigates pallet picking by using visual feedback using a fork lift truck. A manipulator with limited degrees of freedom and differential constraint mobility together with large dimensions of the machine require reliable visual feedback (pallet pose) from relatively large distances. To address this challenge, we propose a control architecture composed of three main sub-systems: (1) pose estimation: body and fork pose estimation in the pallet frame; (2) path planning: from the current pose to the origin (pallet frame); and (3) feedback motion control. In this architecture, the pallet becomes the local earth fixed frame in which poses are resolved and plans are formulated. Choosing the pallet as the origin provides a natural framework for fusing the wheel odometry/inertial sensor data with vision, and planning is required only once the pallet is detected for the first time (because the target is always the origin). Visual pallet detection is non-real-time and unreliable, especially owing to large distances, unfavorable vibrations, and fast steering. To address these issues, we introduce a simple and efficient method that integrates the vision output with odometry and realizes smooth and non-stop transition from global navigation to visual servoing. Real-world implementation on a small-sized forklift truck demonstrates the efficacy of the proposed visual servoing architecture.**

I. Introduction

Visual Servoing (VS) or vision-based robot control found a wide range of applications such as humanoid robots and robotic manufacturing. VS is usually used in object detection for manipulation or target tracking. In this study, we address the problem of VS for a mobile manipulation task, namely, pallet picking by a forklift truck. Because bulky and massive materials are commonly handled on a pallet, automatic pallet picking by a forklift truck has attracted considerable research attention. The main challenge in this task is the reliable detection of the pallet while eliminating typical failures associated with vision systems. To increase the robustness of vision, previous studies have mostly used auxiliary sensors such as laser pointers [1] or laser scanners [2]; another study [3] applied a neural extended Kalman filter for object detection. In this study, we propose a fault-tolerant pallet pose (position and orientation) estimation in which vision data is fused with odometry and inertial measurements. The estimation output is then used for VS and motion control.

VS can be classified into three categories: *Image-Based Visual Servoing* (IBVS), *Position-Based Visual Servoing* (PBVS), and a hybrid of both. IBVS has been studied most widely; it is based on the comparison of a desired image pose and the current image to produce feedback [4]. Although it is computationally expensive, it can be run in real-time through certain approaches [5]. It is best suited to two dimensional problems with initial knowledge of the depth or scale [6] without having a large rotational movement [7]. IBVS is mostly used for short distances and eye-in-hand applications [8]. On the other hand, PBVS is based on the estimation of the target pose using image features. Because pose estimations (e.g., in Cartesian space) are provided in PBVS, the vision output can be readily integrated with other sensor modalities. However, the calibration of the extrinsic parameters of the camera with respect to the robot body plays a significant role in the success of manipulation [9]. A hybrid method such as $2\frac{1}{2}$ D VS [9] combines both IBVS and PBVS methods to extract the translational movement of a camera from the PBVS method and its rotation in a two-dimensional image coordinate system [10], [7]. Several studies have been reported in this regard [10], [7], and an earlier one [11].

The present study focuses on mobile manipulation and fusion of vision pose estimates with inertial sensors. We have used an on-board monocular camera in combination with fiducial markers on the pallet. The pose of the markers is computed using the Alvar computer vision library [12] in a non-real-time fashion. The integration of inertial sensors with vision provides real-time pallet pose estimation even in the event of extended delays in visual pallet detection or when the view to the markers is blocked by the boom and fork. The proposed method is general and should be independent of the computer vision method used to detect the pallet pose.

Generally, in the PBVS method, an error vector based on the current pose (from vision) and the desired pose of the end effector (or the work tools) is calculated, and proper control signals are generated to drive the error to zero. This error can be defined in either the image frame, Cartesian body frame, or both [13], [14], [15]. Contrary to common practice, in this study, we propose a control architecture in which the pallet frame plays the role of the reference frame or the origin with respect to which all the errors are defined. After the pallet is robustly detected for the first time (as explained later), the control system switches to the pallet frame as a local frame. Therefore, the target pose remain the same (i.e., the origin) throughout VS. Our control architecture comprises three main subsystems: (1) pose estimation: estimation of the body and fork in the pallet frame; (2) path planning: planning a route that leads from the current location to the origin; and (3) motion control: path following and manipulator control. Defining the pallet as the origin has the advantage that path planning is performed only once at the start when the pallet has been detected, unlike in previous studies [16], [17].

Authors are with Intelligent Hydraulics and Automation Department, Tampere University of Technology, 33101, Finland.
Email: m.aref@ieee.org .

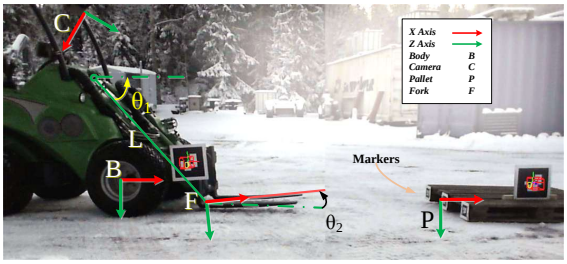


Fig. 1. The attached coordinate frames

The remainder of this paper is organized as follows. In Section II, we define the VS problem at hand and corresponding error spaces related to the mobile manipulation problem. In Section III III, we introduce the sensor system (encoders, IMUs, and vision) required to address this PBVS problem and we provide a fault tolerant solution to pallet pose estimation. In Section IV, we describe the feedback control laws that drive the errors to zero and solve this PBVS problem. Finally, we examine the overall performance of the proposed control system by experimental evaluation on a forklift truck. In Section V we present the results and demonstrate the capability of the proposed PBVS controller in the presence of failures in marker detection, wheel slippage, and kinematic constraints. The case study machine (Figure 1) is a small prototype wheel loader based on Avant 635 that was built at the Department of Intelligent Hydraulics and Automation (IHA), Tampere University of Technology (TUT), under the GIM project [18]. This machine is hereafter referred to as the GIM mobile machine. The body of the machine comprises two units, and steering is performed by controlling the angle between the units; this is referred to as articulated frame steering (AFS). Figure 2 shows a simplified model of the same. The manipulator manipulator has three degrees of freedom-boom, fork, and telescopic boom-that can be controlled independently [19].

II. Problem Definition

Automatic pallet picking is one of the functionalities of fully autonomous heavy vehicles. It can be described as follows. The approximate locations of the pallets are known in a global map, and the machine can navigate such that it is near the target pallet using standard global localization systems such as *Global Positioning System* (GPS), inertial sensors, and LIDAR-based *Simultaneous Localization and Mapping* (SLAM) [20]. However, most of these solutions are not accurate enough for pallet handling tasks. When close enough, with the aid of vision, the target pallet pose is estimated and pallet picking is converted to a PBVS problem. We propose a methodology to achieve robustness against vision failures and describe a control architecture that realizes smooth and non-stop transition from navigation in the global frame to VS and leads the forks into the target pallet. Next, we introduce the coordinate frames necessary to mathematically formulate the problem.Let $\{\mathcal{I}\}$ denote the global frame. The body fixed coordinate frame $\{\mathcal{B}\}$ is attached to the midpoint of the front axle as shown in Figure 1. The pose of the vehicle is defined by a homogeneous transformation matrix ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}$:

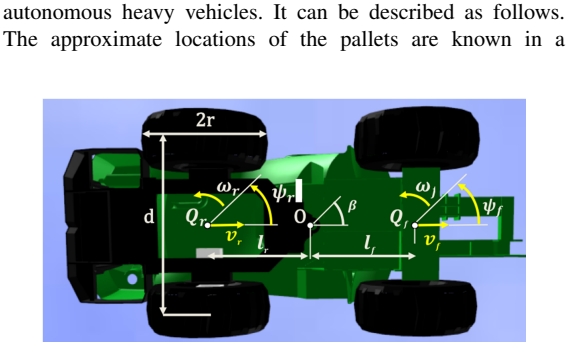


Fig. 2. GIM mobile machine with two body units pivoting around O

$$ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(x,y,z,\phi,\theta,\psi) = \left[\begin{array}{c|c} {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}(\phi,\theta,\psi) & {}^{\mathcal{I}}\boldsymbol{P}_{\mathcal{B}} \\ \hline \mathbf{0} & 1 \end{array}\right]_{4\times 4} \tag{1} $$

in which the rotation matrix ${}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}$ defines the orientation of the front unit as a function of ϕ, θ, and ψ (roll, pitch, and yaw, respectively). Moreover, ${}^{\mathcal{I}}\boldsymbol{P}_{\mathcal{B}} = (x, y, z)$ is the position of the body origin in the inertial frame. Let $\{\mathcal{P}\}$ and $\{\mathcal{F}\}$ denote the pallet frame and the fork frame, respectively. They are defined in such a manner that the control objective is achieved when ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{P}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{F}}$, in other words, when $\{\mathcal{P}\}$ and $\{\mathcal{F}\}$ coincide. This should be clear from Figure 1. The manipulator has three degrees of freedom. The boom can rotate around the y axis of the body and stretch in/out. Thus the generalized coordinates of the boom are specified in terms of the angle θ_1 and length L. The fork can only rotate around the y axis of the body by an angle of θ_2. Thus, the fork in the body can be defined by

$$ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}} = \boldsymbol{T}(L, \theta_1, \theta_2) \tag{2} $$

Manipulation is realized by controlling the boom angle, telescopic boom, and fork angle by u_1, u_L , and u_2, respectively. Moreover, let u_s and u_v denote the steering and speed commands, respectively. The control objective is to devise control laws for these five control signals such that

$$ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{F}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(x,y,z,\phi,\theta,\psi)\ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}(L,\theta_1,\theta_2) \tag{3} $$

coincides with ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{P}}$. It should be noted that the roll motion ϕ (rotation around x-axis of the pallet) is not controllable. Therefore, we assume that the pallet has the same roll angle as the ground, which then determines the roll angle of the machine. To control the five remaining degrees of freedom (three positions and two angles), we now divide the error space into two orthogonal planes:

1) $x-y$ plane error variables (x, y, ψ) are controlled by drive (u_s, u_v)
2) $x-z$ plane error variables (x, z, θ) are controlled by the manipulator $(u_1,\ u_L,\ u_2)$

Clearly, there is a redundancy in the control space in that x can be controlled by both the drive and the manipulator. The accuracy of position control using the machine speed is limited. Therefore, u_v is used to drive the fork near the pallet, and the manipulator is used if necessary to reduce the remaining residual error to zero (usual less than 10cm). A state-machine resolves this and other logical control issues. Section IVdescribes the motion control solution, in which a path-following strategy is introduced to address control in the $x-y$ plane and feedback laws are formulated for the manipulator to control the errors in the $x-z$ plane. In Section III-B, we will show how dead-reckoning (wheel odometry and IMU) can be used to improve visual feedback and how the proposed VS can tolerate outdoor conditions where marker detection can fail frequently and wheel slippage is unavoidable especially in the snowy weather of Finland.

III. Sensor Data and State Estimation

Figure 3 shows the mechatronic architecture of the GIM machine, where the sensors, actuators, and computing units are shown. Next, we describe the sensor information used in the proposed control system.

A. Sensor Information

Visual. In order to observe the $\{\mathcal{P}\}$ frame, a set of markers have been installed on the face of the pallet as shown on Figure 1. By using a camera, the onboard pose of the markers, and thus that of the pallet, that is, ${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{P}}$, with respect to the camera frame $\{\mathcal{C}\}$ is determined. Therefore, the camera in the body can be resolved if the calibration parameter ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}$ is known:

$$ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{P}} = {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}\ {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{P}} \tag{4} $$

which defines the visual feedback. For image processing, we have used the Alvar toolbox [12], an open-source augmented reality library. It can be used to track fiducial markers based on the known shape and size. It outputs the position and the orientation of the marker relative to the camera. This toolbox is capable of automatically tuning distortion and intrinsic parameters for a pin-hole camera by detecting a predefined checkerboard marker from fifty different poses. A comparison of the calibration results shows that the calibration of intrinsic parameters under indoor illumination conditions (before installation on the vehicle) provides better results. Other necessary measurements are performed using the wheel encoders and inertial sensors.

Wheel encoders. Each wheel of the GIM mobile machine is retrofitted with two hall sensors to construct an encoder with a seventeen-teeth steel plate (for practical reasons). Owing to the low resolution, pulse widths are measured. The speed measurement is performed at $20Hz$, and it is not valid for speeds under $0.25m/s$. We never drive under this critical speed.

Intrinsic Sensors. The body articulation angle is measured at the rate of $100Hz$ using a resolver. An IMU is installed near the midpoint of the front axle. It includes three-axes gyros and three-axes accelerometers providing data at 500 Hz. There are two other IMUs, one of which is installed on the boom and the other, on the fork. Each includes one gyro and two accelerometers assembled on three orthogonal axes. These three IMUs are used to measure the boom and fork angles with respect to the body. This mechanism is described in Section III-B. The length of the telescope or boom extension is measured using a potentiometer.

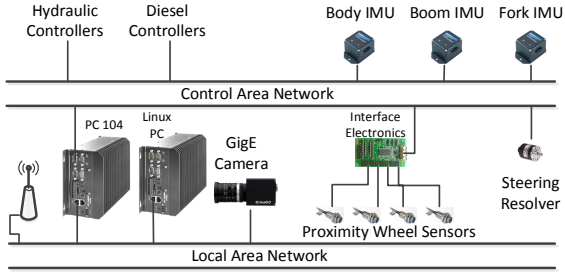


Fig. 3. Part of control hardware architecture for GIM Machine

B. State Estimation

Body Inclination. The orientation of the body frame is calculated in the quaternion representation using the IMU installed on the front axle. The gyro and accelerometer signals are filtered using a nonlinear complementary filter on the special orthogonal group $SO(3)$ proposed in [21]. The filter eliminates random noises and biases; however, it suffers from gyro scaling factors. Additionally, when the machine is stopped, a window averaging is used to eliminate the gyro biases. Steady-state errors are less than 0.5° which is tolerable in our application. The fork and boom angles in the inertial frame are also measured using IMUs by applying the same complementary filters about one axis. The fork and boom angles in the body frame are then calculated by subtracting the body pitch angle. More details on the use of inertial sensors in mobile machine applications are found in [22]. In summary, the outputs of the complementary filters are roll ϕ, pitch θ, boom angle θ_1, and fork angle θ_2.

Dead-reckoning When the GIM machine moves on uneven terrain, it is subject to motions in all directions. However, when aggressive maneuvers are avoided during pallet picking, as in our case, neglecting the y and z components of the velocity provides a good approximation. By this assumption, we use the following kinematic equations for dead-reckoning [23]:

$$ \begin{aligned} \dot{x} &= v_x \cos\theta \cos\psi \\ \dot{y} &= v_x \cos\theta \sin\psi \\ \dot{\psi} &= \omega_z \frac{\cos\phi}{\cos\theta} + \omega_y \frac{\sin\phi}{\cos\theta} \end{aligned} \tag{5} $$

Dead-reckoning involves integrating (5) to obtain at $x(t), y(t)$,and $\psi(t)$. Toward this end, we require the speed v_x, and angular velocities ω_z and ω_y. Wheel odometry is simple, inexpensive, and easy to accomplish in real-time. Because the machine has four wheel drive (there are no free-wheeling tires), and hydraulics flows are not controlled independently,

the wheels are prone to slip and skid. In [24], it is shown how kinematics constraints on four wheel speeds and body articulation angle can be used to detect slippage and improve v_x and ω_z estimates. In the present study, by following a similar idea, we have further used z-axis gyro measurements to improve these estimates. The details of the same are omitted. Variables ω_y, ϕ, and θ used in (5) are the outputs of the complementary filter explained above.

Fault Tolerant Sensor Fusion and Synchronization The dead-reckoning data is updated at $20Hz$ or every 50ms in real-time. Experiments show that the time intervals elapsed between two successful consecutive marker detections are mostly around $180-200ms$, and they can increase to 1.5sec. When the markers are visible, the success of detection depends on the angle and distance to the markers and the camera motion. Unfavorable conditions include a large distance to the markers and fast steering/yaw motions. Fast yaw motions are unavoidable when the machine approaches the pallet with large initial errors in the y-axis or yaw angle. Owing to the size of the machine and steering mechanism and the fact that markers are reliably detected only when the fork is $3-4m$ away from the pallet, tight maneuvers are often necessary. In this section, we propose a state estimation filter that addresses the problem of marker loss situations, for example, in the case of fast yaw motions or when the pallet is in "shadow". Shadow situation occurs at the last stage of operation when the boom and fork block the view to pallet when the fork is around $0.3m$ from the pallet. The Alvar library also provides the process time, which indicates how long ago the image of the corresponding detection results has been acquired. These process time values are 105-140 ms. The shorter the distance, the shorter is the process time. The process time, denoted by T_d, is used to synchronize the vision data with dead-reckoning. This will be explained below. Using wheel odometry and the IMU, dead-reckoning and complementary filters provide

$$\begin{aligned} {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}(t) &= \boldsymbol{R}_z(\psi)\,\boldsymbol{R}_y(\theta)\,\boldsymbol{R}_x(\phi) \\ {}^{\mathcal{I}}\boldsymbol{P}_{\mathcal{B}}(t) &= \begin{bmatrix} x & y & 0 \end{bmatrix}^T \end{aligned} \tag{6}$$

with respect to some inertial frame $\{\mathcal{I}\}$ which is the point at which dead-reckoning integrators are reset. This can be performed at any time before VS has started.

The output of the vision at time t is ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}(t-T_d)$ as calculated from (4). To synchronize the dead-reckoning and vision, we store a series of $[x, y, \phi, \theta, \psi](k) : k = t, t-T_s, t-2T_s, \ldots$, where $T_s = 50ms$. It should be noted that because the pallet is not moving, ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$ is constant. Every time the object is detected, we calculate

$${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}} = {}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}(t-T_d)\ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{I}}(t-T_d) \tag{7}$$

If ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{I}}$ at time $t-T_d$ is not available in the stored set, we make a simple linear interpolation to calculate ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{I}}(t-T_d)$, that is, $[x, y, \phi, \theta, \psi](t-T_d)$ is calculated by linear interpolation between two stored dead-reckoning data of the two closest time instances. Then, ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{I}}(t-T_d)$ is calculated by (6). However, ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$ calculated from (7) is noisy. Averaging over several instances has been shown to improve and considerably smoothen the estimate. Let ${}^{\mathcal{P}}\hat{\boldsymbol{T}}_{\mathcal{I}}$ denote the average of n latest measurement of ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$ using (7). If the markers are found at least n times, the pallet tracking filter is initiated and VS is started. To recover ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}(t)$ between consecutive instances of marker detection, we can calculate

$${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}(t) = {}^{\mathcal{P}}\hat{\boldsymbol{T}}_{\mathcal{I}}\ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(t) \tag{8}$$

as fast as dead-reckoning is updated. This will provide a real time fault tolerant feedback to the control system.

The outliers are eliminated in two stages. First, by using knowledge of the approximate location of the target pallet, we eliminate those ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}$ measurements that are out of bound. Second, by using knowledge of ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$ being constant, before averaging n measurements, we eliminate those that are too far from the rest (using k-means clustering). Common faults such as marker detection failure, faults, and error accumulation of localization are partially canceled using the above-described method. The length of the averaging window is obtained via a trade-off. The longer the averaging sequence, the smoother is the estimate. However, wide averaging window is not optimal, because as the machine approaches the pallet, the vision data become more accurate. Moreover, if we use an overly long averaging window, the machine needs to go very slow to obtain reliable feedback data sufficiently early and to have room to maneuver before getting too close to the pallet. Experience shows that averaging over $n = 5$ latest measurements provides a proper trade-off.

It should be noted that averaging over several rotation matrices will not lead to a rotation matrix. One approach is to keep the principal components of the resulting average. Let us denote $\boldsymbol{R}_n = \frac{1}{n}\Sigma_{k=1:n}\ {}^{\mathcal{P}}\boldsymbol{R}_{\mathcal{I}}(k)$ and its singular value decomposition by $\boldsymbol{R}_n = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$. Then ${}^{\mathcal{P}}\hat{\boldsymbol{R}}_{\mathcal{I}} = \boldsymbol{U}\boldsymbol{V}^T$ is the closest rotation matrix to $\boldsymbol{R}_n$. There are two issues worth mentioning. By averaging over 5 measurements, we have assumed that the added error by dead-reckoning in five consecutive measurements is negligible compared to the vision errors. We have also assumed that the statistical properties of vision errors remain the same. None of these assumptions are correct, and this issue remains to be explored in future work.

IV. Motion Control

Recall that the control objective is to drive the fork frame into the pallet frame; thus, the error can be defined by translation ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{F}} = {}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}\ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}$. We have already described how measurements ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}$ and ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}$ are generated at the rate of $20Hz$. See (2) and (8). As justified before, we divide the control space into two groups, and the errors are defined in two orthogonal spaces. In other words, the manipulators solves the pallet picking problem in the *x-z plane* and the drive-steering solves the pallet picking problem in the *x-y plane*.

A. Manipulator Control

Toward this effect, we define the errors in the *x-z plane*

$$\begin{aligned} e_\phi &= \arg({}^{\mathcal{P}}\boldsymbol{R}_{\mathcal{F}}, \boldsymbol{y}) \\ e_z &= \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}{}^{\mathcal{P}}\boldsymbol{P}_{\mathcal{F}} \\ e_x &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}{}^{\mathcal{P}}\boldsymbol{P}_{\mathcal{F}} \end{aligned} \tag{9}$$

where the $\arg(.,.)$ function returns the angle argument of the rotation matrix around the axis defined in the second argument (y-axis in this case).

The manipulator actuators are driven by the pressure-compensated proportional valves and it is reasonable to assume that

$$\dot{\theta}_1 = u_1, \dot{\theta}_2 = u_2, \dot{L} = u_L, \tag{10}$$

where u_1, u_2, u_L are the valve commands. Employing the Lyapunov function $V = 0.5k_1e_x^2 + 0.5k_2e_z^2 + 0.5e_\phi^2$, it can be shown that the Jacobean-like control laws are as follows:

$$\begin{aligned}
\dot{\theta}_1 &= u_1 = -k_{p1}\text{sat}\left(-k_1e_xL\sin\theta_1 - k_2e_zL\cos\theta_1\right)\\
\dot{L} &= u_L = -k_{p3}\text{sat}\left(k_1e_x\cos\theta_1 - k_2e_z\sin\theta_1\right)\\
\dot{\theta}_2 &= u_2 = -k_{p2}\text{sat}\left(e_\phi\right)
\end{aligned} \tag{11}$$

The drive the errors $e_\phi,\ e_x$, and e_z to zero. In the equations above,

$$\text{sat(a)} = \begin{cases} 1 & \text{if } a > 1 \\ -1 & \text{if } a < -1 \\ a & \text{otherwise} \end{cases} \tag{12}$$

is a saturation function, and gains $k_{pi},\ i = 1, 2, 3$ define the velocity bounds of the actuators and unit conversions.

B. Steering-Drive control

Owing to the non-holonomic constraints of the machine, the control in the x-y plane takes a rather more complex structure. When the pallet tracking filter is initialized, the pallet pose estimates are relatively accurate. A smooth path $\boldsymbol{p}_d$ is then generated that extends from the current location $[x_{init}\ \ y_{init}]$ of the machine to the pallet frame, that is, the origin $[0\ \ 0]$. Recall that the errors and position $[x_{init}\ \ y_{init}]$ are defined in the pallet frame $\{\mathcal{P}\}$. In our experiments, we have used a Bezier curve of order 4, with the following control points:

$$\begin{aligned}
P_1 &= [x_{init}\ \ y_{init}],\\
P_2 &= [x_{init} + 1.4\ \ 0],\\
P_3 &= [x_{init} + 0.7\ \ 0],\\
P_4 &= [0\ \ 0]
\end{aligned} \tag{13}$$

This choice guarantees that path $\boldsymbol{p}_d$ is tangential to the x-axis of $\{\mathcal{P}\}$ at its origin. Now, we define the error $({}^{\mathcal{T}}\boldsymbol{R}_{\mathcal{F}}, {}^{\mathcal{T}}\boldsymbol{P}_{\mathcal{F}})$, where $\{\mathcal{T}\}$ is a tangent frame to the path attached to $\boldsymbol{p}_d(\tau)$, where $\tau \in [0, 1]$ is the path parameter. It should be noted that $\{\mathcal{T}\}$ coincides with $\{\mathcal{P}\}$ for $\tau = 1$. The path following errors can now be defined,

$$\begin{aligned}
e_\psi &= \arg({}^{\mathcal{T}}\boldsymbol{R}_{\mathcal{B}}, \boldsymbol{z}) = \arg({}^{\mathcal{T}}\boldsymbol{R}_{\mathcal{F}}, \boldsymbol{z})\\
e_x &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}{}^{\mathcal{T}}\boldsymbol{P}_{\mathcal{B}}\\
e_y &= \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}{}^{\mathcal{T}}\boldsymbol{P}_{\mathcal{B}} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}{}^{\mathcal{T}}\boldsymbol{P}_{\mathcal{F}}
\end{aligned} \tag{14}$$

Notice that errors in the yaw and y-axis are the same in the body and in the fork frame. Therefore, if the machine follows the path closely and the pose estimates are accurate enough, the errors in the yaw and y-axis will be sufficiently small when the fork frame reaches the pallet. The error in the x-axis is not relevant and as discussed before, the system is redundant in this coordinate. It should also be noted that the closer the machine to the pallet, the more accurate are the pose estimates. Therefore, even if the trajectory of the body frame in the real world is initially not the same as $\boldsymbol{p}_d$, it converges to it as the pose estimation errors decrease.

Definition: Path following problem. Design a control law for steering signal u_s, to drive e_ψ and e_y to zero.

A solution to this problem is devised in [19], which can be summarized as follows. We showed in [24] that the motion of the front unit of the GIM machine closely follows the kinematics

$$\begin{aligned}
\dot{x} &= v_x\cos\psi\\
\dot{y} &= v_x\sin\psi\\
\dot{\psi} &= \omega_z
\end{aligned} \tag{15}$$

and

$$\omega_z = \frac{l_r\dot{\beta} + v_x\sin\beta}{l_f\cos\beta + l_r}, \tag{16}$$

where β is the body articulation angle and l_f, l_r are geometrical parameters (see Figure 2). Notice that (15) is a simplified version of (5) in the horizontal plane, but it is accurate enough for feedback control. The solution in [24] provides a control law for the desired angular velocity $\omega_z^c = C_{PF}(e_x, e_y, e_\psi))$, where the speed v_x remains free and can be controlled separately. Freedom in assigning the linear speed is instrumental in our case, because without causing instability in the path-following controller, the machine can be slowed down or stopped. For example, in case the manipulator errors are not yet small enough, collision with the pallet can possibly be avoided. Now, given ω_z^c, we use (16) to calculate the required steering valve command:

$$\begin{aligned}
\dot{\beta}^c &= -\frac{v_f}{l_r}\sin\beta + \left(\frac{l_f}{l_r}\cos\beta + 1\right)\omega_f^c\\
u_s &= K\dot{\beta}^c
\end{aligned} \tag{17}$$

where we assumed that the valve command is proportional to the speed of body articulation.

V. Experimental Results

It is reasonable to assume that, in practice, the position of the pallet is known with certain accuracy. In the case of the GIM machine, the pallet locations are registered with an accuracy of 0.5 m using on-board GPS and the inertial navigation system after the last placement. This accuracy is

TABLE I
SUMMARIZED SPECIFICATIONS OF GIM MOBILE MACHINE COMPONENTS

Device	Specification	Value
Wheel	Radius	$r = 0.32m$
Vehicle	Max. Speed	$3.3^{m/s}$
	l_f	$0.6m$
	l_r	$0.6m$
IMU Gyro	Angular random walk	$N_{IMU} = 10^{-3}\ \frac{\text{rad}/s}{\sqrt{Hz}}$
	Bias instability	$B_{IMU} = 10^{-6}$ rad$/s$
	Time correlation	$T_{IMU} = 750\ s$
Wheel Encoder	Resolution	17 pulses/round

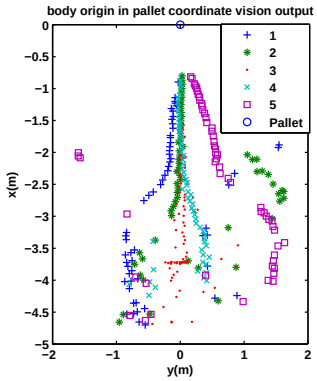


Fig. 4. Marker detection output, raw data. Five representative tests

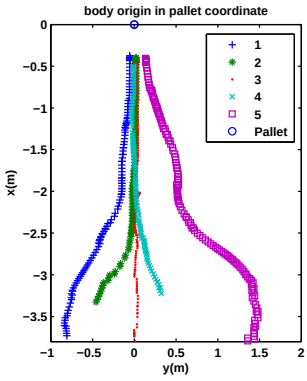


Fig. 6. Tracking filter output. Five representative tests

enough to send the machine to the proximity of the pallet, and therefore, the face of the pallet is visible to the vision system. Based on this observation, we have set our experiments as follows. In all the test runs, the machine started from different initial configurations at distances of 7-8 m in front of the pallet and facing toward the pallet with different initial headings. Initially, the lateral distances of the body frame with respect to the center of the pallet were in the range of 1 m. The machine then moves in a straight line until the pallet is detected and the pallet tracking filter is initialized. After the filter is initialized, PBVS is activated. The on-board camera was a Gima GO423C with 1296×966 resolution. Recall that owing to the reduced number of pulses per revolution of wheel odometry devices, speeds under $0.25m/s$ are not reliably measured. Therefore, in the experiments reported in this paper, we have set the speed commands to a constant value of $0.3m/s$ or zero (stop). The actual speed varied from 0.3–$0.4m/s$ in the test runs. We conducted a total of 13 experiments. The experiments were conducted on a cloudy day between 14:00 and 15:00 in December 2012. A few of the final experiments were performed with the machine's front lights on[1]. All the experiments were successful. The tracking filter was initialized at the following distances (when the markers are detected the first five times): $[3.17, 3.63, 6.97, 3.66, 3.52, 4.09, 4.21, 4.09, 4.54, 3.35, 3.54, 4.30, 4.02$ m. The fork blocks the camera view or the marker is lost at the following distances: $[0.31, 0.21, 0.36, 0.24, 0.36, 0.17, 0.29, 0.27, 0.25, 0.30, 0.21, 0.42, 0.6($ m.

Figure 4 shows the trajectory of the body frame in the pallet frame in the x-y pallet plane. For clarity of presentation, we show only five representative tests: two each starting from the left and the right side of the pallet and one from the middle. This figure shows the marker detection output before filtering. It can easily be seen that detection is very unreliable at large distances and that it improves as the machine approaches the pallet. Figure 6 shows the same trajectories after filtering. Notice that how the trajectories have been smoothened out. Figures 5, 7, and 8 show the pitch, yaw, and z-axis errors, respectively, versus time for experiments 1-4 in four subplots (the numbers correspond to the numbers in Figure 4). Some observations can be made from these figures. The tracking filter starts long after the vision output becomes available, and it continues producing pose estimates even after the vision output is not available. It is visually clear how the tracking

[1]In this latitude, sun sets around 15:00 in December.

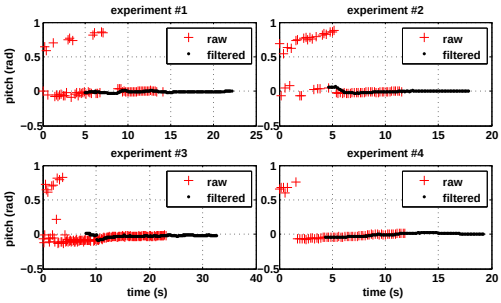


Fig. 5. Pitch angle errors, before and after filter. Four representative tests

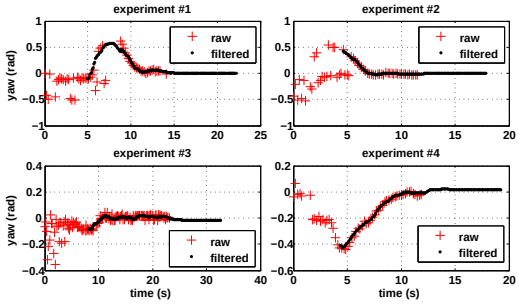


Fig. 7. Yaw angle errors, before and after filter. Four representative tests

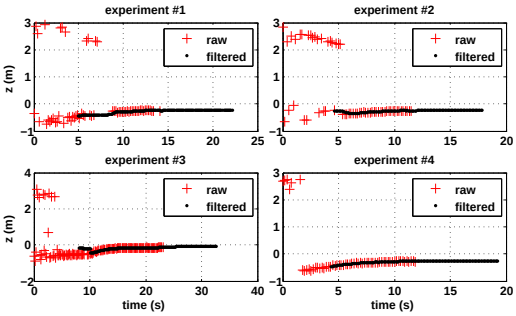


Fig. 8. Height errors, before and after filter. Four representative tests

filter has smoothened the vision output.

VI. Conclusion

In this paper, we introduced a position-based VS control framework and a fault tolerant pose estimation algorithm. This method was developed and experimented on a small-sized forklift truck. The choice of the coordinate frames and references provides a natural framework to divide the error space into two orthogonal planes and to fuse the vision output with the dead-reckoning data. One plane was then controlled by drive-steering and the other, by the manipulators. The redundancy resolution (along the x-axis) was achieved by the state-machine (the details are omitted). Real-world experiments showed the robustness and efficacy of the methodology for the problem in hand. Future work includes the optimal fusion of the image processing data with dead-reckoning instead of averaging. A truly reliable integration of vision into VS, especially in outdoor conditions, requires the extension of the logic control and planning algorithm for better exception handling. In real-world scenarios, there are always broken pallets that make pallet detection a real challenge. Alternative detection methods without the use of markers, for example, using active sensors, will also improve the reliability of the system. Further development of this work can be a visual servoing platform to utilize a path follower controller addressed in [25] for complex mobile robot architectures.

VII. Acknowledgment

This research is supported in part by the Academy of Finland under the GIM project. The first author has benefited from a PhD grant by the European Union's Seventh Framework Program under the Marie Curie Initial Training Network within the framework of the PURESAFE project.

References

[1] W. Xie, Z. Li, X. Tu, and C. Perron, "Switching control of image-based visual servoing with laser pointer in robotic manufacturing systems," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 520–529, 2009.

[2] L. Baglivo, N. Biasi, F. Biral, N. Bellomo, E. Bertolazzi, M. Da Lio, and M. De Cecco, "Autonomous pallet localization and picking for industrial forklifts: a robust range and look method," *Measurement Science and Technology*, vol. 22, no. 8, p. 085502, 2011.

[3] Z. Miljković, N. Vuković, M. Mitić, and B. Babić, "New hybrid vision-based control approach for automated guided vehicles," *The International Journal of Advanced Manufacturing Technology*, pp. 1–19, 2012.

[4] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, 2010.

[5] V. Lippiello, F. Ruggiero, B. Siciliano, and L. Villani, "Visual grasp planning for unknown objects using a multifingered robotic hand," *IEEE/ASME Transactions on Mechatronics*, vol. PP, no. 99, pp. 1 –10, 2012.

[6] F. Janabi-Sharifi and M. Marey, "A kalman-filter-based method for pose estimation in visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 939–947, 2010.

[7] F. Chaumette and S. Hutchinson, "Visual servo control. II. advanced approaches [tutorial]," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[8] G. Chesi and A. Vicino, "Visual servoing for large camera displacements," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 724–735, 2004.

[9] E. Malis, F. Chaumette, and S. Boudet, "$2\frac{1}{2}$D visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

[10] F. Chaumette and S. Hutchinson, "Visual servo control. I. basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[11] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[12] VTT Technical Research Centre of Finland, "Alvar toolkit website." http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/index.html, 2013.

[13] A. Ferreira, C. Cassier, and S. Hirai, "Automatic microassembly system assisted by vision servoing and virtual reality," *IEEE/ASME Transactions on Mechatronics*, vol. 9, no. 2, pp. 321–333, 2004.

[14] H. Taghirad, S. Atashzar, and M. Shahbazi, "Robust solution to three-dimensional pose estimation using composite extended Kalman observer and Kalman filter," *Computer Vision, IET*, vol. 6, no. 2, pp. 140–152, 2012.

[15] Y. Wang, H. Lang, and C. de Silva, "A hybrid visual servo controller for robust grasping by wheeled mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 757–769, 2010.

[16] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Transactions On Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.

[17] G. Lopes and D. Koditschek, "Visual servoing for nonholonomically constrained three degree of freedom kinematic systems," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 715–736, 2007.

[18] Finnish Centre of Excellence in Generic Intelligent Machines (GIM) Research. http://gim.aalto.fi, 2013.

[19] R. Ghabcheloo, M. Hyvönen, J. Uusisalo, O. Karhu, J. Järä, and K. Huhtala, "Autonomous motion control of a wheel loader," in *ASME Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control*, Hollywood, CA, US, 2009.

[20] A. Aghamohammadi, A. Tamjidi, and H. Taghirad, "SLAM using single laser range finder," *Proceedings of 17th World Congress of the International Federation of Automatic Control (IFAC)*, 2008.

[21] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *Automatic Control, IEEE Transactions on*, vol. 53, no. 5, pp. 1203–1218, 2008.

[22] J. Honkakorpi, J. Vihonen, and J. Mattila, "Sensor module for hydraulic boom state feedback control," *International Journal of Fluid Power*, vol. 13, no. 3, pp. 15–25, 2012.

[23] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 141–166, 2007.

[24] R. Ghabcheloo and M. Hyvönen, "Modeling and motion control of an articulated-frame-steering hydraulic mobile machine," *Proceedings of the 17th Mediterranean Conference on Control and Automation*, Greece, June 2009.

[25] R. Oftadeh, R. Ghabcheloo, and J. Mattila, "A novel time optimal path following controller with bounded velocities for mobile robots with independently steerable wheels," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Japan,2013.

Publication II

M. Aref, M., Ghabcheloo, R., and Mattila, J., "A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong*, pp. 6816–6822, 2014

A Macro-Micro Controller for Pallet Picking by an Articulated-frame-steering Hydraulic Mobile Machine

Mohammad M. Aref, Reza Ghabcheloo, and Jouni Mattila

***Abstract*— This paper addresses the macro-micro configuration of a mobile manipulation problem for a forklift; specifically, it investigates pallet picking with visual feedback. A manipulator with limited degrees of freedom and differential constraint mobility, together with the large dimensions of the machine, requires reliable visual feedback (pallet pose) and navigation from relatively large distances. It has been shown that the problem can be divided into two parts in order to solve the related issues based on path following theories and visual servoing. Moreover, visual pallet detection is non-real-time and unreliable, especially due to large distances, unfavorable vibrations, and fast steering. To address these issues, we introduce a simple and efficient method that integrates the vision output with odometry and realizes a smooth and non-stop transition from global navigation to visual servoing. Real-world implementation on a small-sized forklift demonstrates the efficacy of the proposed macro–micro architecture.**

I. Introduction

Visual Servoing (VS), or vision-based robot control, is found in a wide range of applications, such as in humanoid robots and robotic manufacturing. VS is usually used in object detection, for manipulation or target tracking. In this study, we address the problem of VS for a mobile manipulation task, namely, pallet picking using a forklift. Because bulky and massive materials are commonly handled on a pallet, automatic pallet picking with a forklift has attracted considerable research attention. The main challenge in this task is the reliable detection of the pallet while eliminating typical failures associated with vision systems. To increase the robustness of the vision, previous studies have often used auxiliary sensors such as laser pointers [1] or laser scanners [2]; another study [3] applied a neural extended Kalman filter for object detection. In this study, we propose a fault-tolerant pallet pose (position and orientation) estimation in which vision data is fused with odometry and inertial measurements. The estimation output is then used for VS and motion control.

VS can be classified into three categories: *Image-Based Visual Servoing* (IBVS), *Position-Based Visual Servoing* (PBVS), and a hybrid of both. A more detailed study on the visual servoing backgrounds for pallet picking was studied in [4].

The present study focuses on mobile manipulation and the fusion of vision pose estimates with inertial sensors. We have used an on-board monocular camera in combination with fiducial markers on the pallet. The pose of the markers is computed using the Alvar computer vision library [5] in a non-real-time fashion. The integration of inertial sensors with vision provides real-time pallet pose estimation, even in the event of extended delays in visual pallet detection or when the view to the markers is blocked by the boom and fork. The proposed method is general and should be independent of the computer vision method used to detect the pallet pose.

Authors are with Intelligent Hydraulics and Automation Department, Tampere University of Technology, 33101, Finland.
Email: m.aref@ieee.org .

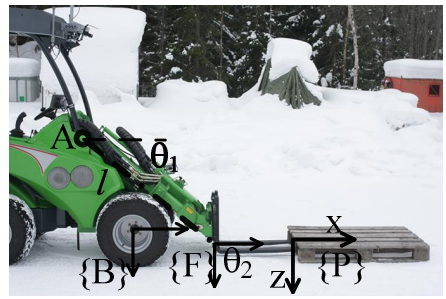


Fig. 1. The micro robot's coordinate frames.

In this study, we propose a macro-micro control architecture in which the pallet frame plays the role of the reference frame or the origin, with respect to which all the errors are defined. After the pallet is robustly detected for the first time (as explained later), the control system switches to the pallet frame as a local frame. Therefore, the target pose remains the same (i.e., the origin) throughout VS. Defining the pallet as the origin has the advantage that path planning is performed only once, at the start, when the pallet has been detected. Unlike in previous studies [6], [7], the macro part's controller keeps the vehicle's path follower in the control loop. Then, when the robot approaches the pallet, micro part does the visual servoing.

The remainder of this paper is organized as follows. In Section II, we define the VS problem at hand and corresponding error spaces related to the mobile manipulation problem. In Section III, we describe the feedback control laws that drive the errors to zero and solve this PBVS problem. Finally, we examine the overall performance of the proposed control system by experimental evaluation on a forklift. In Section IV, we present the results and demonstrate the capability of the proposed macro-micro controller in the presence of failures in marker detection, wheel slippage, and kinematic constraints.

The case study machine (Figure 1) is a small prototype

wheel loader based on Avant 635 that was built at the Department of Intelligent Hydraulics and Automation (IHA), Tampere University of Technology (TUT), under the GIM project [8]. This machine is hereafter referred to as the GIM mobile machine. The body of the machine comprises two units, and steering is performed by controlling the angle between the units; this is referred to as articulated frame steering (AFS). Figure 2 shows a simplified model of the same. The manipulator manipulator has three degrees of freedom (boom, fork, and telescopic boom)that can be controlled independently [9].

II. Problem Definition

Automatic pallet picking is one of the functionalities of fully autonomous heavy vehicles. It can be described as follows. The approximate locations of the pallets are known in a global map, and the machine can navigate such that it is near the target pallet using standard global localization systems such as the *Global Positioning System* (GPS), inertial sensors, and LIDAR-based *Simultaneous Localization and Mapping* (SLAM) [10]. However, most of these solutions are not accurate enough for pallet handling tasks. When close enough, with the aid of vision, the target pallet pose is estimated and pallet picking is converted to a PBVS problem. We propose a methodology to achieve robustness against vision failures, and describe a control architecture that realizes smooth and non-stop transition from navigation in the global frame to VS, and leads the forks into the target pallet. Next, we introduce the coordinate frames necessary to mathematically formulate the problem.

Let $\{\mathcal{I}\}$ denote the global frame. The body fixed coordinate frame $\{\mathcal{B}\}$ is attached to the midpoint of the front axle as shown in Figure 1. The pose of the vehicle is defined by a homogeneous transformation matrix ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}$:

$$ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(x,y,z,\phi,\theta,\psi) = \left[\begin{array}{c|c} {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}(\phi,\theta,\psi) & {}^{\mathcal{I}}\boldsymbol{P}_{\mathcal{B}} \\ \hline \mathbf{0} & 1 \end{array}\right]_{4\times 4} \quad (1) $$

in which the rotation matrix ${}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}$ defines the orientation of the front unit as a function of ϕ, θ, and ψ (roll, pitch, and yaw, respectively). Moreover, ${}^{\mathcal{I}}\boldsymbol{P}_{\mathcal{B}} = (x, y, z)$ is the position of the body origin in the inertial frame. Let $\{\mathcal{P}\}$ and $\{\mathcal{F}\}$ denote the pallet frame and the fork frame, respectively. They are defined in such a manner that the control objective is achieved when ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{P}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{F}}$, in other words, when $\{\mathcal{P}\}$ and $\{\mathcal{F}\}$ coincide. This should be clear from Fig.1. The manipulator has three degrees of freedom. The boom can rotate around the y axis of the body and stretch in/out. Thus the generalized coordinates of the boom are specified in terms of the angle θ_1 and length L. The fork can only rotate around the y axis of the body by an angle of θ_2. Thus, the fork in the body can be defined by:

$$ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}} = \boldsymbol{T}(L,\theta_1,\theta_2) \quad (2) $$

Manipulation is realized by controlling the boom angle, telescopic boom, and fork angle by u_1, u_L , and u_2, respectively. Moreover, let u_s and u_v denote the steering and speed commands, respectively. The control objective is to devise control laws for these five control signals such that:

$$ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{F}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(x,y,z,\phi,\theta,\psi)\ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}(L,\theta_1,\theta_2) \quad (3) $$

coincides with ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{P}}$. It should be noted that the roll motion ϕ (rotation around the x-axis of the pallet) is not controllable. Therefore, we assume that the pallet has the same roll angle as the ground, which then determines the roll angle of the machine. To control the five remaining degrees of freedom (three positions and two angles), we now divide the error space into two orthogonal planes:

1) $x-y$ plane error variables (x, y, ψ) are controlled by drive (u_s,u_v).
2) $x-z$ plane error variables (x, z, θ) are controlled by the manipulator (u_1, u_L, u_2).

Clearly, there is a redundancy in the control space in that x can be controlled by both the drive and the manipulator. The accuracy of position control using the machine speed is limited. Therefore, u_v is used to drive the fork near the pallet, and the manipulator is used if necessary to reduce the remaining residual error to zero (usually less than 10cm). A state-machine resolves this and other logical control issues. Section III describes the motion control solution, in which a path-following strategy is introduced to address control in the $x-y$ plane, and feedback laws are formulated for the manipulator to control the errors in the $x-z$ plane. In [4] we show how dead-reckoning (wheel odometry and IMU) can be used to improve visual feedback, and how the proposed VS can tolerate outdoor conditions where marker detection can fail frequently and wheel slippage is unavoidable, especially in the snowy weather of Finland.

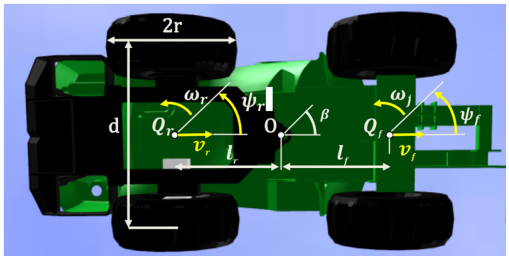


Fig. 2. GIM mobile machine with two body units pivoting around O.

III. Visual Servoing Controller

A. Visual Servoing Preliminaries

Consider that $\boldsymbol{S}_i (i = 1,\ 2,\ 3, \ldots, k)$ are the feature points defining the $\{\mathcal{M}\}$ and $\{\mathcal{M}^*\}$ three-dimensional coordinate systems for the current and desired object poses ${}^{\{\mathcal{C}\}}\boldsymbol{M}$ and ${}^{\{\mathcal{C}\}}\boldsymbol{M}^*$ with respect to the current camera frame.

$$ \boldsymbol{S}_i = [X_i, Y_i, Z_i]^T,\ \boldsymbol{S}_i^* = [X_i^*, Y_i^*, Z_i^*]^T \quad (4) $$

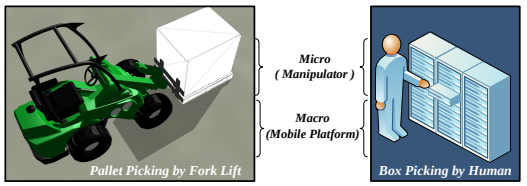


Fig. 3. Macro-micro configuration for picking objects in everyday life.

So, the correspondence pixels on the image can be expressed by

$$\boldsymbol{s}_i = [1, u_i, v_i]^T, \ \boldsymbol{s}_i^* = [1, u_i^*, v_i^*]^T \tag{5}$$

By assuming a pinhole model for the camera, relationships between the matching points become:

$$\boldsymbol{s}_i = \frac{1}{X_i}\boldsymbol{A}\boldsymbol{S}_i \ , \ \boldsymbol{s}_i^* = \frac{1}{X_i^*}\boldsymbol{A}\boldsymbol{S}_i^* \tag{6}$$

in which $\boldsymbol{A}_{3\times3}$ is known as the camera matrix. It can be the output of the camera's intrinsic parameters calibration described in Subsection III-E. Since the geometric properties of the object are known, the scale of normalized positions in (6) can be determined. Therefore, hereafter, we use $\{\mathcal{M}^*\}$ and $\{\mathcal{M}\}$ instead of comparing the $\boldsymbol{S}_i^*$ and $\boldsymbol{S}_i$ points, while it means consideration of the matched $\boldsymbol{s}_i^*$ and $\boldsymbol{s}_i$ features which include the depth or scale. It should be noted that we ignore the increment of uncertainties along the image depth axes, which relies on the z axis on the camera frame $\{\mathcal{C}\}$ as shown in Figure 1.

Installation of a camera in front of the forklift manipulator is practically impossible. Therefore, we define an imaginary coordinate attached to the manipulator, namely $\{\mathcal{F}\}$. Then, the output of the actual camera $\{\mathcal{C}\}$ attached to the vehicle's roof has to be transformed into $\{\mathcal{F}\}$ as a virtual camera attached to the manipulator. Although it could increase any error effects of the calibration results, the given calibration method in III-E keeps the robot's functionality and precision acceptable, practically.

Geometrically, $\{\mathcal{M}\}$ becomes coincident with $\{\mathcal{M}^*\}$ if and only if the current virtual camera $\{\mathcal{F}\}$ coincides $\{\mathcal{F}^*\}$. It should be noted that if the actual camera could be installed at the place of $\{\mathcal{F}\}$, the image Jacobian could be near singularity. On the other hand, visual servoing on the frame of the actual camera $\{\mathcal{C}\}$ is not useful because of the rank-deficiency of the image Jacobian. In other words, from the vision viewpoint, we are mapping the visual servoing problem because the pose of the ${}^{\mathcal{I}}\boldsymbol{C}$ is suitable for the object detection ($\dot{\boldsymbol{s}}$), while ${}^{\mathcal{I}}\boldsymbol{F}$ is suitable for movement detection ($\dot{\boldsymbol{v}}$). According to the above explanations, the error tensor for the visual servoing becomes

$$\boldsymbol{T}_e = \ {}^{\mathcal{F}}\boldsymbol{T}_{\mathcal{F}^*} \tag{7}$$

The desired value for such a matrix is to become near the identity matrix, $\boldsymbol{I}_{4\times4}$ (approximately).

B. Macro-Micro Formulation for Visual Servoing

The concept of designing macro-micro VS in this research was inspired by a comparison between an everyday life task and the pallet picking concept as shown in Figure 3. Asking a person to pick a known object means doing several steps for the person:

1) Remember its approximate location as a memorized first guess.
2) Go toward the location.
3) Look for the object while going to the given location.
4) After finding, walk a few steps to approach vicinity of the object.
5) Pick the object by hand.

The progress for pallet picking can be similar to the aforementioned steps. In our scenario, the first item is done by a stationary global planner which has access to the transportation database of all autonomous and manual machines. The second and third tasks are the main tasks for the macro robot. Then, the rest must be done by the micro robot. Let the macro robot become the vehicle itself, without consideration of the manipulator boom or fork which carries the actual camera coordinate $\{\mathcal{C}\}$. The manipulator part is considered to be the micro robot to which the virtual camera is connected.

This separation leads to several benefits including:

- Better accuracy (actuator and sensor).
- Benefit of the global planner for the set of coordinated autonomous vehicles, including the GIM Machine.
- Simplifying formulation and the controller.
- Maintaining satisfactory experimental results.

According to the definitions in Sections II and III-B, we divide the problem of pallet picking into a VS problem for the macro and micro robots. For that purpose we expand the transformations applicable to the coordinates such that:

$$\boldsymbol{T}_e = \ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}^{-1} \ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{B}^*} \ {}^{\mathcal{B}^*}\boldsymbol{T}_{\mathcal{F}^*} \tag{8}$$

in which ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}^{-1}$ and ${}^{\mathcal{B}^*}\boldsymbol{T}_{\mathcal{F}^*}$ determine the required movements for the micro robot, and the remaining, ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{B}^*}$ governs the demanded motion of the macro robot. This expansion leads to the new definition of two consistent errors for both macro and micro robots:

$$\boldsymbol{T}_{e_1} = \ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{B}^*} \tag{9}$$

$$\boldsymbol{T}_{e_2} = \ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}^{-1} \ {}^{\mathcal{B}^*}\boldsymbol{T}_{\mathcal{F}^*} \tag{10}$$

Mathematically speaking, it should be noted that (9) and (10) are *not* the direct results of (8), unless the macro robot arrives at its destination, $\{\mathcal{B}\} \longrightarrow \{\mathcal{B}^*\}$. We disregard this difference in order to simplify the macro and micro controllers formulations independently. However, it cannot cause any problems because before the arrival to the pallet, the principle axes of the coordinates for the macro and micro robots always become parallel to each other. Recall that the pallet is assumed to be approximately the same angle about the x-axis of the body. It is the only direction in which the robot has lack of actuation.

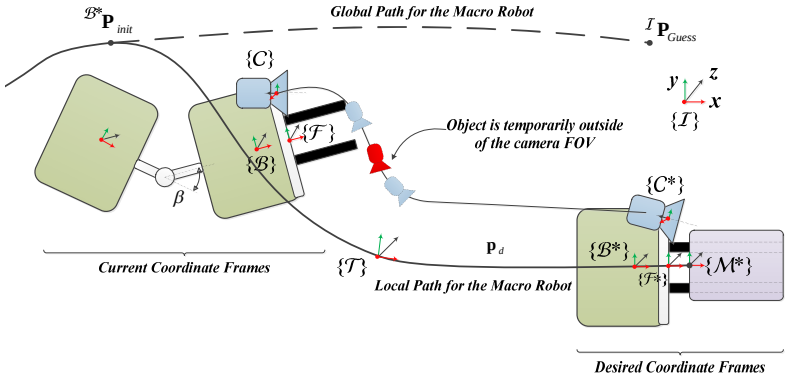


Fig. 4. Defined coordinate frames for the macro robot.

C. Motion Control for the Macro Robot

The aim of VS for the macro robot is to guide the body frame $\{\mathcal{B}\}$ to reach the desired coordinate $\{\mathcal{B}\}$. Its reflection can be found in the variation of (9) within its $x-y$ plane as well as the rotation around its z axis. Therefore, the transformation matrix for the macro robot could be written as ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}\,(x, y, \psi)$ because the other parameters for the location of the macro robot are not controllable. Due to the non-holonomic constraints of the machine, the control in the x-y plane takes a rather more complex structure. The macro robot's autonomous driving architecture within the global $\{\mathcal{I}\}$ is valid until receiving a task command that contains a VS flag. It navigates to the vicinity of the given location ${}^{\mathcal{I}}\boldsymbol{B}_{Guess}[x, y, \psi]$ defined in the inertial global frame $\{\mathcal{I}\}$. At the same time, by activating the vision toolbox, it looks for the given mark of the object. Since the FOV of the camera is reasonably bigger than the accuracy of the initial guess, we can be sure of finding the pallet. After detection of the pallet, the robot replies to the higher level with an update to the initial guess. Again, the planner transmits a new path. In contrast with the last path, the recent one is in the local coordinate $\{\mathcal{B}^*\}$.

A smooth and obstacle-free path $\boldsymbol{p}_d$ is generated, which extends from the current location ${}^{\mathcal{B}^*}\boldsymbol{P}_{init}[x, y, \psi]$ of the macro robot to the origin of $\{\mathcal{B}^*\}$, as illustrated in 4. Since the path is presented within $\{\mathcal{B}^*\}$ itself, ${}^{\mathcal{B}^*}\boldsymbol{B}^*$ becomes $[0, 0, 0]$. It should be noted that ${}^{\mathcal{B}^*}\boldsymbol{P}_{init}$ is the first position that the object has been detected in five consequent times. The start point of the path is fixed in the global coordinate and the end point is considered to be a relative coordinate to the origin of $\{\mathcal{B}^*\}$. Therefore, any change in the estimated pose of the pallet *will not* cause a time consuming update of the path by the planner. At the same time, any enhancement in the pose estimation affects the goal point of the path. This change leads to increments in the process time efficiency for real-time VS tasks, because a request for an AI-based path planning takes plenty of CPU time depending on the complexity of the planner configuration space [11]. From the ${}^{\mathcal{B}^*}\boldsymbol{P}_{init}$ point forward, the macro robot follows the motion control signals to zero the path following errors in order to minimize $\boldsymbol{T}_{e_1}^{-1}$ error.

Now, we define the error ${}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{B}}$, where $\{\mathcal{T}\}$ is a tangent frame to the path attached to $\boldsymbol{p}_d(\tau)$, where $\tau \in [0, 1]$ is the path parameter. It should be noted that $\{\mathcal{T}\}$ coincides with $\{\mathcal{B}^*\}$ for $\tau = 1$. The path following errors can now be defined:

$$\begin{aligned} e_x &= [1\ 0\ 0\ 1]\ {}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{B}} \\ e_\psi &= \arg({}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{B}}, \hat{\boldsymbol{z}}) = \arg({}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{F}}, \hat{\boldsymbol{z}}) \\ e_y &= [0\ 1\ 0\ 1]\ {}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{B}} = [0\ 1\ 0\ 1]\ {}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{F}} \end{aligned} \tag{11}$$

where the $\arg(.,.)$ function returns the angle argument of the rotational part of the transformation matrix around the axis defined in the second argument (z-axis in this case).

Notice that errors in the yaw and y-axis are the same in the body and in the fork frame. Therefore, if the machine follows the path closely and the pose estimates are accurate enough, the errors in the yaw and y-axis will be sufficiently small when the fork frame reaches the pallet. This makes the rotational part of (8) equal to (9) and (10). It should also be noted that the closer the machine is to the pallet, the more accurate the pose estimates. Therefore, even if the trajectory of the body frame in the real world is initially not the same as $\boldsymbol{p}_d$, it converges to it as the pose estimation errors decrease. Note that it is possible to design the macro robot controller as an eye-in-hand visual servoing problem, however, following the planned path is more robust [12]. Particularly, in a multi-robot and multi-machine environment, the macro robot's path is supposed to be coordinated with the others. *Definition: Path following problem.* Design a control law for steering signal u_s to drive e_ψ and e_y to zero. A solution to this problem is devised in [9], which can be summarized as follows. We showed in [13] that the motion of the front unit of the GIM machine closely follows the kinematics:

$$\begin{aligned} \dot{x} &= v_x \cos\psi \\ \dot{y} &= v_x \sin\psi \\ \dot{\psi} &= \omega_z \end{aligned} \tag{12}$$

$$\omega_z = \frac{l_r\dot{\beta} + v_x \sin\beta}{l_f \cos\beta + l_r}, \tag{13}$$

where β is the body articulation angle and l_f, l_r are geometrical parameters (see Figure 2). Notice that (12) is a simplified version of [4]:

$$\begin{aligned} \dot{x} &= v_x \cos\theta\cos\psi, \;\; \dot{y} = v_x \cos\theta \sin\psi \\ \dot{\psi} &= \omega_z \frac{\cos\phi}{\cos\theta} + \omega_y \frac{\sin\phi}{\cos\theta} \end{aligned} \tag{14}$$

in the horizontal plane, but it is accurate enough for feedback control. The solution in [13] provides a control law for the desired angular velocity $\omega_z^c = C_{PF}(e_x, e_y, e_\psi)$, where the speed v_x remains free and can be controlled separately. Freedom in assigning the linear speed is instrumental in our case, because without causing instability in the path-following controller, the machine can be slowed down or stopped. For example, in the case where manipulator errors are not yet small enough, collision with the pallet can possibly be avoided. Now, given ω_z^c, we use (13) to calculate the required steering valve command:

$$\begin{aligned} \dot{\beta}^c &= -\frac{v_f}{l_r}\sin\beta + \left(\frac{l_f}{l_r}\cos\beta + 1\right)\omega_f^c \\ u_s &= K\dot{\beta}^c \end{aligned} \tag{15}$$

where we assumed that the valve command is proportional to the speed of the body articulation.

D. Visual Servoing for the Micro Robot

As described in the previous subsection, the path follower controller is capable of driving the macro robot to the vicinity of the pallet with a fairly small error, $\{\mathcal{B}\} \longrightarrow \{\mathcal{B}^*\}$. As a hydrostatic rough terrain forklift, the macro robot is not suitable for the motions below $20cm$ in actuation accuracy and repeatability, especially along its longitude. Therefore, we use the manipulator to adjust the height and angle of the forks (z and θ) in addition to compensation for the macro robot error in the ${}^{\mathcal{B}}\boldsymbol{x}$ direction. As shown in Fig.1, there are three joint parameters and three degrees of freedom for the micro robot such that:

$$\begin{aligned} \dot{\boldsymbol{q}} &= \boldsymbol{J}(\boldsymbol{q})\boldsymbol{K}\boldsymbol{v} \\ \begin{bmatrix} \dot{\theta}_1 \\ \dot{L} \\ \dot{\theta}_2 \end{bmatrix} &= \begin{bmatrix} k_1 L\sin\theta_1 & k_2 L\cos\theta_2 & 0 \\ -k_1\cos\theta_1 & k_2\sin\theta_2 & 0 \\ 0 & 0 & -k_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_x \\ \boldsymbol{v}_z \\ \boldsymbol{\omega}_y \end{bmatrix} \end{aligned} \tag{16}$$

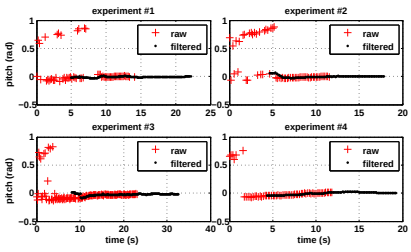


Fig. 5. Pitch angle errors, before and after filter in four representative tests.

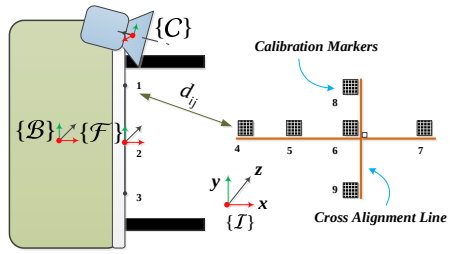


Fig. 6. Calibration of the camera's extrinsic parameters.

in which the $\boldsymbol{K}$ stands for the coefficients of conversion between the oil debit of hydraulic proportional valves and their displacements, and the movements of the joints. Toward this formulation, we define the errors in the $x-z$ plane:

$$\begin{aligned} e_\phi &= \arg({}^{\mathcal{F}^*}\boldsymbol{T}_{\mathcal{F}}, \hat{\boldsymbol{y}}) \\ e_z &= [0\;0\;1\;1]\;\; {}^{\mathcal{F}^*}\boldsymbol{T}_{\mathcal{F}} \\ e_x &= [1\;0\;0\;1]\;\; {}^{\mathcal{F}^*}\boldsymbol{T}_{\mathcal{F}} \end{aligned} \tag{17}$$

Recall that for small errors in the macro robot, these errors can be considered as corresponding components of the error tensor for the micro robot addressed in (10). If we limit the control space to just the micro robot joints and degrees of freedom, such error components can also be presented as an error among the matched features introduced in (4), transformed into the virtual camera coordinate:

$$\boldsymbol{e}_{vs} = {}^{\mathcal{F}}\boldsymbol{T}_{\mathcal{C}} \left({}^{\mathcal{C}}\boldsymbol{S}_i - {}^{\mathcal{C}}\boldsymbol{S}_i^* \right) \tag{18}$$

Substituting the PBVS control law as addressed in [14] and [12], and to enforce an exponential decoupled decrease of the error, we apply $\dot{\boldsymbol{e}_{vs}} = -\lambda \boldsymbol{e}_{vs}$ to derive:

$$\dot{\boldsymbol{q}} = -\lambda \boldsymbol{J}(\boldsymbol{q})\; \boldsymbol{K}\; \boldsymbol{e}_{vs} \tag{19}$$

The manipulator actuators are driven by the pressure-compensated proportional valves, and it is reasonable to assume that:

$$\begin{aligned} u_1 &= c_1 \text{sat}(\dot{\theta}_1) \\ u_2 &= c_2 \text{sat}(\dot{\theta}_2) \\ u_L &= c_3 \text{sat}(\dot{L}) \end{aligned} \tag{20}$$

where u_1, u_2, u_L are the valve commands. Employing the Lyapunov function $V = 0.5k_1 e_x^2 + 0.5k_2 e_x^2 + 0.5e_\phi^2$, it can be shown that the control law addressed in this section can drive the micro robot errors to zero. In the equations above:

$$\text{sat(a)} = \begin{cases} 1 & \text{if } a > 1 \\ -1 & \text{if } a < -1 \\ a & \text{otherwise} \end{cases} \tag{21}$$

is a saturation function, and gains $c_i,\; i = 1, 2, 3$ define the velocity bounds of the actuators and unit conversions.

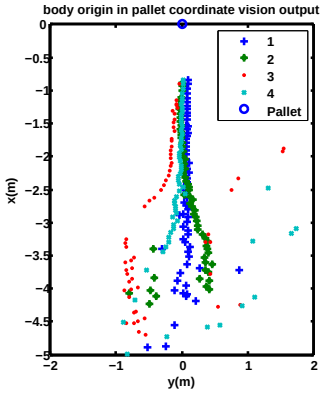


Fig. 7. Marker detection output, raw data. Five representative tests

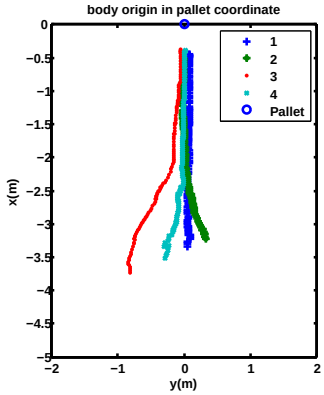


Fig. 8. Tracking filter output. Five representative tests

E. Camera Calibration

Since the camera provides integrated photometric calibration, the camera calibration process consists of the calibration for the intrinsic and extrinsic parameters. While we assume that camera distortion is independent of its resolution, the tuning of the intrinsic parameters has been done through the Alvar toolbox [5] based on Open-CV [15]. The only notable parts are the image's bigger radial distortion coefficients, r_1 and r_2, compared to the tangential distortion coefficients, t_1 and t_2, of the camera image:

$$(r_1, r_2, t_1, t_2) = (-0.186, -0.039, 0.0002, -0.0147) \quad (22)$$

Because the pose of the detected objects must be mapped in different coordinates, calibration of the extrinsic parameters becomes critical for the system. To overcome this issue, we set up a cross-shaped base with nine markers as shown in Fig.6 in the camera FOV. The body frame has also been indicated by three additional markers. Then, the ${}^{C}\boldsymbol{a}_i$ vectors representing the position of the markers in the camera coordinate are extracted. The $d_{ij}, i = (1, 2 \ldots, 9), j = (1, 2, 3)$ distances are measured, and the set of 27 equations are optimized to extract ${}^{B}\boldsymbol{a}_i$ components by hand measurements. Transformation of the two representations of all points extracted by the camera and hand measurements gives the linear transformation between the two coordinates. It should be noted that the vector between points 8-9 and 5-7 was calculated. The principle axes of the rotation matrix can be written:

$$\boldsymbol{R}_x = \frac{\boldsymbol{a}_8 - \boldsymbol{a}_9}{\|\boldsymbol{a}_8 - \boldsymbol{a}_9\|}, \boldsymbol{R}_y = \frac{\boldsymbol{a}_5 - \boldsymbol{a}_7}{\|\boldsymbol{a}_5 - \boldsymbol{a}_7\|}, \boldsymbol{R}_z = \boldsymbol{R}_x \times \boldsymbol{R}_y \quad (23)$$

IV. Experimental Results

It is reasonable to assume that, in practice, the position of the pallet is known with certain accuracy. In the case of the GIM machine, the pallet locations are registered with an accuracy of 0.5 m using on-board GPS and the inertial navigation system after the last placement. This accuracy is enough to send the machine to the proximity of the pallet, and therefore, the face of the pallet is visible to the vision system. Based on this observation, we have set our experiments as follows. In all the test runs, the machine started from different initial configurations at distances of 7-8 m in front of the pallet and facing toward the pallet with different initial headings. Initially, the lateral distances of the body frame with respect to the center of the pallet were in the range of 1 m. The machine then moves in a straight line until the pallet is detected and the pallet tracking filter is initialized. After the filter is initialized, PBVS is activated. The on-board camera was a Gima GO423C with 1296×966 resolution. Recall that owing to the reduced number of pulses per revolution of wheel odometry devices, speeds under $0.25m/s$ are not reliably measured. Therefore, in the experiments reported in this paper, we have set the speed commands to a constant value of $0.3m/s$ or zero (stop). The actual speed varied from 0.3–$0.4m/s$ in the test runs. We conducted a total of 13 experiments. The experiments were conducted on a cloudy day between 14:00 and 15:00 in December 2012. A few of the final experiments were performed with the machine's front lights on[1]. All the experiments were successful.

Figure 7 shows the trajectory of the body frame in the pallet frame in the x-y pallet plane. For clarity of presentation, we show only five representative tests: two each starting from the left and the right side of the pallet and one from the middle. This figure shows the marker detection output before filtering. It can easily be seen that detection is very unreliable at large distances and that it improves as the machine approaches the pallet. Figure 8 shows the same trajectories after filtering. Notice that how the trajectories have been smoothened out. Figure 5, shows the pitch, yaw, and z-axis errors, respectively, versus time for experiments 1-4 in four subplots (the numbers correspond to the numbers in Figure 7). Some observations can be made from these figures. The tracking filter starts long after the vision output becomes available, and it continues producing

[1]In this latitude, sun sets around 15:00 in December.

pose estimates even after the vision output is not available. It is visually clear how the tracking filter has smoothened the vision output.

V. Conclusion

In this paper, we introduced a position-based VS control framework and a fault tolerant pose estimation algorithm. This method was developed and experimented on a small-sized forklift truck. The choice of the coordinate frames and references provides a natural framework to divide the error space into two orthogonal planes and to fuse the vision output with the dead-reckoning data. One plane was then controlled by drive-steering and the other, by the manipulators. The redundancy resolution (along the x-axis) was achieved by the state-machine (the details are omitted). Real-world experiments showed the robustness and efficacy of the methodology for the problem in hand. Future work includes the optimal fusion of the image processing data with dead-reckoning instead of averaging. A truly reliable integration of vision into VS, especially in outdoor conditions, requires the extension of the logic control and planning algorithm for better exception handling. In real-world scenarios, there are always broken pallets that make pallet detection a real challenge. Alternative detection methods without the use of markers, for example, using active sensors, will also improve the reliability of the system.

Appendix I
Parametric Path for The Experiments

For a smooth path in our experiments, we have used a cubic Bezier curve,

$$\begin{aligned} {}^{\mathcal{B}^*}P_d(s) =& (1-s)^3 {}^{\mathcal{B}^*}P_1 + 3s(1-s)^2 {}^{\mathcal{B}^*}P_2 \\ &+ 3s^2(1-s)^3 {}^{\mathcal{B}^*}P_3 + s^3 {}^{\mathcal{B}^*}P_4 \end{aligned} \tag{24}$$

with the following control points:

$$\begin{aligned} {}^{\mathcal{B}^*}\boldsymbol{P}_1 &= [\quad {}^{\mathcal{B}^*}x_{init} \quad {}^{\mathcal{B}^*}y_{init} \quad], \\ {}^{\mathcal{B}^*}\boldsymbol{P}_2 &= [\quad {}^{\mathcal{B}^*}x_{init} + 1.4 \quad 0 \quad], \\ {}^{\mathcal{B}^*}\boldsymbol{P}_3 &= [\quad {}^{\mathcal{B}^*}x_{init} + 0.7 \quad 0 \quad], \\ {}^{\mathcal{B}^*}\boldsymbol{P}_4 &= [\quad 0 \quad 0 \quad]. \end{aligned} \tag{25}$$

This choice guarantees that path $\boldsymbol{p}_d$ is tangential to the x-axis of $\{\mathcal{B}^*\}$ at its origin. Therefore, the object is in sight of the camera.

References

[1] W. Xie, Z. Li, X. Tu, and C. Perron, "Switching control of image-based visual servoing with laser pointer in robotic manufacturing systems," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 520–529, 2009.

[2] L. Baglivo, N. Biasi, F. Biral, N. Bellomo, E. Bertolazzi, M. Da Lio, and M. De Cecco, "Autonomous pallet localization and picking for industrial forklifts: a robust range and look method," *Measurement Science and Technology*, vol. 22, no. 8, p. 085502, 2011.

[3] Z. Miljković, N. Vuković, M. Mitić, and B. Babić, "New hybrid vision-based control approach for automated guided vehicles," *The International Journal of Advanced Manufacturing Technology*, pp. 1–19, 2012.

[4] M. M. Aref, R. Ghabcheloo, A. Kolu, M. Hyvönen, K. Huhtala, and J. Mattila, "Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics, Automation and Mechatronics (RAM), Manila, Philippine*, pp. 218–224, 2013.

[5] VTT Technical Research Centre of Finland, "Alvar toolkit website." http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/index.html, 2013.

[6] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Transactions On Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.

[7] G. Lopes and D. Koditschek, "Visual servoing for nonholonomically constrained three degree of freedom kinematic systems," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 715–736, 2007.

[8] Finnish Centre of Excellence in Generic Intelligent Machines (GIM) Research. http://gim.aalto.fi, 2013.

[9] R. Ghabcheloo, M. Hyvönen, J. Uusisalo, O. Karhu, J. Järä, and K. Huhtala, "Autonomous motion control of a wheel loader," in *ASME Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control*, Hollywood, CA, US, 2009.

[10] A. Aghamohammadi, A. Tamjidi, and H. Taghirad, "SLAM using single laser range finder," *Proceedings of 17th World Congress of the International Federation of Automatic Control (IFAC)*, 2008.

[11] A.-a. Agha-mohammadi, S. Chakravorty, and N. Amato, "Firm: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements," *International Journal of Robotics Research (IJRR)*, vol. 33, no. 2, pp. 268–304, 2014.

[12] F. Chaumette and S. Hutchinson, "Visual servo control. II. advanced approaches [tutorial]," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[13] R. Ghabcheloo and M. Hyvönen, "Modeling and motion control of an articulated-frame-steering hydraulic mobile machine," *Proceedings of the 17th Mediterranean Conference on Control and Automation*, Greece, June 2009.

[14] G. Silveira and E. Malis, "Real-time visual tracking under arbitrary illumination changes," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–6, IEEE, 2007.

[15] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

Publication III

M. Aref, M., Oftadeh, R., Ghabcheloo, R., and Mattila, J., "Fault tolerant control architecture design for mobile manipulation in scientific facilities," *International Journal of Advanced Robotic Systems*, vol. 12, p. 4, 2015

INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Fault Tolerant Control Architecture Design for Mobile Manipulation in Scientific Facilities

Regular paper

Mohammad M. Aref*,1, Reza Oftadeh[1], Reza Ghabcheloo[1] and Jouni Mattila[1]

[1]Department of Intelligent Hydraulics and Automation (IHA), Tampere University of Technology, Tampere, Finland.
* Corresponding author E-mail: m.aref@ieee.org

Originally published in Volume , Year

Abstract This paper describes one of the challenging issues implied by scientific infrastructures on a mobile robot cognition architecture. For a generally applicable cognition architecture, we study the dependencies and logical relations between several tasks and subsystems. The overall view of the software modules is described, including their relationship with a fault management module that monitors the consistency of the data flow among the modules. The fault management module is the solution of the deliberative architecture for the single point failures, and the safety anchor is the reactive solution for the faults by redundant equipment. In addition, a hardware architecture is proposed to ensure safe robot movement as a redundancy for the cognition of the robot. The method is designed for a four-wheel steerable (4WS) mobile manipulator (*iMoro*) as a case study.

Keywords mobile robot, autonomous vehicle drive, architecture design, remote handling, cognition.

1. Introduction

Scientific facilities *(SFs)* and research infrastructures are indispensable in realizing the dreams of human beings. They necessitate a combination of state-of-the-art technologies together with reliability, availability, maintainability and safety *(RAMS)* [1]. The development of autonomous or semi-autonomous mechatronic systems for such an environment is a challenging issue. The challenge arises when the facilities turn into hazardous environments for humans and/or machines. In this case, a robot is utilized in order to minimize human intervention in the non-suitable conditions of the workspace. The robot is used in order to reduce the risks for humans "as low as reasonably possible or achievable" *(ALARA* or *ALARP)* [2]. Consequently, it is not acceptable to employ robots or any complex system in SFs if they will become an insecure system which needs on-site maintenance [3]. Therefore, the service robot has to be safe in order to avoid undesirable changes in the environment, and it has to be fault tolerant in order to recover itself without direct human intervention, especially for leaving the test environment.

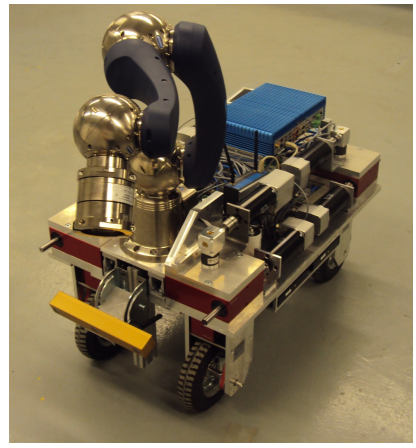


Figure 1. The four wheel steerable mobile manipulator (*iMoro*).

This necessity is a key requirement for such a robot or any device acceptable for SFs [4].

The aim of this paper is a discussion of a fault-tolerant architecture for the fulfilment of the above-mentioned requirements as well as common needs for service robots, such as flexibility and modularity.

The contributions of this paper can be divided into higher-level strategies and lower-level practical issues for design improvements of a safety standards-compatible, fault-tolerant robot for SFs. This paper reflects the decomposition of a high-level mission to generate several commands and the structural behavior of the robot. Next, we describe the necessary modules for achieving the required functionalities of the robot. Moreover, to make the robot compatible with safety requirements, a safety anchor is proposed to provide a low-cost redundant decision-making unit. It maintains the robot's behavior during its mission as well as its exit in the case of unforeseen problems. The design considers the requirements for a robot tasked to explore the Large Hadron Collider (LHC) tunnels of the European Organization of Nuclear Research (*CERN*) as a sample SF.

Generally speaking, a cognition architecture can be divided into four control schemes: deliberative, reactive, hybrid and behavior-based [5];

Deliberative Sense, plan, and then act.

Reactive React on a predefined action to a perception relation.

behavior-based Distribute thinking over acting.

Hybrid Combine deliberative and reactive – think slowly, react quickly.

Although all autonomous robots have to include deliberation [6, 7], a pure deliberative architecture cannot accomplish the requirements of contemporary applications [8]. In some cases, it is used in assisting the operator using autonomous functionalities, and in order to predict possible accidents [9]. A study conducted recently addressed several aspects of a deliberative control schema as a divide-and-conquer problem-solving strategy [7].

The hybrid control topology is developed to overcome the limitations of deliberative control and reactive control. It uses the reactive control scheme to respond to environmental changes and high-level planning to handle complex tasks [10, 11]. During this research, we partially apply a hybrid control topology for the architecture of a mobile platform. Moreover, the safety of the mobile manipulator is another issue addressed in this study via a service robot with a lightweight arm [12].

The aim of this paper is to determine a strategy for utilizing a mobile manipulator in inspections planning and scheduling [13] and interventions instead of human intervention[2]. In order to lower the risks, it is more reliable to control the behavior of the autonomous identification or model-based control process by a safe configuration [14]. We propose a redundant safety-compatible architecture to tolerate the faults of the main architecture. It contains a set of actions activated by pre-selected processes through triggers.

2. Problem Definition

This paper represents an architecture for a *Modular Mobile Manipulator System* (MMMS) to be used as a road map for the design and implementation process of mobile robots targeting SFs. The concepts are designed based on the statement of needs and requirements as a part of the PURESAFE project systems engineering documents[4]. The project, *"Preventing hUman intervention for incREased SAfety in inFrastructures Emitting ionizing radiation" (PURESAFE)*, has been initiated to advance theoretical and experimental knowledge regarding semi-autonomous mobile manipulation within the ionizing radiation and contaminated environment of accelerators, in particular the tunnels inside the European Organization for Nuclear Research *(CERN)* and the GSI Helmholtz Centre for Heavy Ion Research, Facility for Anti-proton and Ion Research *(GSI/FAIR)*. While perception and navigation are challenging issues for mobile platforms, robot intervention in SFs and hazardous environment imply more significance of architecture design considering fault tolerance and safety in robot cognition.

Although a variety of requirements should be considered in the architecture design of the robot, we only discuss a sample solution to a task scenario described in 2.1. In addition to the challenging issues of mobile manipulator architectures, this research considers an important and mandatory constraint for the robot:

> "As a service robot, the robot should not cause human intervention in the test environment. In case of failure in mission, it should be able to return to its station."

This restriction affects the risk assessment results and highlights the immobility of the robot as the most severe failure of the robot. Moreover, the presence of the robot in the neighborhood of scientific test equipment during operation or maintenance time, arises failure severity even with a low occurrence chance. For instance, any damage to the experiment infrastructure results in a SF shutdown and human intervention in hazardous environment. The robot perception, planning and navigation should be safe and reliable so as to guarantee avoiding any destruction or unplanned shutdown in the SF and to ensure the robot's ability to escape from the test environment in an emergency. To overcome this problem, the proposed architecture design of the robot will be addressed in the following sections. We use ***I**HA **Mo**bile **Ro**bot* (*iMoro*) as a case study for experiments and the simulations proposed in this paper. It is also under investigation for the further development of robot intervention for SFs by Tampere University of Technology.[1]

Similar to other complex systems, SFs are the subject of frequent inspections and maintenance during their lifetimes. Moreover, modifications, repairs and inspections have to be accomplished over a short duration to prevent the shutdown of scientific experiments, aging of

[1] Robot videos are available at: https://www.youtube.com/channel/UCxemrk8NrIj-db6h06VBVLA

the robot body, the effects of ionizing or non-ionizing contamination, and hardware corrosion. In the following section, we will define a sample mission for *iMoro*and describe the necessary hardware and software components in the rest of the paper that are compatible with the described constraints.

2.1. Task Description

A mobile manipulator can have wide range of applications within SFs. In this section, to summarize the system description, the authors define a mission as a sequence of tasks to develop necessary hardware and software components. A sample inspection covers the following description:

> *Mission* Receive commands on control room Human Machine Interface *(HMI)*: In point "A", Fast forward to section "B" addressed on the CAD-based map. Prepare for tele-operation around the device number "36" and then inform the operator for a semi-autonomous operation. Touch the object defined by the operator. Inspect the area defined by the operator and transmit preliminary results. Finally, come back to your station and give detailed data report. [2]

To task the robot mobile manipulator inside a hazardous area of a SF, there are several limitations and requirements. Some of the key requirements are listed below:

Requirement 1. The physical or chemical effects of materials and test equipment in the environment can affect the work performance and lifetime of electrical components.

Requirement 2. The facilities are mostly designed for humans, the test equipment or special machines accessible through different corridors. There is no dedicated environment or specific landmarks - or setups for a robot except its station. In other words, the environment is unstructured for a robot.

Requirement 3. A lack of communication can frequently occur. The robot shall be tolerant to failures of Wi-Fi or delays of GSM transmitted signals.

Requirement 4. The purpose of utilizing a robot is to prevent human intervention in hazardous environments. Therefore, during the operation of the robot, the area is not suitable for humans, and all robot operations shall be remotely controllable.

Requirement 5. In case of failure, the robot should be able to leave the SF.

Considering these requirements, the mission can break down into sequential tasks. Since, in most cases, the mobile manipulator applications are similar to the described mission, we describe here those tasks, while we describe the overall solution later on.

Task 1. Initialization

Get the global map ${}^{\mathcal{I}}\mathbb{M}$, initial pose ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}_0}$, destination ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}f}$[3], a final workspace ${}^{\mathcal{I}}\mathbb{W}$, the initial status of the robot, and calibration parameters. Monitor output of the sensors and command their initialization. Verify their functionality and consistency with the initialized variables. Initialize the actuators. Perform a predefined motion and examine the actuators and sensors by verifying the consistency of the commands and feedback.

Task 2. Planning

Convert the global map into a 2D occupancy grid ${}^{\mathcal{I}}\mathbb{M}'$. Plan an energy-optimal obstacle-free path from ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}_0}$ to ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}f}$ within ${}^{\mathcal{I}}\mathbb{M}'$. Plan for wondering around the position ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}f}$ within ${}^{\mathcal{I}}\mathbb{W}$. Estimate the elapsed time and power consumption for all of the future tasks programmed within the mission with a safety margin. Compare the power status of the batteries with the estimated power consumption. Wait for extra charge in case of insufficient power.

Task 3. Autonomous Running Mode

Run these modules in parallel, repetitively:

Perception Perceive any change in the environment and status of the robot compared to the preliminary planned motions.

Planning Look–ahead and plan for local movements regarding the high–level commands received and the local obstacles and costs for doing the job.

Actuation Run the planned actions while receiving the joints' feedback for the closed–loop actuator–level control.

Exception Handling In case of single point failures, bypass the current sequence and save the algorithm. Batch the logs and transmit to the supervisory room.

Fault Detection Once a while, conduct a survey among the software and hardware modules with their model. In case of the detection of unpredicted motions and feedback signals, decide for one of these solutions. Run the relevant fault management component. Call the operator for decision making or manual driving. Run the Safety Anchor (described in Section 5).

Watchdog Independently maintain surveillance of the robot software and hardware modules during the operation.

Task 4. Update Mode

After reaching the target position, map the workspace on a detailed map. Extract the optimum features for accurate localization. Define the local stationary coordinate attached to the features. Obtain transformations between the new coordinate and the predefined coordinates. Upload the collected *information* to the supervisory room while considering possibility of data loss and a delay in connection. Finalize the autonomous mode and switch to the semi-autonomous mode. Call the operator through all of the available networks, such as Wi-Fi or GSM.

[2] The numbers and names here are only samples for non-quantitative addresses by the operator.

[3] It should be noted that ${}^{\mathcal{A}}\boldsymbol{T}_{\mathcal{B}}$ stands for the homogenous transformation representing a coordinate frame $\mathcal{B}$ with respect to another coordinate frame $\mathcal{A}$

Task 5. Tele-operation Mode

Let the operator define the movements in the virtual environment and execute the commands in order. Update the current status for the operator by the minimal transmission of data to overcome any lack of connection bandwidth and to enable the possibility of augmented reality. In the case of selecting an object, run the visual servoing to approach the vicinity of the target object and do any inspection or manipulation as commanded by the operator. At the same time, maintain the prediction of power consumption. Set an alarm when the power reaches the limit of the safety margin. Quit from the position if the operator has finished the operation or else if the power limitation implies a forced return.

Task 6. Finalization Mode

Return to the nearest safe station. Set all the actuators to their home positions. Next, transmit all the gathered data by the high bandwidth connection, including the final status of the robot and its components and the evaluation of the modules that are gathered during run mode. Clear any temporary information. Recharge the embedded power source.

Task 7. Standby Mode

Turn off any high-consumption hardware and keep the control blocks in sleep mode. Wait for the next trigger and another mission.

According to the tasks and their descriptions, the rest of this paper gives an overview of the hardware and then describes the necessary software modules and the architecture designed for the concisely described tasks.

2.2. Sensors

The robot is equipped with several sensors to collect data from its components and the environment. We describe them in the following subsections.

2.2.1. Laser Range Finder

(LRF) The laser scanner or LRF provides a 2D map of the environment based on the reflection of its infrared beams. The output data comprise pairs of distance and angle based on a predefined polar coordinate attached to the $\{\mathcal{LRF}\}$ coordinate. The origin of $\{\mathcal{LRF}\}$ which is virtually defined as attached to the axis of rotation of its motor in such a way that the $+z$ direction is collinear to the axis of rotation. Therefore, all the transmitted points rely on the $x-y$ plane of $\{\mathcal{LRF}\}$. The corresponding Cartesian representation of the detected point number k by the laser scanner can be defined as ${}^{\mathcal{LRF}}\Lambda_k(x_k, y_k, 0)$. The number of detected points during one scan, $\max(k)$, depends on the *field of view* (FOV) and the angular resolution of the laser scanner. Note that the source of systematic and random errors in the positions of the detected points are still in the polar coordinate as errors in the distance and angle [15, 16]. We can write for the vector of points:

$$ {}^{\mathcal{B}}\mathbf{\Lambda}_{\mathcal{LRF}} = {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{LRF}}\ {}^{\mathcal{LRF}}\mathbf{\Lambda} \tag{1}$$

The output of the laser scanner determines the position of the robot relative to its environment by a *Simultaneous Localization and Mapping Toolbox* (SLAM) as described in Subsection 4.2. Moreover, its output can be a local map for obstacle avoidance and heading estimation.

Despite the fact that the LRF is a robust sensor for the navigation of the mobile platform, it fails in the detection of transparent and reflective materials in the environment [17]. It is also hard to detect darker materials at longer distances. Regarding these issues, the sensor can be considered as a safe sensor for short distances (under $20m$). Besides the accurate distance angle measurements, it is possible to define warning zones on the same plane. In this case, the scanner transmits the corresponding situation of each zone. This communication can be safe and totally independent of the points measurements, and can be utilized in the safety anchor mechanism described in Section 5. The warning zones can be defined as relay outputs or logical switches on the hard-wired network.

2.2.2. Wheel Odometry

Each wheel of the robot is equipped by an encoder on the steering and driving servo motors. The measured values for the joint parameters used in the kinematics and then in the path-following controller are estimated based on these data. The steering actuators provide ϕ_i as a feedback, and the driving actuators feed the driving speeds v_i. These parameters have been used to determine the velocity vectors of each wheel as shown in Figure 2. The resultant data determine the robot's velocity v and angular velocity ω_b and motion direction ψ_v.

2.2.3. Current Sensor

The overall current of the mobile manipulator passes through a current sensor. The measured sensor data can be used for the power consumption and then for the estimation of the remaining autonomy time. Moreover, it can sense any abnormal electrical loss during standby or the run-time of the robot. The energy management methods[3] are also applicable to analyze the information gathered from this sensor.

3. Kinematics

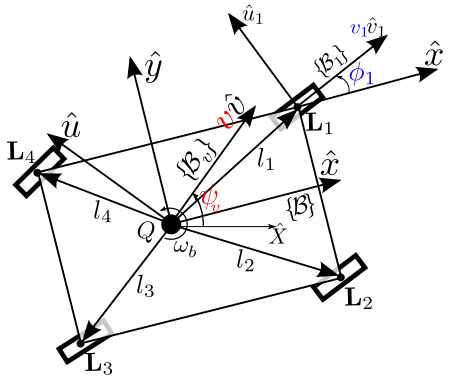


Figure 2. Denoted kinematic parameters.

In this section, we present a kinematics formulation in the velocity space for the 4WS mobile platform. Although the equation is quite straightforward, it embeds substantial information pertinent to the relative velocities of the body and wheel modules. A general solution is then given for estimating the body's linear and angular velocity using kinematic constraints and encoders data. The solution uses the least squares method to measure the relative consistency of the legs' velocities, thus detecting faulty encoder data. The mobile robot can be fully functional even after losing the functionality of one of its wheel modules. Hence, this approach can be used to single out a faulty leg during the robot's operation.

3.1. Inverse Kinematics

Figure 2 shows a schematic view of a 4WS mobile robot. The denoted parameters for the configuration of the body and the leg modules are presented in the figure. The coordinate frame $\boldsymbol{U}\{\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}\}$ is the inertial frame. Frame $\boldsymbol{B}\{\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}\}$ is a fixed-body frame defining the heading of the robot, and $\boldsymbol{B}_v\{\hat{\boldsymbol{v}}, \hat{\boldsymbol{u}}\}$ is the velocity frame, that is, the unit vector $\hat{\boldsymbol{v}}$ determines the direction of the robot's base linear velocity vector. Both $\boldsymbol{B}$ and $\boldsymbol{B}_v$ are attached to the robot's base at point $\boldsymbol{Q}$, which can be chosen at will. The angles ψ_v and θ_B are the angles $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{x}}$, respectively, in $\boldsymbol{U}$. The scalar value v is the magnitude of the linear velocity of point Q. The variables $\omega_B = \dot{\theta}_B$ and $\omega_v = \dot{\psi}_v$ are the angular velocities of the base and of $\hat{\boldsymbol{v}}$, respectively. The constant vectors $\overrightarrow{\boldsymbol{QL}_i}$ presented in frame $\boldsymbol{B}$ are denoted by ${}^{\boldsymbol{B}}\boldsymbol{\ell}_i, i = 1..4$. The angle ϕ_i is the heading angle of the i^{th} leg while $v_i\hat{\boldsymbol{v}}_i$ is the velocity vector of the attachment point $\boldsymbol{L}_i$. The steering and speed control commands for leg i are calculated using ϕ_i and v_i, respectively. The following kinematic relations hold:

$$ {}^{\boldsymbol{B}}\hat{\boldsymbol{v}} = \boldsymbol{R}(\psi_v - \theta_B)[1\ 0\ 0]^T \tag{2a} $$

$$ v_i\ {}^{\boldsymbol{B}}\hat{\boldsymbol{v}}_i = v\ {}^{\boldsymbol{B}}\hat{\boldsymbol{v}} + \omega_B(\hat{z} \times {}^{\boldsymbol{B}}\boldsymbol{\ell}_i) \tag{2b} $$

in which, $\hat{z} = [0\ 0\ 1]^T$ and $\boldsymbol{R}(\psi_v - \theta_B)$ is the rotation matrix with angle $\psi_v - \theta_B$ around the z-axis, that is, frame $\boldsymbol{B}_v$ in $\boldsymbol{B}$. For motion control purposes, once the control signals ψ_v, ω_B and v are calculated for the body frame, the desired signals ϕ_i and v_i can be derived using above kinematics equations. Notice that only angle ϕ_i and not its derivatives appears in (2a-2b) as the angle of vector ${}^{\boldsymbol{B}}\hat{\boldsymbol{v}}_i$. This fact justifies having steering actuators to be controlled in position mode.

3.2. Forward Kinematics

As mentioned earlier, the speed of the drive actuators and the positions of the steering actuators are measured, that is, positions ϕ_i for the steering actuators and speeds v_i for the driving actuators. In this subsection, the goal is to derive ψ_v, ω_B and v based on the encoders' measurement data, that is, the sets $\{v_i, \phi_i\}$ for $i = 1 \ldots 4$. Clearly, the relative velocity of the wheel modules with respect to each other must be zero. In other words, the set $\{v_i, \phi_i\}$ applied to a leg should be consistent with other sets. Although the set points for the steering and drive commands are derived based on the equation (2b), and hence consistent, the actual value of the actuators' encoders will not comply exactly with the kinematics relations due to different loads and different servo dynamics. Since the legs are connected to a rigid base, this inconsistency causes the legs to slip with respect to one another. This slippage is different than the overall slippage of the robot, and cannot be observed through odometry data. In order to measure the consistency of the actuators' actual values, we define the following measure:

$$ e_i = \frac{1}{4}\sqrt{\sum_{j=1, j\neq i}^{4} e_{ij}^2} \quad , i \in \{1,2,3,4\} \tag{3} $$

where:

$$ e_{ij} = (v_i\ {}^{\boldsymbol{B}}\hat{\boldsymbol{v}}_i - v_j\ {}^{\boldsymbol{B}}\hat{\boldsymbol{v}}_j).\boldsymbol{\ell}_{ij} \tag{4a} $$

$$ \boldsymbol{\ell}_{ij} = \frac{{}^{\boldsymbol{B}}\boldsymbol{\ell}_i - {}^{\boldsymbol{B}}\boldsymbol{\ell}_j}{||{}^{\boldsymbol{B}}\boldsymbol{\ell}_i - {}^{\boldsymbol{B}}\boldsymbol{\ell}_j||_2} \tag{4b} $$

In the above equations e_{ij} is the relative velocity line L_i to L_j. Clearly, if both wheels are synchronized perfectly and no slippage occurs, e_{ij} is zero. However, as explained before, in practice these variables are not zero and the value e_{ij} indicates the relative slippage of the wheel modules i and j. The variable e_i is the measure of the relative slippage of one wheel with respect to the other three wheels. In practice, for the mobile robot to move properly, all e_is must be small. The value of e_is determine the overall consistency of wheel i with respect to the other modules. Several cases can be considered for different values of e_i.

- In the case where:

  $$ \min(e_i) > e_{max} \tag{5} $$

  this means that the legs are somewhat faulty and that they are not consistent with one another. Observing such a condition for a pre-specified interval of time should cause the controller to issue a fault signal to stop the robot.
- If e_i is large only for one wheel, it can be interpreted such that the wheel is not consistent with the other wheels and that it is being pushed by them. In this case, different strategies can be considered, such as stopping the robot or singling out the faulty leg from the control loop and setting the actuators of the faulty wheel to rotate freely.
- Alternatively, more complex strategies can be designed based on the robot's assigned task and application.

After calculating e_i for all the legs and evaluating their values, the forward kinematics problem can be solved as follows. Let ${}^{\boldsymbol{B}}\hat{\boldsymbol{v}}$ and ${}^{\boldsymbol{B}}\hat{\boldsymbol{v}}_i$ be denoted by their components by $[v_x\ v_y]^T$ and $[v_{ix}, v_{iy}]^T$. Next, (2b) for all the legs can be rearranged into the following equation:

$$ \boldsymbol{A}\begin{bmatrix} vv_x \\ vv_y \\ \omega_B \end{bmatrix}_{3\times 1} = \boldsymbol{B} \tag{6} $$

in which:

$$ \boldsymbol{A}^T = \begin{bmatrix} 1 & 0 & \ldots & 1 & 0 \\ 0 & 1 & \ldots & 0 & 1 \\ -l_{1y} & l_{1x} & \ldots & -l_{4y} & l_{4x} \end{bmatrix}_{3\times 8} \tag{7a} $$

$$ \boldsymbol{B}^T = \begin{bmatrix} v_{1x}\ v_{1y}\ \ldots\ v_{4x}\ v_{4y} \end{bmatrix}_{1\times 8} \tag{7b} $$

Obviously, in the general case the matrix equation (6) consists of eight equations and three unknowns. A simple least squares solution is given by:

$$\begin{bmatrix} vv_x \\ vv_y \\ \omega_B \end{bmatrix}_{3\times 1} = \boldsymbol{A}^{\dagger}\boldsymbol{B} \tag{8}$$

Note that the elements of $\boldsymbol{A}$ are the only function of the constant vectors ${}^{B}\boldsymbol{\ell}_i$. Hence, $\boldsymbol{A}^{\dagger} = (\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{A}^T$ needs to be calculated once. In practice, we need at least three rows of matrices A and B to solve the forward kinematics. Thus, one way to improve the solution is to eliminate the inconsistent wheel module and eliminate the corresponding rows from (6). Our strategy is based on isolating the leg with a maximum of e_i. Next, the rows with respect to that leg are eliminated from matrix A in 7a and the least squares is calculated based on this new matrix. In [18], we show the efficacy of the above strategy and further details of the implementation of this approach.

4. Robot Software Modules

4.1. Dead Reckoning

Generally speaking, the time integration of the velocity vector – known as 'deductive reckoning' or *Dead Reckoning* (DR) – for a mobile platform gives its approximate position with respect to its initial pose [19, chap. 20]:

$${}^{\mathcal{I}}\boldsymbol{x}_{\mathcal{B}}(t) = {}^{\mathcal{I}}\boldsymbol{x}_{\mathcal{B}_0} + {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}_0}\int_{t_0}^{t} {}^{\mathcal{B}_0}\boldsymbol{R}_{\mathcal{B}}\,\boldsymbol{V}_x(\mathbf{t})\,\mathrm{d}\mathbf{t} \tag{9}$$

$${}^{\mathcal{I}}\boldsymbol{y}_{\mathcal{B}}(t) = {}^{\mathcal{I}}\boldsymbol{y}_{\mathcal{B}_0} + {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}_0}\int_{t_0}^{t} {}^{\mathcal{B}_0}\boldsymbol{R}_{\mathcal{B}}\,\boldsymbol{V}_y(\mathbf{t})\,\mathrm{d}\mathbf{t} \tag{10}$$

$${}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}(t) = {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}_0}\,\boldsymbol{R}_z(\theta_{\mathcal{B}}(t)),\ \theta_{\mathcal{B}}(t) = \left(\int_{t_0}^{t}\boldsymbol{\omega}_z(\mathbf{t})\,\mathrm{d}\mathbf{t}\right) \tag{11}$$

The driving velocities of the wheels as regards their steering angles and robot kinematics can generate fundamental data for the localization of the mobile platform.

4.2. SLAM

The errors for the mobile manipulator are accumulative during its motion. In order to reduce the errors, it is necessary to compare the position of the mobile base with respect to its environment. The SLAM [20] method provides the change in location of the robot according to its initial configuration. Moreover, it provides the local map of the environment ${}^{\mathcal{I}}\mathbb{M}'$, which is convertible to an occupancy grid for the planning module.

4.3. Path Planning

The path-planning module comprises two modes: local and global planning. The planner initially plans the required path through requests from ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}_0}$ to ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}f}$ as described in *Task 2*, namely a global plan. It considers a configuration space extracted from ${}^{\mathcal{I}}\mathbb{M}'$ based on the initial information and the foreknown structural conditions [21]. The global planner generates a set of via points, ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}via}$, which channelizes the route to approach ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}f}$.

The module also implements local planning. It receives the current status and some of the upcoming via points, together with the current local map ${}^{\mathcal{B}}\mathbb{M}$ imported from the SLAM module based on the LRF sensor. The outcome of the module is a segment of a smooth and obstacle-free path passing near the via points. It can also be extracted from a motion-planning roadmap framework [22, 23].

4.4. Path Following

A path follower controller navigates the robot on the given path and returns the resultant pose. The path-following module considers actuator limitations at maximum velocity and acceleration [24–26]. The path follower also avoids obstacle on the robot's route.

4.5. Mobile Manipulator Controller

Manipulators are usually provided together with their specialized controllers. However, coordinated tasks have to be performed synchronously between the path follower for the mobile base and the arm controller [27]. We use a mobile manipulator controller to close the outer control loop that follows the task commands distributed among the mobile base and the arm in a similar manner to [28].

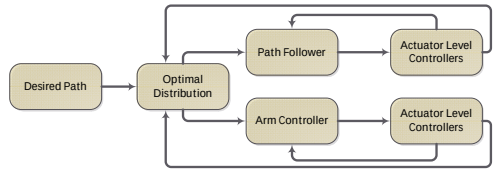


Figure 4. The loop for the synchronization of the mobile manipulation tasks.

4.6. Fault Management

Besides software fault detection and fault-tolerant techniques [29], it is necessary to monitor, verify and recover the robot's sensory data and actuator commands from the instrumentation and control viewpoint. This section contains a brief explanation of the fault management module's functionalities. As a general requirement for the fault tolerance of safety-critical systems [30, chap.1], the robot has redundancy in actuation and sensation. Four steerable wheels [31] reduce the risk of immobility caused by the mechanical hardware. The wheel odometry, IMU, laser scanner and camera provide redundancy for the perception of the robot. The fault management module's job includes:

- Monitoring the input and output of the other modules.
- In the case of a single point failure of a module, using an alternative function.
- Verifying the consistency of the resultant sensory information from the different sources, for each sensor as well as results of sensor fusion:

$$\Delta\boldsymbol{T} = {}^{\mathcal{B}}\boldsymbol{T}_{S_1}\left({}^{\mathcal{B}}\boldsymbol{T}_{S_2}\right)^{-1} \tag{12}$$

where S_1 and S_2 are the localization or detection data from Sources 1 and 2, respectively. For a perfect system, $\Delta\boldsymbol{T}$ becomes the identity 4×4 matrix. It should be

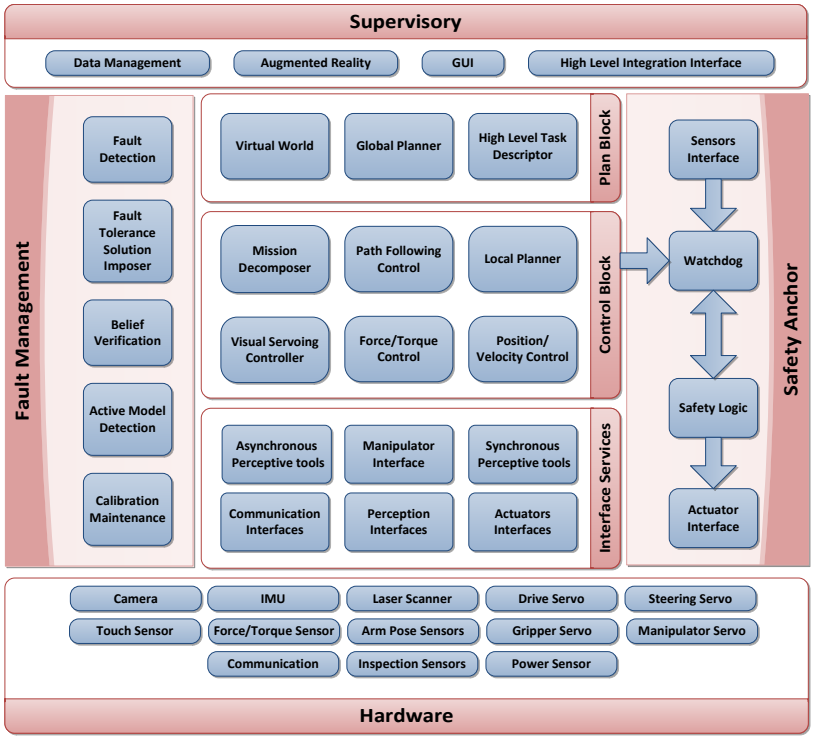


Figure 3. The robot's software and hardware modules.

noted that the inverse of a transformation matrix is calculated by the transpose of its rotational part as addressed in [28]. Therefore, this calculation cannot be singular. Arguments of $\Delta \boldsymbol{T}$ represent the inconsistency of the gathered information.

- If $\Delta \boldsymbol{T} - \boldsymbol{I}_{4\times4}$ gives a sensible difference, the module learns the difference.
- In case of a change in the learned difference, it calls the calibration maintenance module which in turn calibrates the divergent sensor.
- In case of unacceptable divergence of the primary navigation sensors, the module suspends the motion of the robot and reports its status. Following the confirmation of the operator, the control changes to tele-operation mode.

Some errors can be solved by outlier detection and the substitution of the fused values. For instance, in the following we describe how, at each segment of the path, the module compensates for small errors between the position and velocity feedback of the wheels.

The localization is obtained by dead-reckoning of the odometry data as described in the previous section. The *i*Moro mobile platform shown in Figure 1 is used to run the experiments. The desired path that is followed by the mobile platform is a cubic *Bezier* curve with its control points located at $\{(0,0),(2m,0),(2m,2m),(0,2m)\}$. The path is used with two desired heading profiles. Figure 5 shows the body and legs trajectories for following the path with 180° rotation of the main body. Moreover, Figure 6 shows the same trajectories, this time for −90° rotation of the body. Figure 5 also depicts the estimated location of the body's *Instantenous Center of Rotation* (ICR) during its movement as red dots.

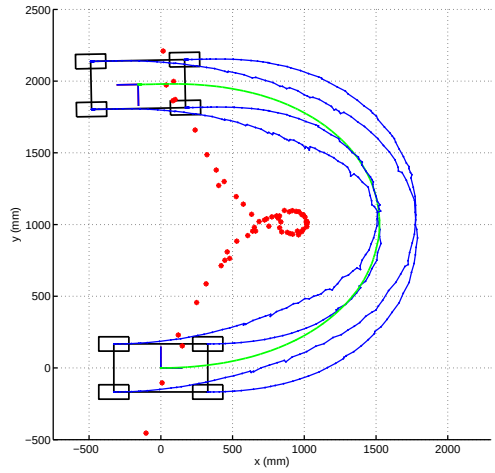


Figure 5. Experimental results: the robot's trajectories, 180° rotation.

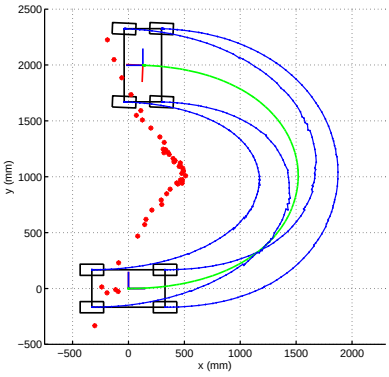


Figure 6. Experimental results: the robot's trajectories, -90° rotation.

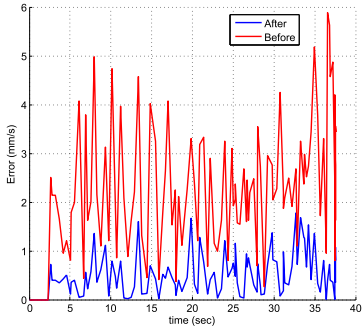


Figure 7. Experimental results: Least squares error before and after the elimination of the outlier leg.

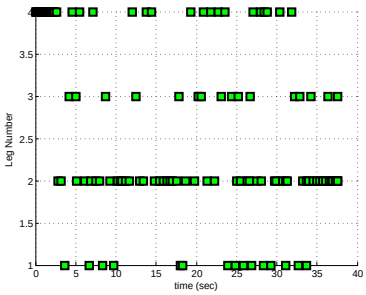


Figure 8. Experimental results: leg number for the most inconsistent leg.

The forward kinematics solution is solved both with and without eliminating the leg with maximum e_i. Figure 7 depicts the pseudo-inverse error before and after the elimination. From the figure, it is clear that the error is reduced considerably by eliminating the leg with maximum e_i from the forward kinematics solution.

Figure 8 shows the sensor number with the biggest e_i at each sample time. From the figure, it is clear that, at each period, a different leg has maximum e_i. This is due to the change in the dynamic loads acting on each leg during movement. Having one specific leg with maximum e_i all the time could indicate a defect in that leg.

5. Safety Anchor (SA)

It is a key requirement for a robot in SFs to be safe or at least compatible with the safety standards' requirements. The first solution is to make the whole system certified by the safety integrated standards, for instance, the *Safety Integrity Level* (SIL) II or SIL III standards. The solution is suitable for mechatronic systems with lower flexibility and mass-produced products due to its complexity and cost. On the other hand, the perception and navigation of the mobile platform include a variety of procedures and instruments which are not necessarily safe. For the software, certain test techniques exist for the verification of the safety standards' compatibility. However, there are many issues confronting the verification of decision-making for an autonomous robot, even for industrial controlled environments, if we assume that the robots are Automated Guided Vehicles (*AGVs*). Moreover, the compatibility of the processes and the required modules for decision-making cause an increase in the robotâĂŹs production costs and verification complexity. Furthermore, the robot must be equipped with safe instruments and safe process units, which is impossible for several sensors. For instance, vision-based estimations and acceleration measurements are not compatible with the current safety regulations.

In order to make the system tolerant to faults in these modules, we define a redundant controller. The controller is supposed to be safety compatible for both the hardware and software. The software must be reactive and simple, and therefore easy to examine. The hardware requirement for such software becomes suitable for implementation on a safe Programmable Logical Controller (PLC) or any other safe process unit. It should be noted that target application for this design comprises tunnels and corridor-shaped SFs. Locally, the watchdog on the robot, and remotely, the operator, can trigger the SA when the functionality of the robot is not as expected.

5.1. Hardware Architecture

A schematic hardware architecture of the mobile platform is shown in figure 9. The sensors and actuators are connected to the embedded PC, which runs all of the necessary cognition subsystems. The embedded PC is connected to the motor drives, arm joints, laser scanners, cameras, IMU and encoders. Most of the robot sensors provide the perception of the robot and its environment for object manipulation. The embedded PC is the target of the remote development method described in [32] during its run-time and debugging and the development of its algorithms. It should be noted that any failure in the embedded PC during its run-time can cause the immobility of the robot or a collision with other facilities in the environment.

In order to avoid failure, the authors propose adding a safety anchor topology as a redundant decision-maker for

the mobile platform, as shown in figure 9. During the robot's run-time, in case of any embedded PC failure, the safety PLC can be triggered by the watchdog or supervisory system to manage the mobile platform in escape mode by a reactive controller. Ignoring complex mathematical procedures and high-level algorithms, the PLC solves the navigation problem at the lowest possible level to guide the robot to safe places [33] before the operation of the accelerators. The redundant system can benefit from off-the-shelf, safe and certified components to guaranty the safety of the system.

The LRF sensors provide for the availability or occupancy of any predefined zone by toggling the corresponding signal. The signal goes through the logical circuits of the PLC. By limited conditional statements programmed on the PLC, the system reacts by commanding the motor drives in real-time. Since the logic, codes and connections are reactive, time-invariant and simple, their verification is easier and more dependable. In the same architecture, we can also program and utilize the embedded PC by means of the remote development method described in [32, 34].

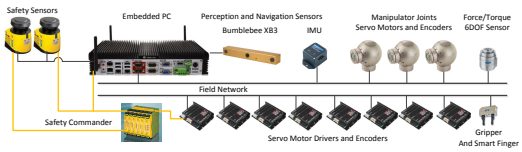


Figure 9. The hardware architecture related to cognition; the yellow parts are the proposed safe components.

5.2. SA Logic

The hardware described in 5.1 needs a special setup, adjustment and programming. The LRF zones need the warning and alarm zones to be set up similar to the zones shown in Figures 10, 11 and 12. The reason for using the LRFs is their flexibility in the definition of sensitive zones and their compatibility with the safety standards. Generally speaking, the pseudo-code for the PLC can show the reactions between the sensors and actuators in Table 1. It should be noted that the system design is reactive. However, for the clarity of the commands' order, it is shown as a sequence of commands.

```
1 Initialize the sensors.
2 Disconnect the actuators from the embedded PC.
3 Stop the wheels.
4 Align the steering.
5 Read LRF output.
6 Adjust the drive speeds by visibility of obstacles.
7 Steer the robot.
8 When robot arrived, finalize the movement.
9 Return to 5.
```

Table 1. The pseudo-code for the PLC.

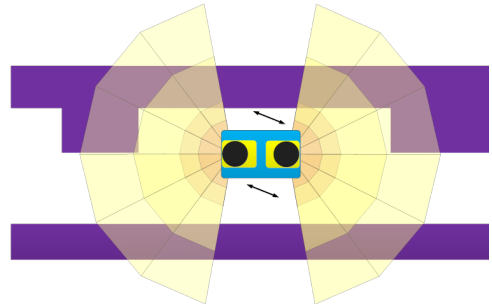


Figure 10. LRF zones, crab steering.

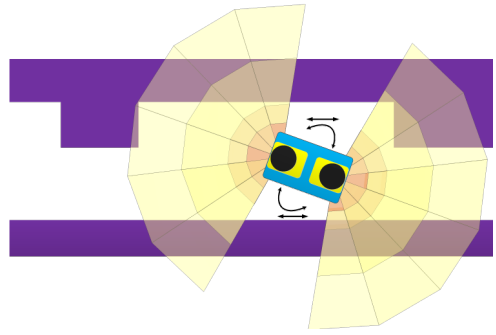


Figure 11. LRF zones, robot spins around.

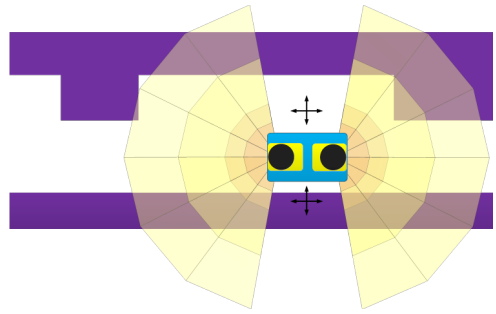


Figure 12. LRF zones, cross movements.

Furthermore, in the solution, we consider the conditions in SFs. For instance, it is common that the experiment's infrastructures are located at one side of a tunnel or a corridor - as they are at CERN LHC or GSI/FAIR - and the transportation corridor usually runs along beside them in parallel or circular patterns. Therefore, within the path-selection routines, we give higher priority to following the simpler wall of the corridor. According to Table 1, let us divide the navigation into two parts: the determination of the robot's velocity (driving), and steering. In the upcoming paragraphs, we describe the overall idea and the benefits of a 4WS mobile robot for this purpose.

The robot's steering capabilities, as described in Section 3, include a vast area of selection and calculation so as to determine it for each wheel of the robot [24, 35]. However, a robust solution, a dependable implementation and a reliable verification method for such calculations become complicated, especially if a safe solution is required. Recall that the robot's immobility is a high-severity failure. For the purpose of simplification, we assume that the robot has only two types of steering: crab steering and spin around to keep the detection [36], planning[37], kinematics, and control [26] simpler than contemporary solutions.

5.2.1. Crab Steering

Crab steering means that the angular velocity ω_B, shown in Figure 2 and mentioned in (2b), becomes approximately zero. In other words, the instantaneous center of rotation (ICR) goes to infinity. As such, (2b) becomes:

$$v_i \, {}^{\boldsymbol{B}}\hat{\boldsymbol{v}}_i = v \, {}^{\boldsymbol{B}}\hat{\boldsymbol{v}} \tag{13}$$

This means that the velocity vectors of the body and legs become parallel (or equal) to each other. In such a case, it is not necessary to solve the system of kinematic equations of the robot and the direction of ${}^{\boldsymbol{B}}\hat{\boldsymbol{v}}$; body's velocity vector defines the wheels' steering angles ${}^{\boldsymbol{B}}\hat{\boldsymbol{v}}_i$. Any change in the steering comes from the change in the body's movement direction (not the heading). In this case, the robot can continue with high-speed linear motion up to a sensible change in the velocity direction regarding the actuator limitations.

Crab steering activates when the robot is approximately parallel to the simpler wall of the corridor while the robot has obstacle in front as shown in Figure 10. The crab steering maintains the distance from the side obstacles or walls without lowering speed. The magnitude of velocity vector v is proportional to the proximity of the obstacles in front of the vehicle. It should be noted that, instead of a distance measurement, a corresponding number for the zone is considered.

5.2.2. Spin Around

There exists another method which leads to a significant simplification of the complicated path-follower kinematics. The 4WS robot can turn around on the spot without any linear movement, or negligible translational movements. In this case, the ICR converges into the centroid of the robot. In this case, another term of the velocity analysis becomes zero. The kinematics relations in (2b) become:

$$v_i \, {}^{\boldsymbol{B}}\hat{\boldsymbol{v}}_i = \omega_B(\hat{z} \times {}^{\boldsymbol{B}}\boldsymbol{\ell}_i) \tag{14}$$

which shows the constant angles of the wheels perpendicular to ${}^{\boldsymbol{B}}\boldsymbol{\ell}_i$. Therefore, the change rate of the heading defines the driving velocity of each wheel v_i while $v\,\hat{\boldsymbol{v}}$ - the linear velocity of the body - becomes zero.

6. Conclusion

In this paper, we study the architecture of a mobile platform. The overall architecture, including a variety of software modules, is studied in detail, and is suitable for risk assessment and RAMS analysis. Moreover, a top-down strategy is described to show how a high-level mission can be broken down into several tasks as well as which hardware/software are required to accomplish the mission. In order to consider safety concerns in Scientific Facilities (SFs), we propose "Safety Anchor", a redundant decision-making system that can ensure that the robot leaves any hazardous environment without the presented cognition architecture. Moreover, Safety Anchor guarantees meeting the challenging limitations of the actuators by solving the kinematics of the robot in its simplified form. The specifications and design requirements of a four wheel steerable mobile manipulator, the IHA Mobile Robot (*iMoro*), are considered as a case study in this paper.

7. Acknowledgements

This work, supported by the European Union's Seventh Framework Program under the Marie Curie Initial Training Network, was carried out within the framework of the PURESAFE, *Preventing hUman intervention for incREased SAfety in inFrastructures Emitting ionizing radiation*, under REA grant agreement number 264336.

8. References

[1] SP Sharma, Dinesh Kumar, and Ajay Kumar. Reliability analysis of complex multi-robotic system using ga and fuzzy methodology. *Applied Soft Computing*, 12(1):405–415, 2012.

[2] Thomas Fabry, Liesbeth Vanherpe, Bruno Feral, and Christian Braesch. Developing an interactive intervention planner-a systems engineering perspective. *International Journal of Advanced Robotic Systems*, 10(337), 2013.

[3] Ramviyas Parasuraman, Prithvi Pagala, Keith Kershaw, and Manuel Ferre. Energy management module for mobile robots in hostile environments. In *Advances in Autonomous Robotics*, pages 430–431. Springer, 2012.

[4] M. M. Aref, R. Oftadeh, J. Koivumäki, R. Ghabcheloo, and J. Mattila. System requirement document for modular mobile manipulator system. *PURESAFE System Engineering Documents*, WP3(RP10-RP14), 2012.

[5] M. Kleinehagenbrock, J. Fritsch, and G. Sagerer. Supporting advanced interaction capabilities on a mobile robot with a flexible control system. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 3649–3655. IEEE, 2004.

[6] N. Palomeras, A. El-Fakdi, M. Carreras, and P. Ridao. COLA2: A control architecture for AUVs. *IEEE Journal of Oceanic Engineering*, 37(4):695–716, 2012.

[7] Kanna Rajan, Frédéric Py, and Javier Barreiro. Towards deliberative control in marine robotics. *Marine Robot Autonomy*, pages 91–175, 2013.

[8] Christian Rauch, Tim Köhler, Martin Schröer, Elmar Berghöfer, and Frank Kirchner12. A concept of a reliable three-layer behaviour control system for cooperative autonomous robots. *35th German*

Conference on Artificial Intelligence, September 24-27 2012.

[9] R. Vasudevan, V. Shia, Yiqi Gao, R. Cervera-Navarro, R. Bajcsy, and F. Borrelli. Safe semi-autonomous control with enhanced driver modeling. *American Control Conference (ACC)*, pages 2896–2903, 2012.

[10] W. Chung, G. Kim, and M. Kim. Development of the multi-functional indoor service robot psr systems. *Autonomous Robots*, 22(1):1–17, 2007.

[11] H. Rajaie, O. Zweigle, K. Häussermann, U.P. Käppeler, A. Tamke, and P. Levi. Hardware design and distributed embedded control architecture of a mobile soccer robot. *Mechatronics*, 21(2):455–468, 2011.

[12] Dongeun Choi, Junho Choi, Changhyun Cho, Shinsuk Park, and Sungchul Kang. A safe robot arm with safe joints and gravity compensator. *International Journal of Control, Automation and Systems*, 11(2):362–368, 2013.

[13] Mathieu Baudin, Pierre Bonnal, and Jean-Michel Ruiz. Combining activity DSM with temporal logic for collaborative planning and scheduling. *International Journal of Advanced Robotic Systems*, 10(337), 2013.

[14] A. Lecchini, A. Lanzon, and B.D.O. Anderson. A model reference approach to safe controller changes in iterative identification and control. *Automatica*, 42(2):193 – 203, 2006.

[15] Craig Glennie. Rigorous 3d error analysis of kinematic scanning lidar systems. *Journal of Applied Geodesy*, 1(3):147, 2007.

[16] Guoqiang Fu, Paolo Corradi, Arianna Menciassi, and Paolo Dario. An integrated triangulation laser scanner for obstacle detection of miniature mobile robots in indoor environment. *IEEE/ASME Transactions on Mechatronics*, 16(4):778–783, 2011.

[17] S.W. Yang and C.C. Wang. Dealing with laser scanner failure: Mirrors and windows. In *IEEE International Conference on Robotics and Automation, ICRA*, pages 3009–3015. IEEE, 2008.

[18] Reza Oftadeh, Mohammad M. Aref, Reza Ghabcheloo, and Jouni Mattila. Mechatronic design of a four wheel steering mobile robot with fault-tolerant odometry feedback. *IFAC Mechatronic Systems*, 1(1):663–669, 2013.

[19] Bruno Siciliano and Oussama Khatib, editors. *Springer handbook of robotics*. Springer, 2008.

[20] A. Tamjidi, H.D. Taghirad, and A. Aghamohammadi. On the consistency of EKF-SLAM: Focusing on the observation models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2083–2088. IEEE, 2009.

[21] J. Choi, R.E. Curry, and G.H. Elkaim. Continuous curvature path generation based on bezier curves for autonomous vehicles. *IAENG International Journal of Applied Mathematics*, 40(2):91–101, 2010.

[22] A. Agha-mohammadi, S. Chakravorty, and N.M. Amato. FIRM: Feedback controller-based information-state roadmap a framework for motion planning under uncertainty. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4284–4291. IEEE, 2011.

[23] Ali-akbar Agha-mohammadi, Suman Chakravorty, and Nancy Amato. Firm: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *International Journal of Robotics Research (IJRR)*, 33(2):268–304, 2014.

[24] R. Oftadeh, R. Ghabcheloo, and J. Mattila. A novel time optimal path following controller with bounded velocities for mobile robots with independently steerable wheels. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Japan,2013.

[25] Filippo Arrichiello, Stefano Chiaverini, Giovanni Indiveri, and Paola Pedone. The null-space-based behavioral control for mobile robots with velocity actuator saturations. *The International Journal of Robotics Research*, 29(10):1317–1337, 2010.

[26] Reza Oftadeh, Reza Ghabcheloo, and Jouni Mattila. Time optimal path following with bounded velocities and accelerations for mobile robots with independently steerable wheels. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2925–2931, 2014.

[27] Mohammad M. Aref, Reza Ghabcheloo, Antti Kolu, M. Hyvönen, K. Huhtala, and Jouni Mattila. Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine. *IEEE International Conference on Robotics, Automation and Mechatronics (RAM), Manila, Philippine*, pages 218–224, 2013.

[28] Mohammad M. Aref, Reza Ghabcheloo, and Jouni Mattila. A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine. *Accepted for IEEE International Conference on Robotics and Automation(ICRA), Hong Kong*, pages 6816–6822, 2014.

[29] Pekka Alho and Jouni Mattila. Software fault detection and recovery in critical real-time systems: An approach based on loose coupling. *Fusion Engineering and Design*, 2014.

[30] Guillaume JJ Ducard. *Fault-tolerant flight control and guidance systems: Practical methods for small unmanned aerial vehicles*. Springer, 2009.

[31] Chih-Jui Lin, Su-Ming Hsiao, Ying-Hao Wang, Cheng-Hao Yeh, Chien-Feng Huang, and Tzuu-Hseng S Li. Design and implementation of a 4WS4WD mobile robot and its control applications. In *System Science and Engineering (ICSSE), 2013 International Conference on*, pages 235–240. IEEE, 2013.

[32] R. Oftadeh, M. M. Aref, R. Ghabcheloo, and J. Mattila. Unified framework for rapid prototyping of Linux based real-time controllers with Matlab and Simulink. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 274–279, Taiwan, 2012.

[33] K. Kershaw, C. Bertone, P. Bestmann, T. Feniet, D. Forkel-Wirth, JL Grenard, and N. Rousset. Remotely operated train for inspection and measurement in CERN LHC tunnel. *Particle Accelerator Conference*, pages 2902–2904, 2009.

[34] R. Oftadeh, M. M. Aref, R. Ghabcheloo, and J. Mattila. Real-time system integration for mobile manipulation. *International Journal of Advanced Robotic Systems*, 11(15), 2013.

[35] Reza Oftadeh, Mohammad M. Aref, Reza Ghabcheloo, and Jouni Mattila. Bounded-velocity motion control of four wheel steered mobile robots. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Australia*, pages 255–260, 2013.

[36] Zahra Ziaei, Reza Oftadeh, and Jouni Mattila. Global path planning with obstacle avoidance for omnidirectional mobile robot using overhead camera. *IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 697–704, 2014.

[37] Ali-akbar Agha-mohammadi, Saurav Agarwal, Aditya Mahadevan, Suman Chakravorty, Daniel Tomkins, Jory Denny, and Nancy M Amato. Robust online belief space planning in changing environments: Application to physical mobile robots. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 149–156, 2014.

Publication IV

M. Aref, M., Ghabcheloo, R., Kolu, A., and Mattila, J., "A multistage controller with smooth switching for autonomous pallet picking," *IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden*, pp. 2535 – 2542, 2016

A Multistage Controller with Smooth Switching for Autonomous Pallet Picking

Mohammad M. Aref, Reza Ghabcheloo, Antti Kolu, and Jouni Mattila

***Abstract*— This paper addresses the problem of pallet picking by an *Articulated–Frame–Steering* (AFS) hydraulic machine. We propose a macro–micro visual mobile manipulation architecture, where a smooth switching logic navigates the robot to pick an object. The state space is divided into several regions depending on the accuracy of the vision and robot's degrees of freedom. The control architecture benefits from the following phenomena: at distance, when the location of the object of interest is detected, its orientation may not be reliably estimated; at some closer distances, orientations also become available; and because pallets are wide with small height, yaw angle estimation are more accurate than pitch is. The switching logic is devised to control the corresponding degree of freedom of the mobile manipulator in each region. Moreover, in different regions, we employ different coordinate frames, namely an earth–fixed frame or an object–local frame, which is more natural for the problem in that region. We show that the architecture accomplishes the following: 1) it eliminates the need for replanning as the accuracy of pose estimation improves; and 2) it provides the mobile base with a longer corridor to steer toward the pallet and align its heading. We also incorporate a robust, accurate solution based on fiducial markers for object manipulation in unstructured outdoor environments and unfavorable weather conditions, which relies solely on a monocular camera for pallet detection. The presented experimental results demonstrate the superiority of the method, as the model starts following the target even when the pallet is still $6m$ away from the vehicle.**

I. Introduction

Autonomous Pallet Picking (APP) as a control system encompasses three main elements, as follows: sensing, control, and mechanical actuation. Forklift machinery, particularly in outdoor applications, usually has a low actuation number. The logistics industry is price conscious, and except in rare case, an increase in degrees of freedom or maneuverability is not favored. This gives rise to several challenges ranging from designing an APP motion controller for nonholonomic mobile platforms with under-actuated manipulators to object recognition and pallet detection methods.

For vision-based controllers, error in the target detection and limitations of the mechanical system are both common sources of error [1]. A proper controller should overcome both obstacles improving the overall performance. In the literature, many studies focus on one of these issues over the other. A group of studies gives more weight to the pallet detection problem. Pallet detection and the integration of a visual sensor in the control topology is the main focus in [2], whereas [3] aims to correct the data while approaching the pallet. Most of these works employ at least two cameras, a monocular vision system with defined CAD models [4], or both a laser scanner and camera [5].

Authors are with the Department of Intelligent Hydraulics and Automation, Tampere University of Technology, 33101, Finland.
Email: m.aref@ieee.org .

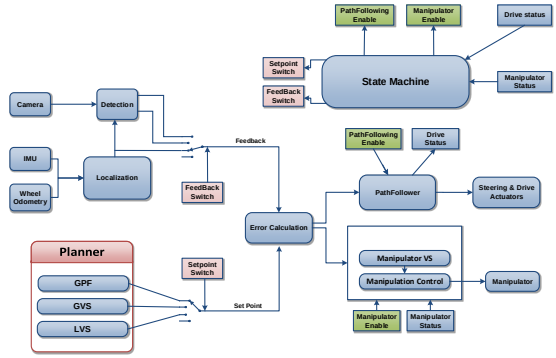


Fig. 1. Control topology for Multistage Macro-Micro Visual Servoing

Typically, the maximal range of the vision detection methods become 3 to $4m$ in front of the vehicle for known targets. For instance, the reported maximal detection ranges are as follows:

- $3m$ for the CAD–based method in [4];
- $4m$ for the double–sensor architecture in [5];
- $3.7m$ for mobile camera–space manipulation *(MCSM)* with stereo vision [6], [7];
- $3.5m$ for the authors' previous work [8]; and
- $2m$ for different sized pallets using a laser scanner [9].

Only research including multiple-view laser scanners provides longer distances for the forklift configuration space, specifically $6m$ [10], [11]. Therefore, regardless of the object detection method, suitable sensory information is available only in the close vicinity of the object. For large articulated–frame–steering machines with reduced mobility, the approach angle must already be corrected from larger distances to avoid the need for extra back and forth maneuvers.

Our method for APP is capable of detection and control of the pallet picking from $6m$ with only a monocular vision system. The method improves the macro–micro controller [8] via a step–by–step switching among the control modes and path segments to incorporate marker detection data to the extent of its validity. Its immediate result is a broadened configuration space with higher impact and a greater application area. The method does not impose any limitations on the detection methods. Therefore, it is capable of integrating

any vision, time–of–flight, or range sensors that are suitable for object detection.

There have been multiple controllers proposed for object manipulation and pallet picking in the literature. Some of them navigate the robot in the image space or camera coordinate, for instance, the MCSM method [12], [12], [7], [13], [14]. The intrinsic complexity, nonlinear dynamics, and nonholonomic constraints of such machines make the motion control problem a difficult task [15], [16], [17], [18]. It requires requires the application of path planning and following algorithms for APP [19],[11], [20]. The docking problem also involves similar challenges [21]. One of the fundamental issues for these categories of APP studies is the definition of the operational coordinate, which becomes a reference for the desired path, errors, and target pose [10]. The operational coordinate of APP has to be either a global or a local coordinate. Each of these options has benefits and disadvantages in relation to the other.

In the case of employing a global coordinate, integration to the normal autonomous navigation modules of the vehicle is easier. However, any update in the desired path requires the re-execution of the global path planner, since the improvements in the target pose estimation occur when the robot approaches to target. In contrast, operation using a local coordinate makes it easier to provide high–frequency location and detection feedback. However, this changes the navigation control architecture and may lead to undesirable drifts in the position estimation.

The method proposed here takes advantage of both operational coordinates. During the time that the robot follows a desired path and detects the object for the first time, our method plans a smooth path before switching into the local operation coordinate. Therefore, the robot's motion avoids jumps and drifts during navigation in both the global and local coordinates, as well as during the switching intervals. Another step change benefit from this method is the ability to start the *Visual Servoing* (VS) from farther distances; this provides more room and therefore smoother steering of the nonholonomic mobile base. In most scenarios, this feature eliminates the need to align the forks by driving backward and replanning the path.

In addition to the aforementioned improvements, the proposed method complies with constraints of the industrial development environment and hardware, such as inaccurate actuators, minimal tolerance compared to the actuators' resolutions, a rough environment, and so on. This paper is organized as follows: Section II, describes the overall problem definition and defines the coordinates and the control variables. Section III provides an overview of the vision feedback and necessary estimation. Next, in Section IV, after introducing the VS controller, the main idea of the proposed *Multistage Micro–Mirco Visual Servoing* (M3VS) controller is described in subsection IV-B. The rest of the section provides more details on the control and system integration of M3VS. Finally, in Section V, the overall results of the pallet picking experiments and effects of switching show the efficacy of the proposed method in action. The

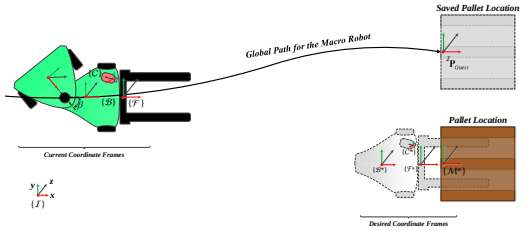


Fig. 2. Defined coordinate frames, path following on the global path, Step1.

results exhibit significant improvement of APP compared to other contemporary methods with only a monocular vision sensor while dealing with inaccurate and accurate detection information by the M3VS control schema.

The experiments are based on GIM Machine, which is one of the autonomous vehicles built at the Department of Intelligent Hydraulics and Automation (IHA) at Tampere University of Technology, Finland based on Avant 635. The experiments also examined through the tests on GIM Sim, a detailed Hardware-in-the-loop simulator developed in the department for design, control, and mapping simulations of hydraulic mobile machines [22] and multi-vehicle tests [23].

II. Problem Definition

APP occurs after long-distance transportation on a given path, $P_{d_{GPF}}$. A path-follower controller navigates the robot on the path receiving localization feedback, for instance, from the *Global Positioning System* (GPS), inertial sensors, or LIDAR-based *Simultaneous Localization and Mapping* (SLAM) [24], with respect to an inertial frame $\{\mathcal{I}\}$. For outdoor GPS-enabled locations, we can attach $\{\mathcal{I}\}$ to a fixed flat–Earth coordinate.

The rotation matrix ${}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}$, parametrized by the Euler angles ϕ(roll), θ(pitch), and ψ(yaw), describes the orientation of the front unit, and ${}^{\mathcal{I}}\boldsymbol{P}_{\mathcal{B}} = (x, y, z)$ is the position of the body origin in the inertial frame. Let $\{\mathcal{P}\}$ and $\{\mathcal{F}\}$ be the coordinate frames attached to the pallet and the fork, respectively. As shown in Figure 3, these frames are arranged such that the control objective is to have those frames coincide, and therefore ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{P}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{F}}$.

This objective is achieved with the machine's three–degrees–of–freedom manipulator, as described below. The boom rotates around the y–axis of the body by the angle θ_1, its telescopic joint stretches in/out by the length L, and the revolute joint at the end of the boom rotates the fork around the y–axis by the angle θ_2. Thus the fork frame is defined by the following:

$$ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}} = \boldsymbol{T}(L, \theta_1, \theta_2) \tag{1} $$

with respect to the body frame. Let u_1, u_L, and u_2 be the manipulation control signals for θ_1, L, θ_2, and let u_s and u_v be the driving control signals for the machine's steering and speed commands, respectively. The control objective is to devise control laws for these five control signals such that

$$ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{F}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(x, y, z, \phi, \theta, \psi)\ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}(L, \theta_1, \theta_2) \tag{2} $$

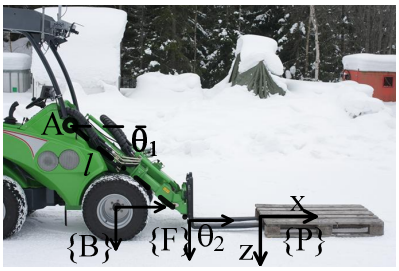


Fig. 3. The micro robot's coordinate frames.

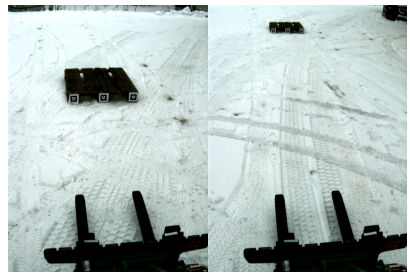


Fig. 4. Undistort camera images: A pallet appears in camera's FOV (Right); Vision toolbox receives accurate estimation of the pallet's orientation (Left).

coincides with ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{P}}$. Note that the roll motion ϕ (rotation around the x–axis of the pallet) is not controllable. Therefore, we assume that the ground, pallet, and machine all have the same roll angles. Corresponding to the above control signals, we project the error space onto two orthogonal planes:

1) $x-y$ plane error variables (x, y, ψ) are controlled by drive $(u_s,\, u_v)$; and
2) $x-z$ plane error variables (x, z, θ) are controlled by the manipulator $(u_1,\, u_L,\, u_2)$.

There is a redundancy in the control space in that x can be controlled by both the drive and the manipulator. The accuracy of the position control by the machine speed is limited. Therefore, u_v is used to drive the fork near the pallet, and the manipulator is used if necessary to reduce the remaining residual error to zero (usually less than 10 cm).

III. Object Detection

To pick up an object, it is a key concept that the controller has to receive an accurate pose as a target with respect to its control point on the robot body. In the M3VS controller, it is even more critical to make a precise estimation of the object because, at the final stage, the imaginary local coordinate attaches to the pallet pose. The *Vision Toolbox*(VT) provides the pose of object with respect to the camera, as described in Section III-A. The constant transformation matrix to define the pose of the effective camera frame in the body is calculated through the calibration method described in [8]. To provide better insight into the visibility and feasibility of object detection, in Section III-B, we conduct a test to present changes in VT transmitted data versus the robot's position.

A. Vision

To observe the $\{\mathcal{P}\}$ frame, a set of markers is installed on the face of the pallet pallet, as shown in Figure 3. By using a camera, the onboard pose of the markers and thus that of the pallet, that is, ${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{P}}$, with respect to the camera frame $\{\mathcal{C}\}$ is determined. Therefore, the pallet's position and orientation in the body frame can be resolved if the calibration parameter ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}$ is known:

$$ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{P}} = {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}\ {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{P}}. \tag{3} $$

This defines the visual feedback. For image processing, we have used the Alvar toolbox [25], an open-source augmented reality library. This can be used to track fiducial markers based on the known shape and size. It outputs the position and the orientation of the marker relative to the camera. This toolbox is capable of automatically tuning distortion and intrinsic parameters for a pin–hole camera by detecting a predefined checkerboard marker from 50 different poses. A comparison of the calibration results shows that the calibration of intrinsic parameters under indoor illumination conditions (before installation on the vehicle) provides better results. Other necessary measurements are performed using the wheel encoders and inertial sensors. Two samples of the camera images are shown in Figure 4

Equation 3 provides the output of the vision system at time t, denotes as ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}(t-T_d)$. The synchronization of the dead-reckoning and vision is performed by storing a series of $[x, y, \phi, \theta, \psi](k) : k = t, t-T_s, t-2T_s, \ldots,$ where $T_s = 50\ ms$. The pallet does not move, and therefore ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$ is constant. Each time the vision detects the object, we calculate ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$ using:

$$ {}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}} = {}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}(t-T_d)\ \ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{I}}(t-T_d). \tag{4} $$

In the case that ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{I}}$ at time $t-T_d$ is not yet in the stored set, a simple linear interpolation is performed to evaluate ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{I}}(t-T_d)$. Then, $[x, y, \phi, \theta, \psi](t-T_d)$ is derived by linearly interpolating between the two stored dead-reckoning data of the closest time instances. The main issue is that the ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$ given by equation (4) is noisy. Averaging over several instances has been shown to improve and considerably smoothen the estimation. Let ${}^{\mathcal{P}}\hat{\boldsymbol{T}}_{\mathcal{I}}$ denote the average of n latest measurement of ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$ using (4). If the markers are found at least n times, the pallet tracking filter is initiated and VS is started. To recover ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}(t)$ between consecutive instances of marker detection, we can calculate

$$ {}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{B}}(t) = {}^{\mathcal{P}}\hat{\boldsymbol{T}}_{\mathcal{I}}\ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(t) \tag{5} $$

as fast as dead–reckoning is updated. This will provide real–time, fault–tolerant feedback to the control system. Using the fact that ${}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$ is constant, before averaging n measurements, we eliminate those that are too far from the rest (using k–means clustering). Common faults, such as marker detection

failure and error accumulation of localization, are partially canceled using this method.

B. Integration of Visual Information

According to our previous studies on sensor fusion and pose estimation of predefined fiducial markers [8], [26], for the early stages of detection, outputs of the detection toolbox are rejected as outliers due to their inconsistency with the IMU and odometery data. This is done to increase robustness and improve the overall control performance. However, further investigation on the transmitted outputs show that the transmitted raw data are capable of providing more information.

Even if the pose detected by the VT is not consistent with the localization information, transmitting *any* result provides that the VT recognized relevant markers in its field of view. Of course, in longer distances, larger variances of the estimated pose is expected. Further comparison between the so-called "filtered outliers" and the accepted detected poses shows significant similarities among the positions but not the orientations. Recall that the final stage of the proposed macro–micro controller switches into the local coordinate attached to the estimated pallet pose, ${}^{\mathcal{I}}\hat{\boldsymbol{T}}_{\mathcal{P}}$. Therefore, inaccurate data was rejected because of the sensitivity of the overall performance to the pallet orientation, ϕ. The method can be considered robust and reliable where the robot is about to engage because of keeping the camera output improvements in the loop. However, VS at farther distances does not require high–frequency updates in the global coordinate. Recall that any major change in the path control points necessitates replanning. To alleviate the severity of these effects, we use the M3VS method detailed in the following section.

IV. Visual Servoing Controller

A. Visual Servoing Preliminaries

Let $\boldsymbol{S}_i (i = 1,\ 2,\ 3, \ldots, k)$, defined by

$$\boldsymbol{S}_i = [X_i, Y_i, Z_i]^T,\ \boldsymbol{S}_i^* = [X_i^*, Y_i^*, Z_i^*]^T \tag{6}$$

be the feature points describing three-dimensional coordinate systems $\{\mathcal{M}\}$ and $\{\mathcal{M}^*\}$, which in turn define the current and desired object poses ${}^{\{\mathcal{C}\}}\boldsymbol{M}$ and ${}^{\{\mathcal{C}\}}\boldsymbol{M}^*$ with respect to the current camera frame. The pixels on the image that corresponds to those features may be expressed by

$$\boldsymbol{s}_i = [1, u_i, v_i]^T,\ \boldsymbol{s}_i^* = [1, u_i^*, v_i^*]^T \tag{7}$$

Based on a pinhole model for the camera, relations between the matching points become

$$\boldsymbol{s}_i = \frac{1}{X_i}\boldsymbol{A}\boldsymbol{S}_i\ ,\ \boldsymbol{s}_i^* = \frac{1}{X_i^*}\boldsymbol{A}\boldsymbol{S}_i^* \tag{8}$$

in which $\boldsymbol{A}_{3\times 3}$ is the camera matrix evaluated by intrinsic parameters calibration. We determine the scale of normalized positions in equation 8 by incorporating the geometrical properties of the object that are known a priori. Therefore, instead of comparing the $\boldsymbol{S}_i^*$ and $\boldsymbol{S}_i$ points we use $\{\mathcal{M}^*\}$ and $\{\mathcal{M}\}$ hereafter, which is equivalent to the employment of the matched $\boldsymbol{s}_i^*$ and $\boldsymbol{s}_i$ features with the depth or scale information. Note that we disregard the accumulation of uncertainties along the image depth axis, which lies on the z axis of the camera frame $\{\mathcal{C}\}$ as shown in Figure 3. Moreover, installation of a camera on the forklift manipulator is practically unviable. Therefore, we emulate a virtual camera attached to the manipulator by transforming the output of the camera $\{\mathcal{C}\}$ attached on the vehicle's roof into the coordinate frame $\{\mathcal{F}\}$ attached to the manipulator. In addition, if the camera were to be installed at the place of $\{\mathcal{F}\}$, the image Jacobian could become near singular at some points in picking the pallet.

Geometrically, $\{\mathcal{M}\}$ coincides with $\{\mathcal{M}^*\}$ if and only if the current virtual camera $\{\mathcal{F}\}$ coincides with $\{\mathcal{F}^*\}$. In other words, from the vision viewpoint, we are mapping the VS problem because the pose of ${}^{\mathcal{I}}\boldsymbol{C}$ is suitable for the object detection ($\dot{\boldsymbol{s}}$), while ${}^{\mathcal{I}}\boldsymbol{F}$ is suitable for the movement detection ($\dot{\boldsymbol{v}}$). According to the above explanations, the error tensor for the VS becomes

$$\boldsymbol{T}_e = {}^{\mathcal{F}}\boldsymbol{T}_{\mathcal{F}^*} \tag{9}$$

The control objective is to drive $\boldsymbol{T}_e$ to identity.

B. A Multistage Controller for Visual Servoing

The concept of M3VS for pallet picking is comparable with an everyday life task. Assume a person has been asked to pick a known object (e.g., a book) with the following instructions. "Pick book (X). It is on the robotics bookshelf on the left side in the library." The person can interpret the sentence to carry out step-by-step actions, as follows:

1) Based on the memorized place, move toward the location;
2) Look for the bookshelf and approximately approach the area that looks like a robotics bookshelf;
3) Find the book and stand next to it. Align your body to make enough room for your arm to move dexterously without collision;
4) Pick the book by hand from the shelf.

Similar procedures are applicable to the pallet picking problem. We can list the robot's main actions through the following corresponding steps:

1) *Path-following:* A stationary global task planner plans a set of paths based on the status of all robots and objects in the configuration space, tasking the robot with pallet picking. The path is defined in the global coordinate, namely *Global Path Follower* (GPF) path. It assumes that the pallet is located at ${}^{\mathcal{I}}\boldsymbol{P}_{Guess}$. Since the location is not accurate, the task planner triggers a flag assigned for VS. The robot follows the given path while its vision sensor scans for the corresponding marker of the pallet;

2) *Macro robot VS in the global frame:* When the vision sensor detects the object, the robot compares two consequent positions of the object with the robot's

motion. If the results are below the error threshold, it switches into another global path considering the new estimation of ${}^{\mathcal{I}}\boldsymbol{P}_{Guess}$ called the *Global Visual Servoing* (GVS) path. Note that this detection happens at farther distances, and the calculated orientations of the objects, and therefore ${}^{\mathcal{I}}\hat{\boldsymbol{T}}_{\mathcal{P}}$, are not acceptable and they exhibit high variance;

3) *Macro robot VS in the local frame*: While following the updated path, as soon as the detected object pose ${}^{\mathcal{I}}\hat{\boldsymbol{T}}_{\mathcal{P}}$ becomes consistent with the robot movements, it switches to the local coordinate. A predefined path for picking and placement is defined for each pallet according to its geometry and environmental limitations, called the *Local Visual Servoing* (LVS) path. The origin of the local coordinate becomes the target pose for picking and the end point of the path for the macro robot. Exploiting this method allows the latest updates to be incorporated smoothly without any bouncing or sudden change in the controller and its outputs. Thanks to the previous step, the robot can wait until the estimation of ${}^{\mathcal{I}}\hat{\boldsymbol{T}}_{\mathcal{P}}$ continuously improves while approaching. Therefore, the change in coordinate frames become smooth and robust to the jumps in detection or localization sensory information. This operation mode works until the pallet enters the workspace of the manipulator; and

4) *Micro robot VS:* After arrival in the desired vicinity, the robot has enough information to run a classic *Position–Based Visual Servoing* (PBVS) for its planar manipulator for pallet engagement.

Pseudo–code for the segment of a path with a VS flag:

---

while $\|\boldsymbol{T}_e - \mathbb{I}\| > \varepsilon_{Micro}$ **do**
  $i \leftarrow 1$
  if $\hat{\boldsymbol{P}}_p \in$Admissible Region **then**
    $i \leftarrow 2$
    if $\{\hat{\boldsymbol{R}}_p, \hat{\boldsymbol{P}}_p\} \in$Admissible Region **then**
      $i \leftarrow 3$
      run Feasibility Check
      if $\|\boldsymbol{T}_{e_1} - \mathbb{I}\| < \varepsilon_{Macro}$ **then**
        $i \leftarrow 4$
        run MVS, $\{\mathcal{F}\} \rightarrow \{\mathcal{F}^*\}$
      end if
    end if
  end if
  if $i \in \{1, 2, 3\}$ **then**
    run Estimation("`Update`"),${}^{\mathcal{P}}\hat{\boldsymbol{T}}_{\mathcal{I}} \rightarrow {}^{\mathcal{P}}\boldsymbol{T}_{\mathcal{I}}$
    run Path-Follower(CP_i), $\{\mathcal{B}\} \rightarrow \{\mathcal{B}^*\}$
  else
    run Path-Follower("`Stop`")
    run Planner("`Update`")
  end if
end while

Let the macro robot become the vehicle itself, without consideration of the manipulator boom or fork that carries the actual camera coordinate $\{\mathcal{C}\}$. The manipulator part is considered to be the micro robot to which the virtual camera is connected.

According to the definitions in Sections II and IV-A, we divide the problem of pallet picking into a VS problem for the macro and micro robots. For that purpose, we expand the transformations applicable to the coordinates such that:

$$\boldsymbol{T}_e = {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}^{-1}\ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{B}^*}\ {}^{\mathcal{B}^*}\boldsymbol{T}_{\mathcal{F}^*} \tag{10}$$

in which ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}^{-1}$ and ${}^{\mathcal{B}^*}\boldsymbol{T}_{\mathcal{F}^*}$ determine the required movements for the micro robot, and the remaining ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{B}^*}$ governs the demanded motion of the macro robot. This expansion leads to the new definition of two consistent errors for both macro and micro robots:

$$\boldsymbol{T}_{e_1} = {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{B}^*} \tag{11}$$

$$\boldsymbol{T}_{e_2} = {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{F}}^{-1}\ {}^{\mathcal{B}^*}\boldsymbol{T}_{\mathcal{F}^*} \tag{12}$$

Mathematically speaking, it should be noted that (11) and (12) are *not* the direct results of (10), unless the macro robot arrives at its destination, $\{\mathcal{B}\} \longrightarrow \{\mathcal{B}^*\}$. We disregard this difference to simplify the macro and micro controllers formulations independently. However, it cannot cause any problems because, before the arrival to the pallet, the principle axes of the coordinates for the macro and micro robots always become approximately parallel to each other. Recall that the pallet is assumed to be the same angle around the x–axis of the body (roll motion). This is the only direction in which the robot exhibits a lack of actuation.

C. Path Definitions

To achieve the goal of APP in a modular system, we want to minimize modification of the other software modules, which have perfect functionality in the robot's navigation. Therefore, we impose the M3VS concepts on the boundary conditions of the *path Control Points* (CP) as described below. The state machine switches setpoints and paths according to the VS steps.

The GPF path comprises sets of n CP positions and velocities needed for well-behaved, smoothly connected Bezier curves of the order $n-1$ generated from the global path planner based on obstacle-free path planning methods [27]. The points are represented with respect to $\{\mathcal{I}\}$. Each segment of the path starts from the correspondent ${}^{\mathcal{I}}\boldsymbol{P}_1$ and finishes by ${}^{\mathcal{I}}\boldsymbol{P}_n$. The last GPF path that the planner transmits to the robot tasks the robot with APP by triggering the VS flag. The state machine turns on the marker detection toolbox and waits for the result of having the pallet in the field of view. The point P_n path has to coincide with ${}^{\mathcal{I}}\boldsymbol{P}_{Guess}$. For the second–last CP, we have:

$${}^{\mathcal{I}}P_{n-1} = {}^{\mathcal{I}}P_n - k_1\hat{\boldsymbol{x}}_{P_{Guess}} \tag{13}$$

The path for GVS consists of n CPs in the global frame. The pose where the robot detects the approximate position

of the pallet is represented by ${}^{\mathcal{I}}\boldsymbol{P}_{GVS}$. For the CPs for this step, we have:

$$ {}^{\mathcal{I}}P_1 = {}^{\mathcal{I}}\boldsymbol{P}_{GVS} \tag{14} $$

$$ {}^{\mathcal{I}}P_2 = {}^{\mathcal{I}}P_1 + k_2\hat{\boldsymbol{x}}_{P_{GVS}} \tag{15} $$

$$ {}^{\mathcal{I}}P_{n-1} = \frac{1}{k_3+k_4}\left(k_3\,{}^{\mathcal{I}}P_1 + k_4\,{}^{\mathcal{I}}\hat{P}_P\right) \tag{16} $$

$$ {}^{\mathcal{I}}P_n = {}^{\mathcal{I}}\hat{P}_P \tag{17} $$

The path for LVS is represented in the pallet coordinate when the robot acquires an acceptable value for ${}^{\mathcal{P}}\hat{\boldsymbol{T}}_{\mathcal{B}}$. The pose of the body, at the moment that the position and orientation of the pallet is detected correctly, can be represented by ${}^{\mathcal{P}}\boldsymbol{P}_{LVS}$. The CPs for the last piece of the path become

$$ {}^{\mathcal{P}}P_1 = {}^{\mathcal{P}}\boldsymbol{P}_{LVS} \tag{18} $$

$$ {}^{\mathcal{P}}P_2 = {}^{\mathcal{P}}P_1 + k_5\hat{\boldsymbol{x}}_{P_{LVS}} \tag{19} $$

$$ {}^{\mathcal{P}}P_n = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{20} $$

$$ {}^{\mathcal{P}}P_{n-1} = {}^{\mathcal{P}}P_n + k_6\hat{\boldsymbol{x}}_P \tag{21} $$

Values for $k_i, i \in \{1, 2, \dots\}$ are positive real constant coefficients that are adjustable depending on the smoothness tolerance of the system.

D. Motion Control of the Macro Robot

The aim of VS for the macro robot is to guide the body frame $\{\mathcal{B}\}$ to reach the desired coordinate $\{\mathcal{B}^*\}$. Its reflection can be found in variation of 11 within its $x-y$ plane, as well as the rotation around its $z-$axis. Therefore, the transformation matrix for the macro robot could be written as ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(x, y, \psi)$ because the other parameters for the location of the macro robot are not controllable. Owing to the nonholonomic constraints of the machine, the control in the $x-y$ plane takes a rather more complex structure. The macro robot's autonomous driving architecture within the global $\{\mathcal{I}\}$ is valid until it receives a task command that contains a VS flag. It navigates to a vicinity of the given location ${}^{\mathcal{I}}\boldsymbol{B}_{Guess}[x, y, \psi]$ defined in the inertial global frame $\{\mathcal{I}\}$. At the same time, by activating the VT, it looks for the given mark of the object. Since the camera's field of view (FOV) is reasonably wider than the accuracy of the initial guess, we can be sure of finding the pallet. After detection of the pallet, the robot replies to the higher level with an update to the initial guess. Again, the planner transmits a new path. In contrast with the last path, the most recent one is in the local coordinate $\{\mathcal{B}^*\}$.

A smooth and obstacle-free path $\boldsymbol{p}_d$ is generated that extends from the current location ${}^{\mathcal{B}^*}\boldsymbol{P}_{LVS}[x, y, \psi]$ of the macro robot to the origin of $\{\mathcal{B}^*\}$, as illustrated in 2. Since the path is presented within $\{\mathcal{B}^*\}$ itself, ${}^{\mathcal{B}^*}\boldsymbol{B}^*$ becomes $[0, 0, 0]$. It should be noted that ${}^{\mathcal{B}^*}\boldsymbol{P}_{LVS}$ is the first position at which the object has been detected five consecutive times. The start point of the path is fixed in the global coordinate, and the end point is considered as a relative coordinate to the origin of $\{\mathcal{B}^*\}$. Therefore, any change in the estimated pose of the pallet *will not* cause a time–consuming update of the path by the planner. At the same time, any enhancement in the pose estimation affects the goal point of the path. This change leads to an increment in the process time efficiency for real-time VS tasks because a request for AI–based path planning takes up plenty of CPU time depending on the complexity of the planner configuration space [28]. From the ${}^{\mathcal{B}^*}\boldsymbol{P}_{LVS}$ point forward, the macro robot follows the motion control signals to zero the path–following errors to minimize $\boldsymbol{T}_{e_1}^{-1}$ error.

Now, we define the error ${}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{B}}$, where $\{\mathcal{T}\}$ is the path tangent frame attached to $\boldsymbol{p}_d(\tau)$ with $\tau \in [0, 1]$ being the path parameter. $\{\mathcal{T}\}$ coincides with $\{\mathcal{B}^*\}$ for $\tau = 1$. The path following errors can then be defined as:

$$ \begin{aligned} e_x &= [1\ 0\ 0\ 1]\ {}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{B}} \\ e_\psi &= \arg({}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{B}}, \hat{\boldsymbol{z}}) = \arg({}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{F}}, \hat{\boldsymbol{z}}) \\ e_y &= [0\ 1\ 0\ 1]\ {}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{B}} = [0\ 1\ 0\ 1]\ {}^{\mathcal{T}}\boldsymbol{T}_{\mathcal{F}} \end{aligned} \tag{22} $$

in which, the $\arg(.,.)$ function returns the angle argument of the rotation part of the transformation matrix of the first argument around the axis determined by the second argument (the z-axis in this case).

The yaw and y-axis errors are equal in the body and in the fork frames. Thus, if the machine closely follows the path, and the pose estimates are accurate enough, by the time the fork frame reaches the pallet, the yaw and y-axis errors will be adequately small, which makes the rotation part of equation 10 equal to 11 and 12. Evidently, as the machine gets closer to the pallet, the pose estimations become more accurate. Therefore, even if the output trajectory of the body frame does not initially coincide with $\boldsymbol{p}_d$, as the pose estimation errors decrease, it converges to $\boldsymbol{p}_d$. Although it is possible to design the macro robot controller as an eye-in-hand VS problem, according to [29], following the planned path is more robust. Particularly, this is advantageous in a multi-robot and multi-machine environment, where the macro robot's path should be coordinated with the other machines.

We showed in [30] that the horizontal plane suitably approximates the kinematic model of the GIM machine front unit as

$$ \begin{aligned} \dot{x} &= v_x \cos\psi \\ \dot{y} &= v_x \sin\psi \\ \dot{\psi} &= \omega_z \end{aligned} \tag{23} $$

$$ \omega_z = \frac{l_r\dot{\beta} + v_x \sin\beta}{l_f \cos\beta + l_r}, \tag{24} $$

where β is the body articulation angle and l_f, l_r are geometrical parameters (see Figure 5).

Definition: Path following problem. Based on the above kinematic model, design a control law for steering signal u_s to drive e_ψ and e_y to zero. The solution in [30] provides a control law for the desired angular velocity $\omega_z^c = C_{PF}(e_x, e_y, e_\psi)$), where the speed v_x is an exogenous input and may be set separately. Freedom in assigning the linear speed is instrumental in our case; it enables us to slow

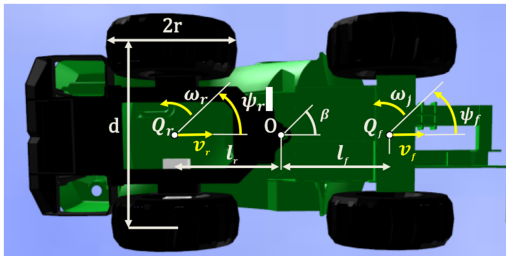


Fig. 5. GIM mobile machine with two body units pivoting around O.

down or stop the machine without causing instability in the path-following controller. For example, if the manipulator errors are not yet small enough, collision with the pallet can possibly be avoided. Given ω_z^c, we use (24) to calculate the required steering valve command:

$$\begin{aligned} \dot{\beta}^c &= -\frac{v_f}{l_r}\sin\beta + \left(\frac{l_f}{l_r}\cos\beta + 1\right)\omega_f^c \\ u_s &= K\dot{\beta}^c. \end{aligned} \tag{25}$$

In the above, we assumed that the valve command is proportional to the speed of body articulation.

E. Visual Servoing for the Micro Robot

The macro robot is a hydrostatic rough terrain forklift, which loses its repeatability for motions that require actuation accuracy of less than 20 cm. The path follower controller is capable of driving the macro robot to the vicinity of the pallet with a relatively small error, $\{\mathcal{B}\} \longrightarrow \{\mathcal{B}^*\}$. For the rest of the motion, we employ the manipulator to compensate for the macro robot errors in the ${}^{\mathcal{B}}\boldsymbol{x}$ direction and to adjust the height and angle of the forks (z and θ). As shown in Figure 3, there are three joint parameters for the micro robot with the following kinematics:

$$\begin{aligned} \dot{\boldsymbol{q}} &= \boldsymbol{J}(\boldsymbol{q})\boldsymbol{K}\boldsymbol{v} \\ \begin{bmatrix} \dot{\theta}_1 \\ \dot{L} \\ \dot{\theta}_2 \end{bmatrix} &= \begin{bmatrix} k_1 L\sin\theta_1 & k_2 L\cos\theta_2 & 0 \\ -k_1\cos\theta_1 & k_2\sin\theta_2 & 0 \\ 0 & 0 & -k_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_x \\ \boldsymbol{v}_z \\ \boldsymbol{\omega}_y \end{bmatrix} \end{aligned} \tag{26}$$

Here, $\boldsymbol{K}$ stands for the coefficients of conversion between the oil debit of hydraulic proportional valves, their displacements, and the movements of the joints. Based on the above model, we define the errors in the *x-z plane* as

$$\begin{aligned} e_\phi &= \arg({}^{\mathcal{F}^*}\boldsymbol{T}_{\mathcal{F}}, \hat{\boldsymbol{y}}) \\ e_z &= [0\ 0\ 1\ 1]\ {}^{\mathcal{F}^*}\boldsymbol{T}_{\mathcal{F}} \\ e_x &= [1\ 0\ 0\ 1]\ {}^{\mathcal{F}^*}\boldsymbol{T}_{\mathcal{F}} \end{aligned} \tag{27}$$

Recall that if the errors of the macro robot are small enough, they may be transformed as corresponding components of the error tensor of the micro robot expressed in equation 12, and can be presented as an error among the matched features introduced in equation 6 transformed into the virtual camera coordinate:

$$\boldsymbol{e}_{vs} = {}^{\mathcal{F}}\boldsymbol{T}_{\mathcal{C}}\left({}^{\mathcal{C}}\boldsymbol{S}_i - {}^{\mathcal{C}}\boldsymbol{S}_i^*\right) \tag{28}$$

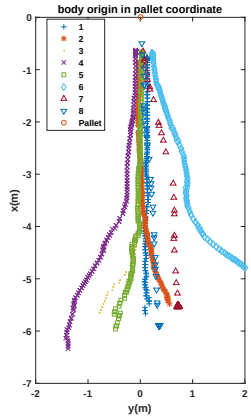


Fig. 6. Tracking filter output. Eight representative tests.

Substituting the PBVS control law, as addressed in [31] and [29], and to enforce an exponential decoupled decrease of the error, we apply $\dot{\boldsymbol{e}_{vs}} = -\lambda \boldsymbol{e}_{vs}$ to derive:

$$\dot{\boldsymbol{q}} = -\lambda \boldsymbol{J}(\boldsymbol{q})\ \boldsymbol{K}\ \boldsymbol{e}_{vs} \tag{29}$$

The final step of M3VS is achieved by applying the above control law to the micro part.

V. Experiments

As a result of GPS errors or any unsaved small changes in the pose of a pallet, we examine the effectiveness of the proposed method by artificial error insertions in the predefined pallet pose, as shown in Figure 6. According to the test results, the method is capable of detection and correction of the errors in the given pallet position up to $5-6$ m. Similar to the authors previous work [8], in distances shorter than 3.5 m, the estimated orientation of the pallet becomes accurate enough for VS in the local coordinate. The steering limitations are no longer a problem compared to those of the other methods. Because of the longer range detection and earlier heading correction, the nonholonomic mobile base has enough room for the rejection of lateral and heading errors. However, the error in the y direction must be smaller than $2m$ to keep the pallet in the camera's FOV.

Figure 6 shows the trajectory of the body frame in the pallet frame in the $x-y$ pallet plane. For clarity of presentation, we show only few representative tests starting from the left, middle, and right side of the pallet. The attached video shows functionality of the controller in clear weather as well as snowy and slippery environment.

VI. Conclusion

The proposed method takes advantage of the both global and local operational coordinates. During the time that the robot follows a desired path and detects the object for the first time, our method plans a smooth path before switching into the local operation coordinate. Therefore, the robot motion avoids jumps and drifts during the global and local

visual servoing (VS). Another step change benefit from this method is the ability to start VS from farther distances, which provides more room and therefore smoother steering of the nonholonomic mobile base. This feature, in most scenarios, eliminates the need to align the forks by driving backward and replanning the path. The method strives to simplify system integration for complex modular control software without altering the high-level planner(s) or low-level controllers.

VII. Acknowledgment

The authors gratefully acknowledge the important technical support of Mobile Hydraulics Laboratory staff at Tampere University of Technology during the field tests and simulations. This work is supported by the Academy of Finland under the project "Integrated Multimodal Sensing of 3D Environment for Intelligent Manipulators", grant no. 286260.

References

[1] J.-D. Yoder, J. West, E. Baumgartner, M. Perrollaz, M. Seelinger, and M. Robinson, "Experiments comparing precision of stereo-vision approaches for control of an industrial manipulator," in *Experimental Robotics*, pp. 245–256, Springer International Publishing, 2013.

[2] L. F. Holeva, E. R. Elston, M. J. Seelinger, and J.-D. S. Yoder, "Identifying and selecting objects that may correspond to pallets in an image scene," May 5, 2015. US Patent 9,025,886.

[3] K. Ban, F. Warashina, M. Yamada, and Y. Namiki, "Robot system comprising visual sensor," Dec. 4, 2012. US Patent 8,326,460.

[4] S. Byun and M. Kim, "Real-time positioning and orienting of pallets based on monocular vision," in *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, vol. 2, pp. 505–508, IEEE, 2008.

[5] L. Baglivo, N. Biasi, F. Biral, N. Bellomo, E. Bertolazzi, M. Da Lio, and M. De Cecco, "Autonomous pallet localization and picking for industrial forklifts: a robust range and look method," *Measurement Science and Technology*, vol. 22, no. 8, p. 085502, 2011.

[6] M. Seelinger and J.-D. Yoder, "Automatic visual guidance of a forklift engaging a pallet," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 1026–1038, 2006.

[7] A. Cárdenas, B. Goodwine, S. Skaar, and M. Seelinger, "Vision-based control of a mobile base and on-board arm," *International Journal of Robotics Research*, vol. 22, no. 9, pp. 677–698, 2003.

[8] M. M. Aref, R. Ghabcheloo, and J. Mattila, "A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong*, pp. 6816–6822, 2014.

[9] D. Lecking, O. Wulf, and B. Wagner, "Variable pallet pick-up for automatic guided vehicles in industrial environments," in *Emerging Technologies and Factory Automation, 2006. ETFA'06. IEEE Conference on*, pp. 1169–1174, IEEE, 2006.

[10] M. R. Walter, M. Antone, E. Chuangsuwanich, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, Y. Friedman, J. Glass, J. P. How, *et al.*, "A situationally aware voice-commandable robotic forklift working alongside people in unstructured outdoor environments," *Journal of Field Robotics*, vol. 32, no. 4, pp. 590–628, 2015.

[11] M. R. Walter, S. Karaman, E. Frazzoli, and S. Teller, "Closed-loop pallet manipulation in unstructured environments," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 5119–5126, IEEE, 2010.

[12] M. J. Seelinger, J.-D. S. Yoder, and S. B. Skaar, "Mobile camera-space manipulation," Feb. 27, 2001. US Patent 6,194,860.

[13] M. Seelinger and J.-D. Yoder, "Automatic pallet engagment by a vision guided forklift," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 4068–4073, IEEE, 2005.

[14] M. Seelinger, J.-D. Yoder, E. T. Baumgartner, and S. B. Skaar, "High-precision visual control of mobile manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 6, pp. 957–965, 2002.

[15] A. Donaire, J. G. Romero, T. Perez, and R. Ortega, "Smooth stabilisation of nonholonomic robots subject to disturbances," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 4385–4390, IEEE, 2015.

[16] R. Oftadeh, M. M. Aref, R. Ghabcheloo, and J. Mattila, "Bounded-velocity motion control of four wheel steered mobile robots," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Australia*, pp. 255–260, 2013.

[17] R. Oftadeh, R. Ghabcheloo, and J. Mattila, "A time-optimal bounded velocity path-following controller for generic wheeled mobile robots," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 676–683, IEEE, 2015.

[18] R. Oftadeh, R. Ghabcheloo, and J. Mattila, "A novel time optimal path following controller with bounded velocities for mobile robots with independently steerable wheels," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Japan,2013.

[19] T. A. Tamba, B. Hong, and K.-S. Hong, "A path following control of an unmanned autonomous forklift," *International Journal of Control, Automation and Systems*, vol. 7, no. 1, pp. 113–122, 2009.

[20] S. Teller, M. R. Walter, M. Antone, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, J. Glass, J. P. How, A. S. Huang, *et al.*, "A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 526–533, IEEE, 2010.

[21] J. Villagra and D. Herrero-Pérez, "A comparison of control techniques for robust docking maneuvers of an AGV," *Control Systems Technology, IEEE Transactions on*, vol. 20, no. 4, pp. 1116–1123, 2012.

[22] T. Krogerus, M. Hyvönen, P. Multanen, J.-P. Hietala, R. Ghabcheloo, and K. Huhtala, "Joint probability distributions of correlation coefficients in the diagnostics of mobile work machines," *Mechatronics*, 2016.

[23] M. Hyvönen, M. Rajala, A. Virtanen, J. Jankkari, K. Huhtala, and R. Ritala, "Assistive situation awareness system for mobile multimachine work environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3403–3413, 2015.

[24] A. Aghamohammadi, A. Tamjidi, and H. Taghirad, "SLAM using single laser range finder," *Proceedings of 17th World Congress of the International Federation of Automatic Control (IFAC)*, 2008.

[25] VTT Technical Research Centre of Finland, "Alvar toolkit website." http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/index.html, 2016.

[26] M. M. Aref, R. Ghabcheloo, A. Kolu, M. Hyvönen, K. Huhtala, and J. Mattila, "Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics, Automation and Mechatronics (RAM), Manila, Philippine*, pp. 218–224, 2013.

[27] J.-W. Choi and K. Huhtala, "Constrained global path optimization for articulated steering vehicles," *Vehicular Technology, IEEE Transactions on*, no. 99, 2015.

[28] A. Agha-Mohammadi, S. Chakravorty, and N. Amato, "FIRM: Feedback controller-based information-state roadmap a framework for motion planning under uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4284–4291, IEEE, 2011.

[29] F. Chaumette and S. Hutchinson, "Visual servo control. II. advanced approaches [tutorial]," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[30] R. Ghabcheloo and M. Hyvönen, "Modeling and motion control of an articulated-frame-steering hydraulic mobile machine," *Proceedings of the 17th Mediterranean Conference on Control and Automation*, Greece, June 2009.

[31] G. Silveira and E. Malis, "Real-time visual tracking under arbitrary illumination changes," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–6, IEEE, 2007.

Publication V

M. Aref, M., Ghabcheloo, R., and Mattila, J., "Real-time vision-based navigation for nonholonomic mobile robots," *IEEE International Conference on Automation Science and Engineering (CASE)*, 2016

Real-Time Vision-Based Navigation for Nonholonomic Mobile Robots

Mohammad M. Aref[1], Reza Oftadeh[2], Reza Ghabcheloo[1], and Jouni Mattila[1]

***Abstract*— Mobile manipulators are one of the major pillars in the thriving field of service robotics, perhaps mainly due to the wide workspace provided by the mobile base. Exploitation of this vast workspace entails accurate positioning of the robot while overcoming several complexities, such as nonholonomic constraints and actuator limitations. In this paper, we analyze the integration of a vision and path-following controller for the accurate localization and tracking of known objects. Due to the broad workspace of mobile manipulators, the improved repeatability of the mobile base positioning can significantly improve overall system performance, especially near the robot's workstation. Our method provides an accurate synchronization and real-time fusion of data that together with the closed-form solution of the given path-following controller manages to accurately navigate the robot to perform a gripping task. The achieved repeatability of the system is below 3 cm, which is demonstrated by experimental results.**

I. Introduction

Mobile manipulation is a highly active branch of robotic research that strives to design and develop software and hardware modules pivotal in various practical applications, such as inspection and manipulation in hazardous environments.From the perspective of low-level software modules, two categories of control and sensing play a major role. The control part is responsible for execution of reliably fast and accurate movements abiding by the mechanism's and actuators' limitations, while the sensors provide the necessary sensing information to the controller.

Generally speaking, the time-critical aspect of control blocks demands real-time criteria and software environments for their execution. In contrast, the detection and sensing algorithms are iterative with high process costs, regardless of the type of the actual sensor that provides the crude data. This study mainly focuses on enhancing the integration and synchronization of the two categories, culminating in precise and fast positioning of wheeled mobile robots (WMR). The algorithm receives asynchronous data from an object detection toolbox (ODT) that provides the motion controller with real-time feedback. During this process, especially for vision-based controllers, error in target detection and limitations of the mechanical system are both common sources of error [1]. Therefore, a proper controller should overcome both obstacles while improving the overall performance.

Comprehensive studies have been carried out on vision-based sensing for mobile robots in [2] and later in [3] or for navigation of underwater robots [4]. However, precision and repeatability have been overlooked in most of these. Moreover, there have been many researches on image-based navigation of mobile robots, especially for long-distance motions [5], [6]. The downside of the image-based visual servoing control schema is that it affects the other control blocks, such as nonlinear actuator-level controllers, which brings up new challenges for system integration. In some cases, such as an autonomous hydraulic mobile machine [7], the system benefits from a complex control module for low-level actuators that tackles the inherent nonlinearities.

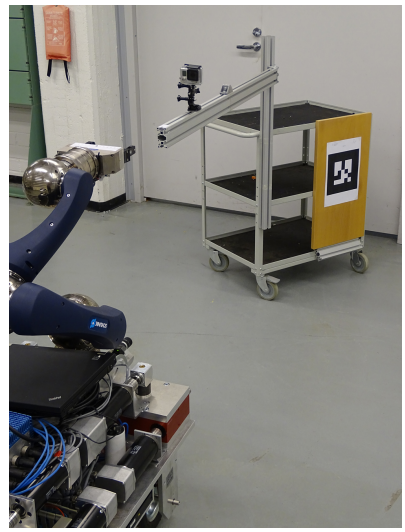


Fig. 1. iMoro mobile robot and the experiment setup.

The target mobile manipulator in this study, called iMoro, is a four-wheel-steerable (4WS) mobile manipulator. The mobile base has 8 servo motors for steering and driving its wheels [8] and is one example of such robots; it does not fit into the common unicycle model [9]. Therefore, we use position-based methods to benefit from both accurate visual tracking and integration with state-of-the-art pre-existing control modules.

A. Related Works

In recent years, accuracy and repeatability of localization and positioning methods of mobile robots, for the purpose of mobile manipulation and service robotics have culminated

[1] The Department of Intelligent Hydraulics and Automation, Tampere University of Technology, 33101, Finland. e-mail: m.aref@ieee.org

[2] The Department of Computer Science and Engineering (CSE), Texas A&M University, TX 77840, United States.

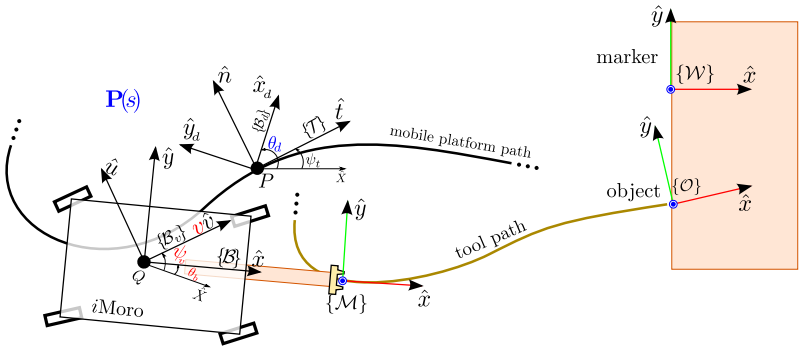


Fig. 2. Defined coordinate frames for the robot, path, and target.

in several interesting studies for indoor applications. For instance, [10] analyzed the repeatability of a laser-based localization and positioning system for a holonomic omni-directional robot, KUKA omniRob platform.

Another recent study [11] integrates localization information in the trajectory planning phase, and the closed loop controller uses odometry feedback for the navigation of KUKA omniRob. In contrast, in this paper, our work benefits from a closed-form solution that accounts for actuator limitations in the controller rather than the planner. Therefore, we integrate global feedback into the closed loop with higher frequencies and as soon as the visual detection toolbox provides the results.

It is usually desirable for many applications to rely on more accurate detection methods, which normally increase the process cost [10]. Although the applications among the aforementioned studies are slightly different, the proposed vision-based controller potentially improves the performance of robot positioning in the vicinity of its workstation. The target application of this research is the accurate control of the moving platforms of mobile manipulators near their workstations for pick and place tasks. Furthermore, we show that this approach improves docking functionalities of Automated-Guided Vehicles (AGV)s. As shown in Table I, the proposed method is capable of improving the results of relevant studies in the field.

Compared to contemporary research, the main contributions of this paper are twofold: precise synchronization and fusion of data from different sources, and taking into account actuators' limits by a path-following controller. Moreover, free of any specific detection algorithm or sensor, the proposed method is capable of fusing localization and tracking data with the high-maneuverability path-following control of 4WS mobile robots. It should be noted that the mobile base of iMoro is a popular mechanism that is closely similar to Rollin' Justin from DLR [14], PR2 from Willow Garage [15], and Care-o-bot from Fraunhofer IPA [16].

This paper is organized as follows: Section II explains the formulation of the overall problem and defines the relevant coordinate frames and the control variables. In II-A we address the solution concepts. Section III provides an overview of the vision feedback and detailed information about the data synchronization. Section IV explains the velocity-bounded path-following controller. Finally, in Section V, the experimental results of the proposed method are shown in action. This paper also has a video attachment, including one of the test scenarios and an examination of the system response for a large disturbance. Further videos of the experimental evaluations are available on YouTube [1].

TABLE I

RESEARCHES ON ACCURATE MOBILE ROBOT NAVIGATION

Ref*	Local Update	Global Update	Positioning Max Acc.	Sensor	Robot
[11]	35 Hz	1.5 Hz	4 cm	Range	omniRob
[10]			6 cm	2xLaser	omniRob
[12]			8 cm	Vision	Pioneer3AT
[13]	50 Hz	5-7 Hz	10 cm	Vision &IMU	Avant Forklift
Current	200 Hz	7-10 Hz	3 cm	Vision	iMoro

* The addressed papers cover a wide range of functionalities. Only the parts relevant to our work are considered in the table. The applications and environments are not exactly similar.Empty cells indicate no information.

II. PROBLEM DEFINITION

For high-performance mobile manipulation tasks, an accurate and responsive closed-loop control of the mobile base is a key requirement. Maneuverability degrees of a planar mobile base can coincide with 1 to 3 Degrees Of Freedom (DOF) and provide redundancy with the first joints of a manipulator. A wheeled mobile robot is known to reach broader workspace with less accuracy in positioning. The aim of this study is to improve the accuracy of the mobile base positioning in the vicinity of a target. It uses fused vision and wheel odometry data to provide feedback for a path-following controller proposed in [9], [17]. This system controls the mobile robot going toward the closest possible

[1] https://www.youtube.com/channel/UCxemrk8NrIj-db6h06VBVLA

vicinity of the target object to pick it up. This architecture considers the following objectives and requirements:

1) The controller feedback has to be real-time and available at each minor sample time of the system, which is $0.005s$. In that interval, all the servo motors derives should receive a synchronized steering or driving command.
2) Overall performance of the system should maintain accuracy and repeatability of the system less than $\pm 3cm$ about each axes. Therefore, the mobile manipulator can catch a fixed object between its gripper fingers without the manipulator motion.
3) Accuracy of the target tracking should be less than $\pm 5mm$. This accuracy makes the feedback convenient for manipulation purposes, too.
4) The system should work with a pre-existing ODT, which can process only 5–10 image frames of per second. The camera and process power are supposed to be low-cost general purpose components.
5) The mobile base of the robot is nonholonomic, and the control should consider kinematic constraints for its navigation and motion control.
6) The actuators of the mobile base are servo motors with bounded velocity and acceleration capabilities.

We define a problem formulation that suits a general test case for the fulfillment of the aforementioned necessities. Assume that position and orientation of the object is given by a 4×4 homogeneous transformation in SE(3) with respect to the workstation coordinate,

$$ {}^{\mathcal{W}}\boldsymbol{T}_{\mathcal{O}}(x, y, z, \phi, \psi, \theta) = \left[\begin{array}{c|c} {}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{O}}(\phi, \psi, \theta) & {}^{\mathcal{W}}\boldsymbol{P}_{\mathcal{O}} \\ \hline \boldsymbol{0} & 1 \end{array}\right]. \quad (1) $$

The rotation matrix ${}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{O}}$, parametrized by the Euler angles ϕ(roll), ψ(pitch), and θ(yaw), describes the orientation of the object, and ${}^{\mathcal{W}}\boldsymbol{P}_{\mathcal{O}} = [x, y, z]^T$ is the position of the grasp point of object $\{\mathcal{O}\}$ in the workstation coordinate $\{\mathcal{W}\}$. Transformation matrices between the other coordinates obey the same definitions with respect to their coordinates, shown in Fig. 2.

Let $\{\mathcal{M}\}$ be the coordinate frame attached to the grasp point of the manipulator. Moreover, coordinate frame $\{\mathcal{B}\}$ is attached to the mobile platform as its control point. For the purpose of mobile robot navigation, $\{\mathcal{M}\}$ is fixed

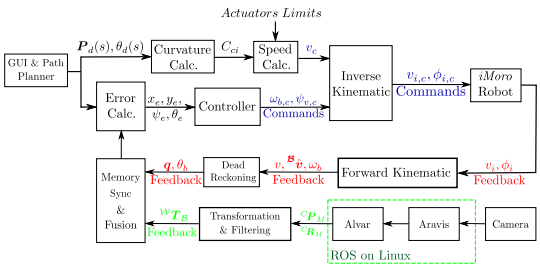


Fig. 3. Vision-based controller architecture.

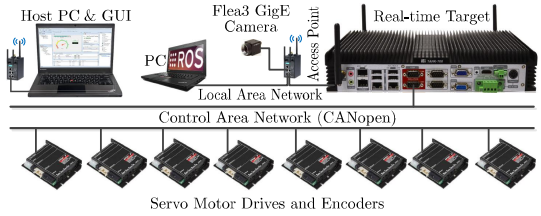


Fig. 4. Relevant hardware architecture for vision-based navigation.

on the mobile base. Therefore ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{M}}$ is a known constant for this problem. Since this paper tackles navigation as a manipulation problem, the test case starts from the moment the robot has the workstation in the camera *Field of View* (FOV) in the vicinity of the workstation where the smoothness and repeatability of the system play a key role in mobile manipulation. According to Fig. 2 and the above-mentioned definitions, the control objective is to control the body frame $\{\mathcal{B}\}$ such that the gripper $\{\mathcal{M}\}$ coincides with object $\{\mathcal{O}\}$, and therefore ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{M}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{O}}$. Since this paper tackles the problem of mobile robot navigation, roll, pitch, and z-axis are not controllable and the object and gripper coordinates are aligned. Therefore, the controller has to navigate ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{M}}(x, y, \theta)$ for object grasping,

$$ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{M}}(x, y, \theta) = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{W}}(x, y, \theta)\ {}^{\mathcal{W}}\boldsymbol{T}_{\mathcal{O}} \quad (2) $$

where ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{W}}(x, y, \theta)$ and ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{M}}(x, y, \theta)$ are subject of observation and control, respectively. The object's pose in the workstation ${}^{\mathcal{W}}\boldsymbol{T}_{\mathcal{O}}$ is considered as a known constant transformation matrix for the controller.

A. Solution Concept

To achieve the controller objectives and requirements, the problem is divided into different parts. We can show the contribution of each part as its effect on the geometric transformation of the coordinates concerning the control objective in (2).

$$ {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}\ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{M}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}\ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}\ {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{W}}\ {}^{\mathcal{W}}\boldsymbol{T}_{\mathcal{O}} \quad (3) $$

in which,

- ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}$ is localization information,
- ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{M}}$ is the manipulator's gesture fixed on the body frame,
- ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}$ is the camera extrinsic calibration parameters,
- ${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{W}}$ is the workstation coordinate detected by the camera, and
- ${}^{\mathcal{W}}\boldsymbol{T}_{\mathcal{O}}$ is the object pose in the workstation.

In Section III, we show how the right-hand side of (3) is estimated. Section IV describes how the controller positions the robot to generate the lhs terms of (3).

III. Sensor Fusion

To pick up an object, the controller has to receive an accurate pose as a target with respect to its control point on the robot body. Near a workstation it is even more

critical to make a precise estimation of the object because the path-following coordinate changes into the workstation's coordinate to prepare to contact the environment. Generally speaking, it is common in sensor fusion that the data are different in

- sources that need to be synchronized,
- qualities that need to be filtered, and
- coordinates that need to be calibrated and transformed.

Each of these differences has its own effect on controller performance, which we discuss in detail in the following. The ODT provides vision feedback, as described in Subsection III-A. It is synchronized by the logics provided in III-D. The acquired results are combined on the same time frame as described in III-C to provide feedback for the motion controller described in Section IV. The data flow between these blocks is shown in Fig. 3. Since the object detection and fusion techniques are thoroughly discussed in the literature, the main part of this section is dedicated to precise timing and synchronization challenges, which are more important for our work compared to contemporary studies in the field.

A. Object Detection

For the purpose of target tracking, localization, or any action that requires feedback from the environment an extrospective sensory information is necessary. Without loose of generality, we consider a fiducial marker detection system as an ODT for navigational purposes. The fusion and navigation methods are compatible with any precise position-based object detection system. The hardware architecture is configured as illustrated in Fig. 4. There are two field networks on the robot: CAN and LAN. All the time-critical sensors and actuators are connected to CAN. For components relevant to ODT or remote programming, all are connected to LAN, including a PC running Robotic Operator System (ROS).

A low-cost GigE Point Grey Flea3 camera is used as the camera sensor. The camera transmits the images to Aravis[2], an Ethernet camera driver that has a ROS interface. We use a ROS wrapper of Alvar[3], as an AR tag tracking library, for marker detection. The intrinsic camera parameters are calibrated by the camera calibration package of ROS. For the extrinsic parameters, a transformation matrix for the pose of the effective camera frame in the body coordinate is calculated through the calibration method described in [7] to extract ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}$. More detailed information is explained in Table II.

B. Practical Issues

For the experimental evaluation, we studied the open-loop behavior of the system. We recorded the extracted poses from different distances in front of the marker, as shown in 5. According to the figures, the data show correlation between ${}^{\mathcal{W}}x_{\mathcal{B}}$ and ${}^{\mathcal{W}}y_{\mathcal{B}}$ because both are under the influence

[2]https://wiki.gnome.org/action/show/Projects/Aravis
[3]http://virtual.vtt.fi/virtual/proj2/multimedia/

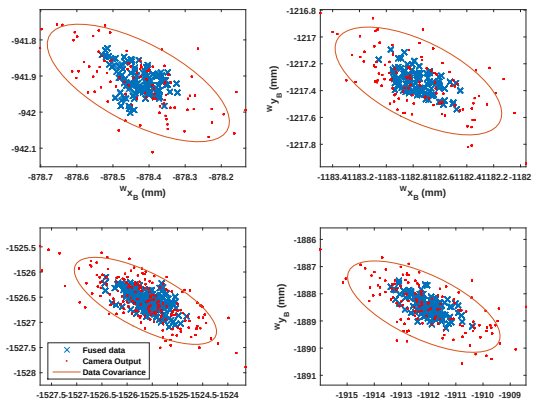


Fig. 5. Position uncertainties are reduced by filtering.

of the estimated yaw angle between the robot frame and the workstation frame.

As mentioned in Subsection III-C, the pose of $\{\mathcal{W}\}$ is assumed to be fixed. However, we used a wheeled table as the workstation. We performed serveral tests and changed the pose of the workstation during the robot's motion to test the dependency of the solution on this assumption. As shown in the multimedia attachment, the robot is still capable of catching the object except when large changes of the workstation that move the marker out of the camera FOV.

Our experience with ODT is similar to [18] except in the near-singular positions. Due to the weakness of monocular vision systems in depth estimation, ODT becomes singular when

- $\{\mathcal{C}\}$, the camera, becomes parallel to the workstation marker $\{\mathcal{W}\}$ and
- the angle between the image frame and marker frame becomes smaller than $15°$, while the distance between the frames is less than $1m$, which is wider than our expectation.

Using a bundle of markers in different orientations can be an effective solution for these issues. However, it affects our analysis on properties of a monocular vision system as in general.

For a short period of time, for example $4s$ at a speed of $0.6\ {}^{m}/_{s}$, the visual information is substituted by (5). However, for longer delays and larger changes to the heading, the robot diverges from the given path until the target appears in the camera's FOV.

C. Geometrical Representations of Fused Data

To track the pose of the object by the manipulator, according to (3) we want to observe the pose of the robot with respect to the workstation coordinate. A marker is installed on one side of the workstation, as shown in Fig. 1, which defines $\{\mathcal{W}\}$ for the camera as illustrated in Fig. 2. ODT transmits the pose of the workstation with

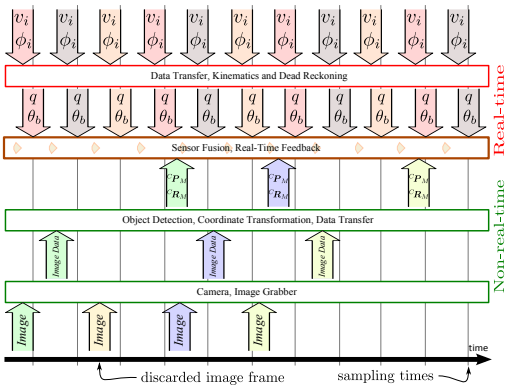


Fig. 6. Timetable for data processing.

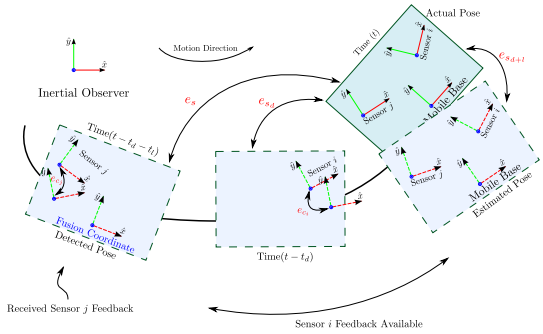


Fig. 7. How synchronization affects position accuracy.

respect to the camera frame ${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{W}}$. Since the workstation and the inertial frame are supposed to be fixed, we estimate the correspondent transformations between the coordinate frames as a loopclosure,

$$ {}^{\mathcal{I}}\hat{\boldsymbol{T}}_{\mathcal{W}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}\, {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}\, {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{W}}. \tag{4} $$

This equation's rhs, compared to (3), includes only the parameters that depend on the moving platform's motion, regardless of the manipulator gesture. The inertial coordinate frame is assumed to be attached to the robot's initial pose. Therefore, ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}$ is provided by the wheel odometry through dead reckoning of the velocity frame [19]. After few iterations, filtering the values of ${}^{\mathcal{I}}\hat{\boldsymbol{T}}_{\mathcal{W}}$ provides a robust and smooth estimation. This estimation resolves the issues regarding slowness of the ODT by providing an estimation of the workstation. Moreover, in case of a failure in the ODT, for instance temporary occlusion of the view, this estimation provides the robot its absolute pose instead of the ODT explained in Subsection III-A,

$$ {}^{\mathcal{W}}\hat{\boldsymbol{T}}_{\mathcal{B}} = \left({}^{\mathcal{I}}\hat{\boldsymbol{T}}_{\mathcal{W}} \right)^{-1} {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}} \ . \tag{5} $$

To avoid numerical issues, the inverse of the homogeneous transformation matrix is derived by the transposing of its rotational part.

D. Synchronization of Vision Data

As addressed in Table I, the controller's sampling time is significantly shorter than the elapsed time for a global feedback. For low-accuracy or slow motions, the movements during the process interval are not large enough to affect control performance. Moreover, being "fast enough" is used as an alternative for being in "real-time". Unlike stationary systems, effects of the timing errors can result in more position/orientation errors for a moving system of coordinates. These delay effects are added to the measurement and calibration errors as we discuss in the following for two types of sensors: i and j according to Fig. 7.

TABLE II
SYSTEM SPECIFICATIONS

Component	Specification	Value
Target PC	CPU	Core i7
	Max Peak to Peak Jitter	6×10^{-15}s
Camera	Image Size	1288×960 px^2
	Shutter Time	0.016s

Sensor type i periodically streams raw data, for example, wheel encoders and Inertial Measurement Units (IMU)s on a real-time network. The algorithms associated with these types of sensors have deterministic amounts of computational costs or predictable numbers of iterations that guarantees availability of data in a certain time interval. Examples of these processes can be transfer functions or kinematic formulations for a joint encoder or an IMU. This configuration can be a base for updates of the real-time feedback if they are real time and fast enough compared to the controller sample time. For the test case of this paper, the robot's wheel encoders fulfill this requirement. The Maxon Epos Drives transmit filtered encoder data to the real-time target every $5ms$. Note that for a low-resolution wheel encoder, this sensor can be triggered by the changes in wheel angle instead of a timer. For example, in the authors' previous work on an autonomous forklift [13], the wheel rotation resolution was $17\frac{\text{pulses}}{\text{round}}$, which means there is not necessarily an update for a short distance movement. Therefore, resolution of sensor i can also affect the minimum allowable velocity, frequency of local updates, and accuracy of positioning for sensor i's data. In this paper, we assume that the robot's odometry is sensitive enough to update the data within the minor sample time of the fusion block.

Sensor type j transmits raw data of an absolute pose, e.g., a camera, Time of Flight Camera (TOF), or Laser Range Finder (LRF), or a Global Position System (GPS). This can be time-based or event-based. The algorithms associated with this type of sensor are based on filtering and matching techniques or other iterative methods. These types of sensors can reject the accumulated errors or sensor drifts of local feedback, although they have high process costs.

As shown in Fig. 7, assume that different types of sensors

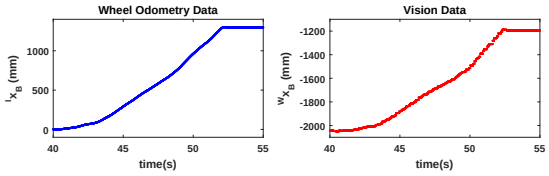


Fig. 8. Recorded raw data for manual driving of the mobile robot.

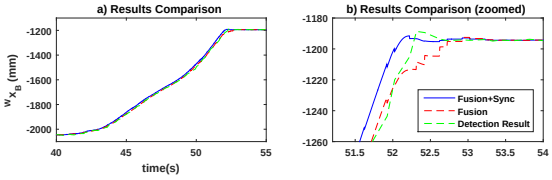


Fig. 9. Effects of synchronization on the experimental results.

(i and j) are attached to the robot's body to update the estimation of (4) and (5). The pose of the sensors have been calibrated with respect to the body frame. Extracted information from sensor j's data arrive at the moment t of the real-time target PC after a few variable time-consuming steps, as shown in Fig. 6. We record and compare the system clock twice: when the image arrives at the ROS PC and when it leaves. The difference between these two timestamps, t_d, is considered to be the process delay. We know that the process cost for image processing and machine vision algorithms can change depending on the environmental conditions and complexity of the scene. A sample set of values for t_d is shown in Fig. 10. We know that the received information from the ODT is not for the current pose of the robot. Ignoring this difference adds the e_s error of Fig. 7.

In addition to the described process delay, there are other sources of delay that are not included in the t_d. There can be several sources for the latencies in buffering, communications on the network, and grabbing the image. Here, we assume them as latency. For the experiments in this paper, we considered $t_l = 0.02s$ as a constant latency in addition to the process delay t_d. These timings are more important when the main sample time for the real-time system becomes $5ms$. Therefore, we temporarily store the latest information data of Sensor i for a time interval of $max(t_d + t_l)$. This provides an option for synchronization to calculate the movements between $\boldsymbol{T}(t-t_d-t_l)$ and $\boldsymbol{T}(t)$. Therefore, the errors e_s or e_{s_d} can be reduced to $e_{s_{d+l}}$. The effects of calibration and measurement errors remain as e_{c_i} and e_{c_j}. The experiments show the improvement of the feedback as a result of synchronization, as shown in Fig. 9. The mathematical representation becomes clearer if we recall that all of the rhs terms in (4) are time variant, and we can have the equilibrium if we consider their values for the same time frame. We rewrite (4) for the estimated captured moment as

$$^{\mathcal{I}}\hat{\boldsymbol{T}}_{\mathcal{W}} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(t-t_d-t_l)\ {}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}\ {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{W}}(t-t_d-t_l). \quad (6)$$

Since we assume a fixed pose for the workstation, (5) is still valid for providing the real-time feedback. It should be noted that this technique significantly reduces the *position errors caused by the delays*. However, timing effects of the delays are obvious in the experimental results. The aim of this technique is to choose "delayed precise pose" rather than "delayed imprecise pose."

IV. Motion Control of the Mobile Base

This section briefly describes the incorporated path-following controller that is utilized in this study, the details of which are given in [9] and [17]. The main benefit of the controller is its decoupled framework. It provides an online asymptotic path planner that guides the mobile platform toward a given desired path. Additionally, a bounded velocity trajectory generation scheme is built that provides synchronous bounded velocities for all the driving and steering actuators while honoring the velocity limits of the actuators. As shown in [9], the controller enables the execution of high maneuvers even close to the platform's singularities.

Let $\boldsymbol{P}_d$ be a sufficiently smooth desired path with s denoting its natural parametrization and $\theta_d(s)$ as the platform's desired heading. The final point of the desired path $\boldsymbol{P}_d(s_f)$ satisfies the desired value for ${}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}$ in (3) by the negligible tolerance. Additionally, $\mathcal{T}\{\hat{\boldsymbol{t}}, \hat{\boldsymbol{n}}\}$ is the tangent frame at point $\boldsymbol{P} = \boldsymbol{P}_d(s)$ and ψ_t represents the angle between $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{X}}$. We define the error signals x_e and y_e as position errors measured along $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{n}}$, respectively, and $\theta_e = \theta_d - \theta_b$ as the platform's heading error in which θ_b is the platform's heading. The following equations describe

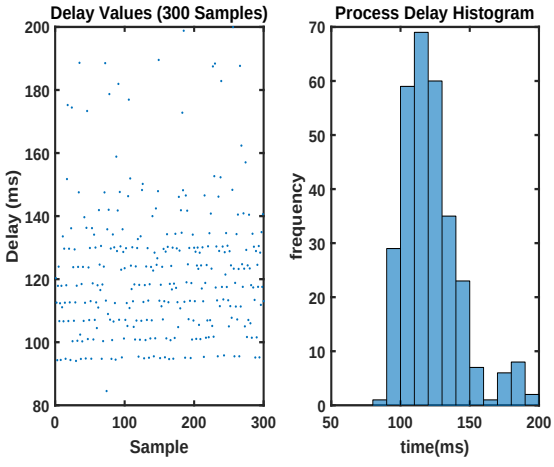


Fig. 10. Variation of the process delays calculated on the ROS PC.

the error dynamics of the system:

$$\dot{x}_e = \dot{s}(C_c(s)y_e - 1) + v\cos(\psi_e) \tag{7a}$$
$$\dot{y}_e = -\dot{s}C_c(s)x_e - v\sin(\psi_e) \tag{7b}$$
$$\dot{\theta}_e = \frac{\partial\theta_d}{\partial s}\dot{s} - \omega_b \tag{7c}$$

in which $\omega_b = \dot{\theta}_b$ is the angular velocity of the platform's body frame $\mathcal{B}$. Define σ as

$$\sigma(y_e) = -\sin^{-1}\frac{k_2 y_e}{|y_e| + \epsilon}, \tag{8}$$

where $0 < k_2 \leq 1$ and $\epsilon > 0$. $\sigma(y_e)$ is a function that generates an appropriate approach angle from the platform to the desired path.

As shown in [9], for the mobile platform's base, the feedback control laws are given by

$$\dot{s} = (k_1 x_e + \cos\sigma(y_e))v = k_s v \tag{9a}$$
$$\omega_b = (k_3\theta_e + \frac{\partial\theta_d}{\partial s}k_s)v = k_b v \tag{9b}$$
$$\psi_v = \psi_t - \sigma(y_e), \tag{9c}$$

where ψ_v is the direction of the base linear velocity and $k_1, k_3 > 0$. The controller provides asymptotic convergence toward the desired path and heading profile for any arbitrary base speed $v(t) \geq v_m > 0$. The speed $v(t)$ is then chosen such that the resultant actuators' control signals have velocities bounded by their prespecified absolute limits. The procedure is as follows: The following scalar constraints map the base speed v to the i^{th} wheel driving velocity v_i and steering velocity $\dot{\phi}_i$:

$$v_i = \sqrt{v^2 + \omega_b^2 l_i^2 + 2v\omega_b l_i \sin\eta_i} \tag{10a}$$
$$v_i^2\ \dot{\phi}_i = l_i\left(\dot{\omega}_b v - \dot{v}\omega_b\right)\cos\eta_i + v\left(\omega_v - \omega_b\right)\left(v + l_i\omega_b\sin\eta_i\right) \tag{10b}$$

in which ψ_v is the direction of the base linear velocity, $\omega_v = \dot{\psi}_v$, l_i is the distance between the wheels' steering axis and origin of the base frame, and η_i is the angle between base linear velocity $\hat{\boldsymbol{v}}$ and $\boldsymbol{\ell}_i$. As shown in [17], insertion of the base control signals into the above kinematic constraints simplifies them into explicit relations in the form of

$$v_i = v_i' v, \qquad \dot{\phi}_i = \phi_i' v, \tag{11}$$

in which v_i' and ϕ_i' are functions of the state variable and their partial differentiation with respect to the desired path variable s. This design allows us to find the base speed v that honers the actuator limits by deriving it using

$$v^{(\max)} = \min_i\left(\frac{v_i^{(\max)}}{|v_i'|}, \frac{\dot{\phi}_i^{(\max)}}{|\phi_i'|}\right), \tag{12}$$

in which $v_i^{(\max)}$ and $\dot{\phi}_i^{(\max)}$ are the maximum limits of the i^{th} wheel driving and steering velocities, respectively. The wheels' driving and steering velocities that are the platform inputs are then derived utilizing $v^{(\max)}$ and the kinematic constraints.

V. Experiments

The proposed method is implemented on the mobile platform of iMoro mobile manipulator in a setup shown in Fig. 1 and hardware architecture shown in Fig.4. According to the evaluations addressed in Subsection III-B, we adjusted the workstation marker to avoid ODT singularity.

A planner sends the robot a set of control points for a cubic Bezier curve defining an obstacle-free path. It starts from the initial pose of the robot and ends at the grasp point of the object. Usually, the robot starts moving from a distance of $3 - 5m$ in the room and finishes the path in the vicinity of the object, such that the object remains inside the jaw fingers of the gripper, as shown in Fig. 13. About 12 tests were done with the same scenario, and all the results had enough accuracy in heading and control of the y axis. Maximum errors were $\pm 2.5cm$ on the y-axis and $\pm 3cm$ on the x-axis with respect to the grasp point.

We also performed tests involving random changes in the place of the workstation to disturb (5) during the robot's motion. The test results were satisfactory by the same margins. However, in cases when we force a large rotational disturbance on the workstation, robot changes its heading during the filter's resampling. Therefore, the object leaves the FOV without enough updates of (4). A sample output of the system is shown in Fig. 11. The path for the wheels for the same motion is illustrated in Fig. 12 with respect to the initial frame of the wheel odometry $\{\mathcal{I}\}$.

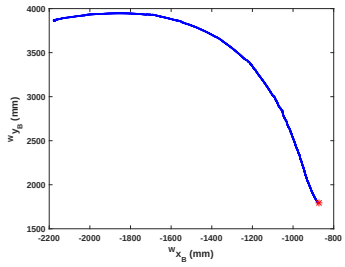


Fig. 11. The path for the robot's coordinate frame with respect to the workstation coordinate frame.

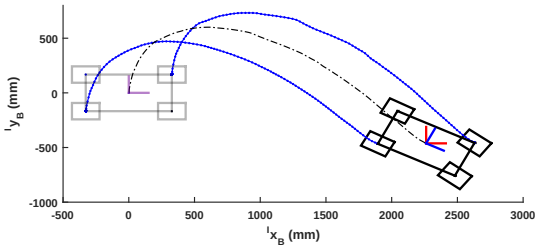


Fig. 12. Robot's footprint based on the test results with respect to the inertial coordinate as the initial position of the wheel odometry.

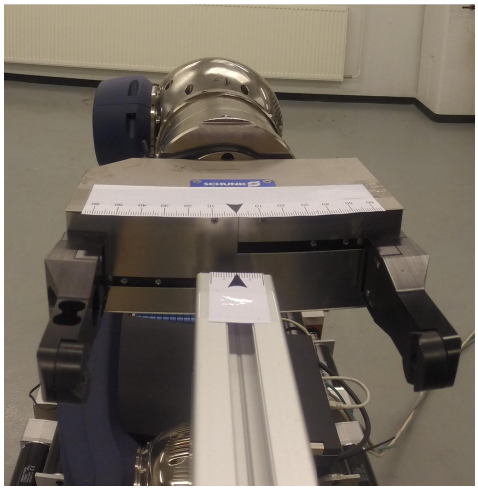


Fig. 13. Robot's final gesture from the viewpoint of the object.

VI. Conclusions

In this paper, we presented a framework for vision feedback integration into real-time motion control that mainly focuses on synchronization issues resulting from the inherent delays in visual feedback. For mobile manipulators, this approach advances the role of the mobile platform as the major contributor in wide motions of the manipulator's end effector. The method is then combined with the authors' Macro–Micro architecture for mobile manipulation, and we show that the synchronized visual feedback provides the controller with a degree of accuracy that enables it to properly track an object and grasp it with the gripper. This precision makes the mobile robot's performance comparable to manipulators. Implementation of dead reckoning and wheel kinematics in the sensor fusion by particle filter is planned for future work.

Acknowledgment

This work is supported by the Academy of Finland under the project "Integrated Multimodal Sensing of 3D Environment for Intelligent Manipulators," grant no. 286260.

References

[1] J.-D. Yoder, J. West, E. Baumgartner, M. Perrollaz, M. Seelinger, and M. Robinson, "Experiments comparing precision of stereo-vision approaches for control of an industrial manipulator," in *Experimental Robotics*, pp. 245–256, Springer International Publishing, 2013.

[2] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 2, pp. 237–267, 2002.

[3] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems*, vol. 53, no. 3, pp. 263–296, 2008.

[4] F. Dalgleish, S. Tetlow, and R. Allwood, "Vision-based navigation of unmanned underwater vehicles: A survey. Part 2: Vision-based station-keeping and positioning," *Proceedings of the Institute of Marine Engineering, Science and Technology. Part B, Journal of marine design and operations*, no. 8, pp. 13–19, 2005.

[5] A. Cherubini and F. Chaumette, "Visual navigation of a mobile robot with laser-based collision avoidance," *The International Journal of Robotics Research*, vol. 32, no. 2, pp. 189–205, 2013.

[6] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *Robotics, IEEE Transactions on*, vol. 23, no. 1, pp. 87–100, 2007.

[7] M. M. Aref, R. Ghabcheloo, and J. Mattila, "A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong*, pp. 6816–6822, 2014.

[8] R. Oftadeh, M. M. Aref, R. Ghabcheloo, and J. Mattila, "Real-time system integration for mobile manipulation," *International Journal of Advanced Robotic Systems*, vol. 11, no. 15, 2013.

[9] R. Oftadeh, R. Ghabcheloo, and J. Mattila, "A novel time optimal path following controller with bounded velocities for mobile robots with independently steerable wheels," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Japan, 2013.

[10] J. Röwekämper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard, "On the position accuracy of mobile robot localization based on particle filters combined with scan matching," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 3158–3164, IEEE, 2012.

[11] C. Sprunk, B. Lau, P. Pfaff, and W. Burgard, "An accurate and efficient navigation system for omnidirectional robots in industrial environments," *Autonomous Robots*, pp. 1–21, 2016.

[12] P. De Cristóforis, M. Nitsche, T. Krajník, T. Pire, and M. Mejail, "Hybrid vision-based navigation for mobile robots in mixed indoor/outdoor environments," *Pattern Recognition Letters*, vol. 53, pp. 118–128, 2015.

[13] M. M. Aref, R. Ghabcheloo, A. Kolu, and J. Mattila, "A multistage controller with smooth switching for autonomous pallet picking," *IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden*, 2016.

[14] A. Dietrich, K. Bussmann, F. Petit, P. Kotyczka, C. Ott, B. Lohmann, and A. Albu-Schäffer, "Whole-body impedance control of wheeled mobile manipulators," *Autonomous Robots*, pp. 1–13, 2015.

[15] S. Cousins, "ROS on the PR2 [ROS topics]," *IEEE Robotics & Automation Magazine,*, vol. 17, no. 3, pp. 23–25, 2010.

[16] C. Connette, A. Pott, M. Hägele, and A. Verl, "Addressing input saturation and kinematic constraints of overactuated undercarriages by predictive potential fields," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4775–4781, IEEE, 2010.

[17] R. Oftadeh, R. Ghabcheloo, and J. Mattila, "A time-optimal bounded velocity path-following controller for generic wheeled mobile robots," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 676–683, IEEE, 2015.

[18] K. Pentenrieder, P. Meier, G. Klinker, *et al.*, "Analysis of tracking accuracy for single-camera square-marker-based tracking," in *Proc. Dritter Workshop Virtuelle und Erweiterte Realitt der GIFachgruppe VR/AR, Koblenz, Germany*, 2006.

[19] M. M. Aref, R. Oftadeh, R. Ghabcheloo, and J. Mattila, "Fault tolerant control architecture design for mobile manipulation in scientific facilities," *International Journal of Advanced Robotic Systems*, vol. 12, p. 4, 2015.

Publication VI

M. Aref, M., Ghabcheloo, R., Kolu, A., and Mattila, J., "Vision guided autonomous forklift," *Advances in Robot Design and Intelligent Control (RAAD 2016)*, pp. 338 – 346, 2017

Vision-Guided Autonomous Forklift

Mohammad M. Aref ✉, Reza Ghabcheloo,
Antti Kolu, and Jouni Mattila

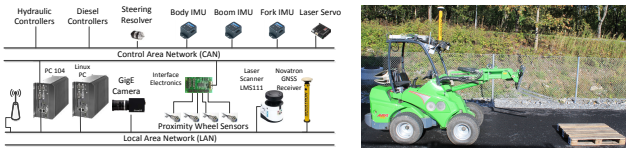
Department of Intelligent Hydraulics and Automation,
Tampere University of Technology, FI-33101 Tampere, Finland.
m.aref@ieee.org,{reza.ghabcheloo,antti.kolu,jouni.mattil}@tut.fi

Abstract. This paper tackles the problem of integrating *Visual Servoing Control* (VSC) into the functionalities of an *Articulated-Frame-Steering* (AFS) hydraulic forklift. The controller is capable of breaking down high-level messages into piecewise commands for the different software modules of the vehicle. It also preserves seamless cooperation of the modules for a successful pallet-picking mission. The proposed architecture have been verified on a real machine. Videos of the test runs are available on YouTube.

Keywords: mobile manipulation, visual servoing, state machine.

1 Introduction

Robotics has become an important aspect of logistics and automation in many application fields. Efficient material handling plays a significant role in the optimization of construction processes among others. The invention of containers and pallets has radically changed the way logistics is performed in all sectors, particularly in the construction business, both for buildings and infrastructures, e.g. roads. Automated Guided Vehicles (AGVs) are possibly among the best-selling robotic units, used mostly for warehouse management, such as the Kiva Systems of Amazon [1]. In indoor cases following the traditions of factory robots both palletizing systems and AGV localizations and navigation are rather accurate, and system operation relies on this certainty. However, in outdoor applications, e.g. in construction, the pallet-handling control system needs to be robust against both localization and control inaccuracies. Localization may be improved using more expensive equipment, though for a cost-conscious industry such as construction, this is not usually acceptable. Forklift trucks used for pallet handling in construction are built to move on uneven terrain, and they are made robust with simple components, which makes the accurate control of these machineries a real challenge. Notice that, contrary to typical logistics, the challenge is not only in navigation in semi-/un-structured and dynamic environments but also in the manipulation and coordination of the body and the boom. In a series of publications, we have proposed methodologies for vision-guided navigation that are robust against the uncertainties mentioned above [2,3,4]. In this paper, we present the details of the control system's architecture and report on



(a) Hardware architecture (b) Autonomous forklift

Fig. 1: Autonomous forklift and its hardware.

the system integration of such a complex system. For vision-based autonomous systems, errors in target detection and limitations of the mechanical system are both common sources of error [5]. A proper controller should overcome both obstacles, improving overall performance. In the literature, many studies focus on one of these issues over the other. A group of studies gives more weight to the pallet-detection problem. Pallet detection and the integration of a visual sensor in the control topology are the main focuses in [6], whereas [7] aims to correct data while approaching the pallet. Most previous work has employed at least two cameras, a monocular vision system with defined CAD models [8], or both a laser scanner and a camera [9]. In this paper, we introduce a modular architecture and a state machine that integrates a VSC [2] with preexisting path-following and manipulation controllers. This architecture can fill the gap between robot functionalities for test cases and real-world problems.

2 Software Modules

This section briefly describes the software modules of the vehicle for the components shown in Fig. 1a. Their interaction is shown in Fig. 2, which is categorized by the relevant environment of each software module. The design criteria for the architecture include reliability, availability, and maintainability. Our proposed solution is a modular approach, which allows us to benefit from team work and from the use of the existing advances made in the required and relevant functionalities. Therefore, the controllers are self-dependent and not necessarily flexible. The desired flexibility is achieved by controlling the interactions between the modules by a state machine, described in Section 3. The machine's control system is composed of primitive functionalities such as path planning, path-following control (body), manipulator control (boom), and pallet detection, which are all coordinated by stateflow machines. VSC then becomes a capability built on the coordination of these primitive functions by a stateflow machine, which is detailed in this paper. An important benefit of this architecture is modularity.

2.1 Control modules

Pallet-picking problems can be defined as generating control commands that lead the fork frame to the pallet frame without any collisions. This clearly restricts the final segment of the fork path to a line that is parallel to the pallet frame. Moreover, the machine has limited maneuverability (detailed in the next section);

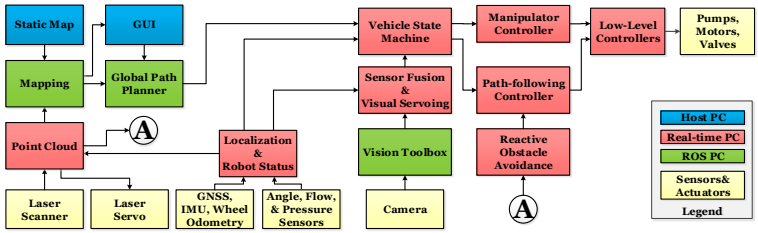


Fig. 2: Software modules of vehicle control

it requires a minimum of 3m to reduce a 0.5m lateral error to 5cm, a value that prevent collision with the pallet.

Path-Following Control: The machine in this study is a hydraulically actuated (AFS) machine (with the steering mechanism of a wheel-loader). An AFS consists of two bodies, front and rear units, joined by a hinge, each with freedom in yaw motion. Normally, these units have freedom in pitch as well, to allow good ground contact over rough surfaces. In a commercial AFS machine, bending angles are limited to about 42 degrees on each side to ensure roll stability. Given a desired 2D geometrical obstacle-free path (smooth Bézier curves, in this case) together with a desired velocity profile (trapezoidal, in this case), a path-following controller generates the linear and angular speed of the body frame attached to the middle of the front axle. A motion controller then generates the required commands for the valve controlling of the steering cylinder and for the drive power transmission controller (pump displacement and diesel RPM) to control the speed. More details are found in [10], where we showed AFS kinematics can be approximated by two attached unicycles, for which a non-linear path-following controller is designed.

Manipulator Control: As shown in Fig.1b, the forklift manipulator in this research has three *Degrees Of Freedom* (DOF) in a rotary–prismatic–rotary (RPR) configuration, in such a way that the height and reach of the fork and its pitch angle are controllable. All DOF are controlled by linear hydraulics actuators, which are driven by pressure-compensated proportional valves. In control terminology, that means an inner loop servo controller compensates for nonlinear dynamics and receives velocity control commands of the cylinders. In other words, we will control the manipulator at the kinematic level. Given the target pose in the body frame, a relevant Jacobian matrix is used to generate the valve speed commands. Notice that the roll angle of the end effector (fork) is not controllable, neither by the manipulator nor the body DOF.

2.2 ROS Packages

Robot Operating System (ROS) is used for high-level control systems for mapping, path planning, obstacle avoidance, and marker detection. The core components of the high level control system are presented in Fig. 2.

Mapping and Obstacle Detection: The mapping system uses the calibration and localization error-tolerant segmented mapping method [11]. A tilting 2D laser scanner is installed in front of the forklift. The laser range measurements are transformed from the sensor frame into the global frame using the localization information. Each point in the point cloud is attached with a height variance value based on the localization's uncertainty at the time of the range measurement. The point clouds are gathered in the mapping module, and they are processed once per tilting motion of the servo. The height of the terrain in each cell is calculated using the height of each point assigned to each cell. The Bayesian update is used to update the height, so the uncertainty of the measurement is considered. The maximum gradient for each cell is calculated, and the predefined threshold value is used to define the cell as free or occupied. A static obstacle map is combined with the map calculated based on the point cloud to avoid forbidden areas, such as roads, and areas with static obstacles, such as buildings and fences.

Path Planning: Path-planning algorithms are presented more closely in [12]. The first step of the algorithm is to compute a finite set of feasible motions connecting discrete robot states to construct a search graph offline. The motion primitives based on Bézier curves are generated by solving the constrained optimization problem. The path planning receives an obstacle map from the mapping system every time the map is updated. When a new goal is received, the current pose of the machine is considered to be the start state for the path-planning algorithm. A^* search is then conducted through the graph of motion primitives, resulting in a list of feasible motions from the start to the goal. This list of Bézier curves are combined when feasible and are then further optimized by solving the constrained optimization problem. This leads to a smooth and feasible set of Bézier curves.

Marker Detection: The images are grabbed using `camera_aravis` and published as unrectified raw images into ROS. The calibration is performed using `camera_calibration` with a calibration plate that has a 12×8 grid of 6cm square cells. The camera calibration file is saved into ROS so that it is found by the marker detection package. The camera used in our tests was a GimaGO GO423C GigE camera with a resolution of 1296×966 pixels. The camera's lens is a Kowa 1.4 lens with a focal length of 12mm. Marker detection is implemented using `ar_track_alvar` that uses the Alvar marker-detection library for agumented reality. It uses unrectified images and uses the camera calibration to rectify the images. A bundle of three individual markers is placed on the front side the pallet as shown in Fig.1b. Alvar library recognizes them as a single marker. This brings better reliability, as one or even two of the markers may be occluded or unrecognizable in the camera image.

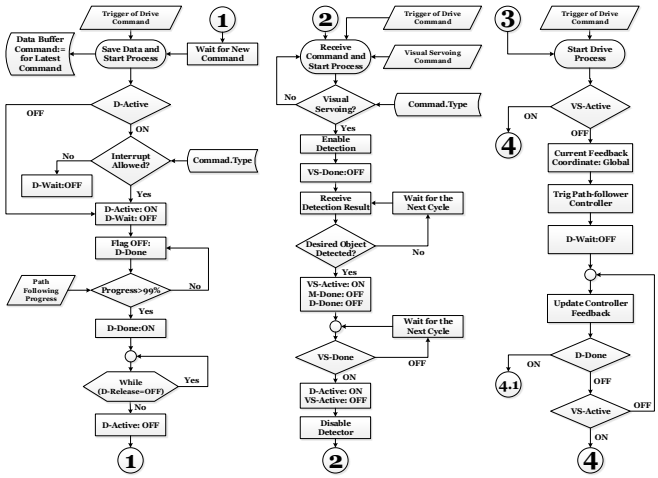


Fig. 3: Flowcharts of the state machines: 1,2) Preprocessing, 3) Processing

3 Vehicle State Machine (VSM)

As shown in Fig. 2, the messages from the planner, GUI, and any higher-level modules arrive to the VSM. This module distributes them as commands among the relevant control modules depending on their content and the robot's status. Some of these *message types* are: driving, manipulation, forced driving, forced manipulation, VSC, and placement. The VSM should handle and monitor the continuous functionality of the control modules as well as interruptions and delays.

For instance, on one hand, it should maintain smooth operation of the robot during the communication delays of the long-distance wireless network or package drop. Therefore, the VSM uses a FIFO (First In, First Out) scheduling system as a buffering method. On the other hand, the vehicle should immediately react to forced message types as the first priority. Therefore, forced message types should bypass the buffers and interrupt the current work.

In the following, a selected part of the VSM is described. It focuses on executing a sequence of driving and manipulation commands individually or as part of a VSC. The VSM runs at each minor sample time of the control system. It consists of three main steps: preprocessing, processing, and post-processing. The separation of phases not only simplifies the VSM's design process but provides synchronization of the actions before tasking the command of the control modules. The notation for the flags referred to in Fig. 3, 4, and 5 are as follows:

X-Active: allows the processing phase to turn on the corresponding controller.
X-Done: marks task accomplishment for local use.

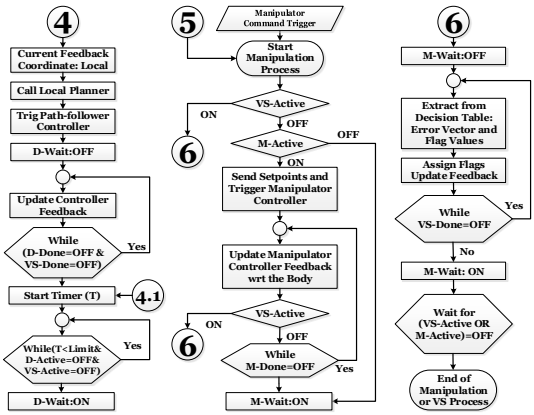


Fig. 4: Processing phase flowchart of state machine

X-Release: declares task accomplishment for the high-level modules. Turns off any busy signs for the corresponding control module.
X-Wait: pauses the command and prevents the release of the controller module.

3.1 Preprocessing

The preprocessing phase of the VSM picks the necessary control modules and updates the VSM's status according to the latest messages from the buffer. As a general rule, a block in the preprocessing phase:

1. can read inputs and flags,
2. can write internal flags, and
3. should not update any outputs.

Based on these rules, we simplify the design of the VSM and avoid vicious circles. A sample preprocessing phase for the drive command is shown by the flowchart numbers 1 and 2 in Fig. 3. Column 1 manages flags relevant to the driving control. The drive message can also carry the VS command and an object ID. Therefore, the second column makes the decision about running VS-related modules. The VSM blocks for the manipulator are similar to those in column 1. However, the statement of accomplishment depends of the 2nd-norm of the controller error. Note that the loops similar to the "While" loop of flowchart 1, avoids resetting the internal flags without informing the high-level planner. X-Release flags are updated during the postprocessing phase.

3.2 Command Processing

This phase provides the necessary data and actions for the controller modules. Flowchart 3 in Fig. 3.1 and the flowcharts in Fig. 4 show the processing phase of the VSM. The blocks of this phase:

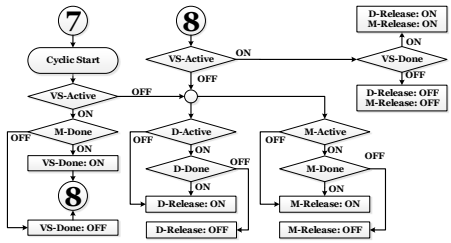


Fig. 5: Post-processing phase of state machine.

1. can read inputs and flags,
2. can write internal flags if they do not change the system status,
3. can update the outputs for the low-level modules, and
4. should not update the outputs for the high-level modules.

The VS task is based on the macro-micro controller proposed in [2].

3.3 Postprocessing

This phase updates the outputs based on the status of the controllers and the decisions made in the earlier phases of the VSM. The blocks in this phase:

1. should not read inputs,
2. should not write internal flags except those for task accomplishment, and
3. can update the outputs for the high-level modules based on the internal flags.

The postprocessing phase for driving and manipulation are shown in Fig. 5. Note that the X-Release flags in this phase are monitored by the high-level modules. Unlike the other phases, this phase is triggered by the system clock. Its blocks run sequentially at each sample time from the start to the last possible block.

4 Conclusions

The proposed method takes advantage of cooperative manipulation and path-following in different operational coordinates. The robot receives messages from a global planner and drives on the desired path. When the path contains a pallet as its end-point, the vehicle state machine (VMS) smoothly switches the modules to fit the visual-servoing architecture. In addition, VMS preserves all the functionalities of the modules to run tasks sequentially or to interrupt messages from the authorized high-level supervisory system. The architecture is implemented on an autonomous forklift made by the Department of Intelligent Hydraulics and Automation at the Tampere University of Technology, based on the Avant 635. The videos of the experimental evaluation are available on YouTube [1].

Acknowledgment. This work is supported by the Academy of Finland under the project "Integrated Multimodal Sensing of 3D Environment for Intelligent Manipulators," grant no. 286260.

[1] `https://www.youtube.com/channel/UCxemrk8NrIj-db6h06VBVLA`

Table 1: Sample Rules for Decision Making

No.	VS Status Description	1st Cond.	2nd Cond.	$[\boldsymbol{e}_x, \boldsymbol{e}_z, \boldsymbol{e}_\theta]$	Flag
1	Global plan	$VS_{Active} = 1$	${}^L\boldsymbol{e}_x \geq D_1$	$[{}^G\boldsymbol{e}_x, {}^G\boldsymbol{e}_z, {}^G\boldsymbol{e}_\theta]$	D-Wait:OFF
2	VS adjusts height	—	${}^L\boldsymbol{e}_x \leq D_1$	$[\ 0\ , {}^L\boldsymbol{e}_z, {}^G\boldsymbol{e}_\theta]$	D-Wait:OFF
3	Acceptable height	${}^L\boldsymbol{e}_z > \epsilon_z$	${}^L\boldsymbol{e}_x \leq D_2$	$[\ 0\ , {}^L\boldsymbol{e}_z, {}^G\boldsymbol{e}_\theta]$	D-Wait:ON
...					

References

1. P. R. Wurman, R. D'andrea, M. T. Barbehenn, A. E. Hoffman, and M. C. Mountz, "Kiva transporting inventory items# 3," April 2014. US Patent App. 14/095,751.
2. M. M. Aref, R. Ghabcheloo, and J. Mattila, "A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong*, pp. 6816–6822, 2014.
3. M. M. Aref, R. Ghabcheloo, and J. Mattila, "Real-time vision-based navigation for nonholonomic mobile robots," *Accepted for IEEE International Conference on Automation Science and Engineering (CASE)*, 2016.
4. M. M. Aref, R. Ghabcheloo, A. Kolu, and J. Mattila, "A multistage controller with smooth switching for autonomous pallet picking," *IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden*, 2016.
5. J.-D. Yoder, J. West, E. Baumgartner, M. Perrollaz, M. Seelinger, and M. Robinson, "Experiments comparing precision of stereo-vision approaches for control of an industrial manipulator," in *Experimental Robotics*, pp. 245–256, Springer International Publishing, 2013.
6. L. F. Holeva, E. R. Elston, M. J. Seelinger, and J.-D. S. Yoder, "Identifying and selecting objects that may correspond to pallets in an image scene," May 5, 2015. US Patent 9,025,886.
7. K. Ban, F. Warashina, M. Yamada, and Y. Namiki, "Robot system comprising visual sensor," Dec. 4, 2012. US Patent 8,326,460.
8. S. Byun and M. Kim, "Real-time positioning and orienting of pallets based on monocular vision," in *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, vol. 2, pp. 505–508, IEEE, 2008.
9. L. Baglivo, N. Biasi, F. Biral, N. Bellomo, E. Bertolazzi, M. Da Lio, and M. De Cecco, "Autonomous pallet localization and picking for industrial forklifts: a robust range and look method," *Measurement Science and Technology*, vol. 22, no. 8, p. 085502, 2011.
10. R. Ghabcheloo and M. Hyvönen, "Modeling and motion control of an articulated-frame-steering hydraulic mobile machine," *Proceedings of the 17th Mediterranean Conference on Control and Automation*, Greece, June 2009.
11. A. Kolu, M. Lauri, M. Hyvonen, R. Ghabcheloo, and K. Huhtala, "A mapping method tolerant to calibration and localization errors based on tilting 2d laser scanner," in *European Control Conference (ECC)*, pp. 2348–2353, IEEE, 2015.
12. J.-W. Choi and K. Huhtala, "Constrained global path optimization for articulated steering vehicles," *Vehicular Technology, IEEE Transactions on*, no. 99, 2015.

Publication VII

M. Aref, M., Vihonen, J., Ghabcheloo, R., and Mattila, J., "On latencies and noise effects in vision-based control of mobile robots," *Advances in Service and Industrial Robotics*, pp. 191−199, 2018

On Latencies and Noise Effects in Vision-Based Control of Mobile Robots

Mohammad M. Aref ✉*, Juho Vihonen**,
Reza Ghabcheloo, and Jouni Mattila

Tampere University of Technology, FI-33101, Tampere, Finland.

Abstract. We study the effects of variable latencies and noise-effects in vision based navigation. Based on the observations, we adapt a new robust estimation solution that is simple to integrate into a path-following controller and shown to provide a smoothed, high-bandwidth feedback for real-time control of a mobile robot. The strong dependency of steering oscillations originating from the noises and inaccuracies of the robot's pose estimates is highlighted. The system is capable of positioning the mobile manipulator's gripper in the vicinity of a target only by navigation of its nonholonomic mobile base.

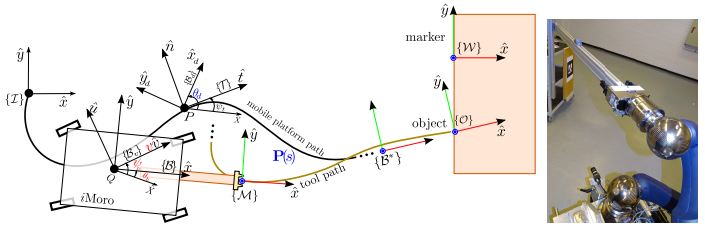
1 Introduction

Developments in prospective robotic markets and increasing demands for service robots emphasize the role of mobile manipulators and moving platforms. A mobile robot as a moving base for a manipulator or as an automated-guided vehicle has broad workspace. Precise positioning of the mobile robot has a vital role in its performance as well as manipulation functionalities. Mobile manipulators are required to be capable of long-distance movements and precise placements in certain neighborhoods so called workstations.

There are challenging issues for development of a precise mobile robot positioning systems. On the one hand, smooth, fast, and reliable commanding of actuators is time-critical. On the other hand, detection and perception algorithms are iterative, sequential and high in process cost. This paper focuses on improvement in positioning of the mobile platform (of the mobile manipulator) by integration of vision feedback into a path-following control topology to obtain necessary accuracy, reliability, and smoothness of mobile robot positioning of omnidirectional wheeled mobile robots for mobile manipulation tasks.

Comprehensive studies have been carried out on vision-based sensing for mobile robots in [1] and later in [2] or for navigation of underwater robots [3].

* Laboratory of Automation and Hydraulics Engineering.
** Laboratory of Signal Processing.
m.aref@ieee.org,{juho.vihonen,reza.ghabcheloo,jouni.mattila}@tut.fi.
This work is supported by the Academy of Finland under the project "Integrated Multimodal Sensing of 3D Environment for Intelligent Manipulators," grant 286260.



(a) Defined coordinate frames for the robot, path, and target. (b) iMoro

Fig. 1: The test case and scenario.

However, simplicity of integration, precision and repeatability have been overlooked in most of these. Moreover, there have been many researches on image-based navigation of mobile robots, especially for long-distance motions [4,5]. The downside of the image-based visual servoing control schema is that it affects the other control blocks, such as nonlinear actuator-level controllers, which brings up new challenges for system integration. We use position-based methods to benefit from both accurate visual tracking and integration with state-of-the-art pre-existing control modules. For visual servoing or vision-based navigation in a global coordinate system, the desired path or trajectory between the body frame and target frame must be mapped. However, such mappings are subject to drift [6], noise, and lack of real-time localization sensor for an affordable price.

The problem of complementary and Kalman filtering has been widely researched in the past, see, e.g., [7,8,9]. In this context, we propose a position estimation method, which is associated with explicit outlier rejection to improve position estimation in accuracy, robustness, and smoothness. There are variable latencies for machine vision techniques and differences between timestamps in the robot's real-time and non-real-time processors. Regardless of these issues, by accommodating the latencies, the proposed method is capable of providing significantly smoother and faster feedback compared to the previous method [10,11], which was developed for nonholonomic outdoor field robots. The word "nonholonomic" refers to robots whose velocity controllability depends on their pose. The methods efficacy for maker-based and marker-less object detection is demonstrated by videos of the tests[1].

2 Problem Definition

The aim of this study is to improve the precision, robustness and smoothness of the mobile base positioning in the vicinity of a target located in the workstation of mobile manipulator as shown in Fig.1b. It uses fused visual feedback and wheel odometry data to provide position estimation feedback a path-following controller [12] in real-time. This system controls the mobile robot going toward

[1] `http://robatronics.com/2017/01/10/vision-based-motion-control/`

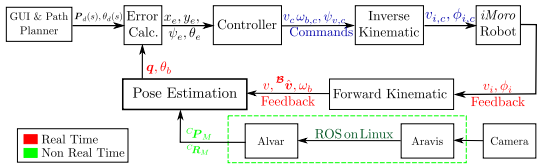


Fig. 2: Vision-based controller architecture.

the closest possible vicinity of the target object to pick it up. A selection of the requirements are:

1. Real-time feedback at each minor sample time of the system, $0.005s$.
2. Bellow $\pm 3cm$ repeatability of positioning along each axes.
3. Maximum target tracking tolerance not more than $\pm 5mm$.
4. Dealing with a variable latency object detection toolbox (ODT) with a speed of 5–10 image frames of per second.

We define a test scenario and a problem formulation to examine the functionality of positioning by different estimation methods. Assume that position and orientation of the object is given by a 4×4 homogeneous transformation with respect to the workstation coordinate,

$$^{\mathcal{W}}\boldsymbol{T}_{\mathcal{O}}(x, y, z, \phi, \psi, \theta) = \left[\begin{array}{c|c} ^{\mathcal{W}}\boldsymbol{R}_{\mathcal{O}}(\phi, \psi, \theta) & ^{\mathcal{W}}\boldsymbol{P}_{\mathcal{O}} \\ \hline \boldsymbol{0} & 1 \end{array}\right]. \tag{1}$$

The rotation matrix $^{\mathcal{W}}\boldsymbol{R}_{\mathcal{O}}$, parametrized by the Euler angles ϕ(roll), ψ(pitch), and θ(yaw), describes the orientation of the object, and $^{\mathcal{W}}\boldsymbol{P}_{\mathcal{O}} = [x, y, z]^T$ is the position of the grasp point of object $\{\mathcal{O}\}$ in the workstation coordinate $\{\mathcal{W}\}$. Transformation matrices between the other coordinates obey the same definitions with respect to their coordinates, shown in Fig. 1a.

For the purpose of mobile robot navigation, $\{\mathcal{M}\}$ is fixed on the mobile base. Since this paper tackles navigation of mobile base, the test case starts from the moment the robot has the workstation in the camera's field of view. In the vicinity of the workstation smoothness and repeatability of the system play a key role in mobile manipulation. The control objective is to make $^{\mathcal{I}}\boldsymbol{T}_{\mathcal{M}}$ equal to $^{\mathcal{I}}\boldsymbol{T}_{\mathcal{O}}$. We do not consider Roll, pitch, and z-axis controllable in this problem. Therefore, the controller has to estimate and navigate $^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(x, y, \theta)$ for the object grasping,

$$^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}(x, y, \theta) = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{W}}(x, y, \theta)\ {}^{\mathcal{W}}\boldsymbol{T}_{\mathcal{B}^*} \tag{2}$$

where $^{\mathcal{I}}\boldsymbol{T}_{\mathcal{W}}$ and $^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}$ are subjects of observation and control, respectively. The object's pose in the workstation $^{\mathcal{W}}\boldsymbol{T}_{\mathcal{B}^*}$ is considered as a known constant transformation matrix to the desired final pose of the mobile platform.

A 1MPx GigE Point Grey Flea3 camera is used as the vision sensor. The camera transmits the images to Aravis[2], an Ethernet camera driver that has an

[2] https://wiki.gnome.org/action/show/Projects/Aravis

interface to Robotic Operation System (ROS). We use a ROS wrapper of Alvar[3], as an AR tag tracking library, for marker detection. The intrinsic camera parameters are calibrated by the camera calibration package of ROS. For the extrinsic parameters, a transformation matrix for the pose of the effective camera frame in the body coordinate is calculated through the calibration method described in [10] to extract ${}^{\mathcal{B}}\boldsymbol{T}_{\mathcal{C}}$. Detailed hardware architecture is similar to [13].

3 Complementary M-Estimation Method

Consider the problem of combining two redundant noisy measurements, $\mathcal{X} = \dot{\mathcal{Y}}$ and $\mathcal{Y}$, to produce an estimate $\hat{\mathcal{Z}}$ of the signal $\mathcal{Z}$ without introducing any unnecessary distortion. Furthermore, assume that the measurement $\mathcal{X}$ is integrated to attenuate noise. If $G(s)$ is made a low-pass filter, $1 - G(s)$ is the complement, i.e., a high-pass filter, and $\frac{1}{s}$ denotes an integrator in Laplace domain, the well-known principles of complementary and Kalman filtering can be applied: i.e., we can write

$$\hat{\mathcal{Z}}(s) = \big(1 - G(s)\big)\frac{\mathcal{X}(s)}{s} + G(s)\mathcal{Y}(s), \tag{3}$$

which restores the signal $\mathcal{Z}$ without introducing any delay, assuming that the noise in $\mathcal{X}$ is mostly low frequency and the noise in $\mathcal{Y}$ is mostly high frequency. Therefore, in the theoretical case of noiseless and nonlinearity-free measurements, the signal $\mathcal{Z}$ is estimated perfectly (e.g. [14]). For $w = 1$, the following differential equation

$$\begin{bmatrix} \dot{\hat{\mathcal{Z}}} \\ \dot{\hat{b}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathcal{Z}} \\ \hat{b} \end{bmatrix} + w \begin{bmatrix} k_{\mathrm{P}} \\ k_{\mathrm{I}} \end{bmatrix} (\mathcal{Y} - \hat{\mathcal{Z}}) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \mathcal{X} \tag{4}$$

implements a classic 1-axis complementary filter, where $\mathcal{Y}$ is used as a low-pass filtered, long-term reference to decouple the interaction of the short-term information-bearing content and the slow time-varying bias dynamics of $\mathcal{X}$. That is, since the filter features a simple PI-type, closed-loop integrator with the proportional gain k_{P} and integral gain k_{I} determining the filter dynamics, a smoothed bias-free estimate of the signal can be rendered without detailed description of the noise processes.

The above can be implemented as a normal Kalman filter, but the ODT noise in our experiments is highly range dependent. Alternatively, a particle lter may be also considered, but a large number of particles is required for position estimation at a high computational cost, as shown in [15]. For this reason, we adjust the two gains adaptively by manipulating the term w in real-time. Thus, if $\hat{\mathcal{Z}}$ represents the estimate of our interest and $\mathcal{Y}$ represents a noisy version of $\mathcal{Z}$, the gains k_{P} and k_{I} can be adjusted to accommodate the varying noise statistics for as follows

$$w = e^{-\left(\frac{\|\mathcal{Y} - \hat{\mathcal{Z}}\|}{c}\right)^2}. \tag{5}$$

That is, we propose evaluating the relative confidence in the context of the robust Welsch function (e.g., [16]), which represents one of the most common

[3] http://virtual.vtt.fi/virtual/proj2/multimedia/

techniques to reduce the influence of outlying data in the statistical literature. For example, occlusions might render the optical readings almost unusable, motivating a solution relying more on wheel odometry readings. Subsequently, the gains were modified in real time to prevent outlying camera-based information from corrupting the filter states (4). Since the function (5) is often utilized in the context of the so-called M-estimators, we refer the filter to as the complementary M-estimator. This estimation for each principle axis of motion is represented as

$$\hat{\mathcal{Z}}_n = f_m(\mathcal{X}, \mathcal{Y}, \hat{\mathcal{Z}}_{n-1}) \tag{6}$$

where $\hat{\mathcal{Z}}_n$ and $\hat{\mathcal{Z}}_{n-1}$ are the filter output for the current sample and the last sample. Variable $\mathcal{X}$ contains velocity data from the forward kinematics of the system based on the wheels and steerings encoders. Variable $\mathcal{Y}$ contains pose data for the same based on ODT output. For spatial rotations, unlike planar motions, $\mathcal{Y}$ has to be on the same coordinate of $\int \omega dt$ instead of Euler angles or roll, pitch and yaw values.

As a remark, we note that evaluating the relative confidence of the measurements is not only important because it allows one to reject clearly outlying data but also because often even the outliers may contain relevant pieces of information. Thus, evaluating the relative confidence like above is by no means a synonym for rejecting some data completely.

Discretized version of this filter is implemented in two steps. First an estimation of ${}^{\mathcal{W}}\boldsymbol{R}_{\mathcal{B}}$ is calculated and then the velocity vector of the body, as a free vector, and ODT provided position vector are mapped into the workstation coordinate. The mapped vectors are fused to extract the body's relative position.

The data acquisition from the optical system as well as the subsequent onboard processing is by no means strictly real-time. Typically, the pose of the robot is resolved in about 70-200 ms. Effects of the timing delays can result in position/orientation errors for a moving system of coordinates. In our case, for example, the received information from the optical system is delayed under fast motions and thereby does not represent the current pose of the robot; i.e., the process cost for image processing and machine vision algorithms change depending on the environmental conditions and complexity of the scene. To synchronize and compensate for the variable ODT delays, we propose projecting the ODT information forwards using interoceptive sensor readings

$$\hat{\mathcal{Y}}(t) = \mathcal{Y}(t - t_d) + \int_{t-t_d}^{t} \mathcal{X}(\tau) d\tau. \tag{7}$$

This is to say that, in reality, $\mathcal{Y}$ in (3)–(6) is delayed by some delay t_d seconds at time t. This delay can be calculated from the timestamps of our ODT PC, which allows us to compensate for the variable ODT delays. Owing to the inherently small delays, the estimation may be also considered drift free. The above-mentioned variables are in the workstation coordinate $\{\mathcal{W}\}$.

4 Benchmarking in Real-time Experiments

To demonstrate the range-dependent noise rejection and latency compensation in action, this section presents an experimental study on the methods and analysis

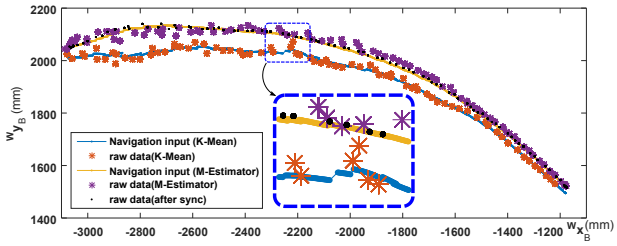


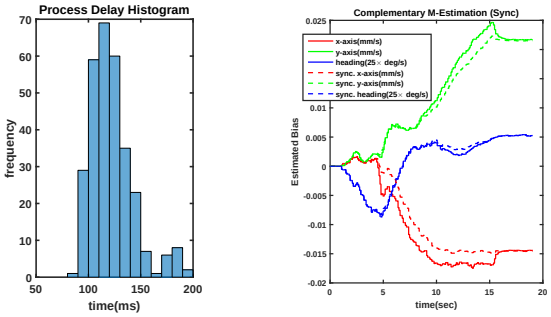
Fig. 3: Raw data and filters' outputs for a couple of experiments approaching the same target pose. Note the high noise affecting the k-means far away from the grasping neighborhood.

of the results. The methods are implemented on iMoro, a four-wheel-steering mobile manipulator. The robot is driven in closed-loop control. We use a fusion method, called K-mean, described in [10] as the reference, which has a finite memory of the most recent k values only motivated by speed and simplicity.

To compare the methods performance in a closed-loop navigation system, the robot receives a path defined by control points of Bezier curve. The results are overlaid in Fig. 3. Clearly, the K-means suffers from the high noise contributions inherent to the camera-based information at distances further away from the marker, resulting in oscillations. The complementary M-estimator, however, delivers smoother performance. Note the gap between the complementary M-estimator and the camera-based reference without the synchronization: the information from the camera contains time-dependent uncertainties, whose significance is best illustrated by the forward projected, synchronized black dots laying smoothly on top of the realized path in yellow (magnified in Fig. 3). This shows that much of the noise-like position uncertainties in the raw ODT data originates from the timings.

The following histogram in Fig. 4a shows the typical latencies when solving the pose of the robot. Next to the histogram in Fig. 4b are the biases for the estimated xy-coordinates and heading plotted. As can be noted, some part of the biases are proportional to the speed of the robot.The difference between estimated biases for before and after synchronization depends on the fast dynamic of the robot compared to the slowness of its ODT. Otherwise, the biases are affected by uncertainties in the extrinsic calibrations or changes in the robot's mechanical and geometrical properties.

Since the steering angles depend on the path curvature, any jump in the estimated pose oscillates the steering angles. As shown in Fig.5 the M-estimation performs three times lower steering actuator motion over the K-mean for a similar path. This is important since the oscillations are undesirable and can affect all of the measurements of all the sensors as well as actuator power consumption in general. It is also worth noting that this is more pronounced in nonholonomic robots due to higher costs of lateral changes in the body's position compared to holonomic robots.



(a) (b)

Fig. 4: Left: Histogram of latency variations for the ODT. Right: Estimated bias for each state with/without the synchronization. Note the smoother convergence.

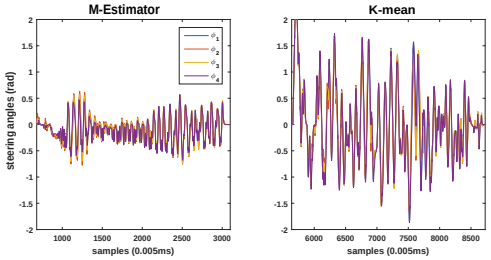


Fig. 5: Wheels' steering oscillations are reduced 3-fold over the K-mean reference.

5 Conclusions

In this paper, we have demonstrated explicit outlier rejection and real-time delay compensation for variable latency, event-based ODT as shown in Fig. 3. The outlier rejection was motivated by the high densities of range-dependent noise, though a rich set of visual features is used by the marker based ODT. Furthermore, as shown in Fig. 4a, the ODT runs at only a rate of some 8 frames per second. In more complicated visual scenes, the ODT process time can go as high as 2 seconds [17], which underlines the importance of the proposed timestamp-based, forward projection of the ODT information using the wheel odometery for synchronization purposes. These advances provide well-behaved feedback from the robot's motion control viewpoint, which is a common issue for robot's localization in any global coordinate and leads to more undesirable and power consuming oscillations in steerable robots. In Fig. 5, these oscillations were reduced to at least by a factor of 3.

As stated in Section 1, video demonstrations show competence of the method in integration of visual information for semi-unknown objects too. In this case, the range-dependent ODT noise is significantly higher than that here.

References

1. G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 237–267, 2002.
2. F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *J. of Intelligent and Robotic Systems*, vol. 53, no. 3, pp. 263–296, 2008.
3. F. Dalgleish, S. Tetlow, and R. Allwood, "Vision-based navigation of unmanned underwater vehicles: A survey. Part 2: Vision-based station-keeping and positioning," *J. of marine design and operations*, no. 8, pp. 13–19, 2005.
4. A. Cherubini and F. Chaumette, "Visual navigation of a mobile robot with laser-based collision avoidance," *Int. J. Robotics Research*, vol. 32, no. 2, pp. 189–205, 2013.
5. G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 87–100, 2007.
6. M. R. Walter, M. Antone, E. Chuangsuwanich, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, Y. Friedman, J. Glass, J. P. How, *et al.*, "A situationally aware voice-commandable robotic forklift working alongside people in unstructured outdoor environments," *Journal of Field Robotics*, vol. 32, no. 4, pp. 590–628, 2015.
7. Z. Durovic and B. Kovacevic, "Command-induced vibration analysis using input shaping principles," *IEEE Trans. Autom. Contr.*, vol. 44, no. 6, pp. 1292–1296, 1999.
8. D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, 2006.
9. X. Wang, N. Cui, and J. Guo, "Command-induced vibration analysis using input shaping principles," *IET Radar Sonar Navig.*, vol. 4, no. 1, pp. 134–141, 2010.
10. M. M. Aref, R. Ghabcheloo, and J. Mattila, "A macro-micro controller for pallet picking by an articulated-frame-steering hydraulic mobile machine," *IEEE Int. Conf. on Robotics and Automation (ICRA), Hong Kong*, pp. 6816–6822, 2014.
11. M. M. Aref, R. Ghabcheloo, A. Kolu, and J. Mattila, "A multistage controller with smooth switching for autonomous pallet picking," *IEEE Int. Conf. on Robotics and Automation (ICRA), Stockholm, Sweden*, pp. 2535–2542, 2016.
12. R. Oftadeh, R. Ghabcheloo, and J. Mattila, "A time-optimal bounded velocity path-following controller for generic wheeled mobile robots," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 676–683, 2015.
13. M. M. Aref, R. Ghabcheloo, A. Kolu, and J. Mattila, "Vision guided autonomous forklift," in *Advances in Robot Design and Intelligent Control Proceedings of the 25th Conf. on Robotics in Alpe-Adria-Danube Region (RAAD)*, Advances in Intelligent Systems and Computing, pp. 338–346, Springer, 2017.
14. R. G. Brown and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering*. New York, NY: Wiley, 1997.
15. S.-H. P. Won, W. W. Melek, and F. Golnaraghi, "A Kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1787–1798, 2010.
16. P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Comm. in Stat., A6*, pp. 813–827, 1977.
17. N. Wojke, F. Neuhaus, and D. Paulus, "Localization and pose estimation of textureless objects for autonomous exploration missions," in *IEEE Int. Conf. on Image Processing (ICIP)*, pp. 1304–1308, 2016.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-4190-2
ISSN 1459-2045