Carlos Pedro Maia da Costa Cândido

AUXILIARY MECHANISMS FOR TELEROBOTICS

Lisboa
2008

UNIVERSIDADE NOVA DE LISBOA

Faculdade de Ciências e Tecnologia

Departamento de Engenharia Electrotécnica

AUXILIARY MECHANISMS FOR TELEROBOTICS

(MECANISMOS AUXILIARES PARA TELEROBÓTICA)

Carlos Pedro Maia da Costa Cândido

Dissertação apresentada na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa para obtenção do grau de Mestre em Engenharia Electrotécnica e de Computadores.

Orientador: Prof. Luís Correia

Elemento de ligação à FCT/UNL: Prof. José Barata

Lisboa
2008

NEW UNIVERSITY OF LISBON

Faculty of Sciences and Technology

Electrical Engineering Department

AUXILIARY MECHANISMS FOR TELEROBOTICS

Carlos Pedro Maia da Costa Cândido

Dissertation presented at the Faculty of Sciences and Technology of the New University of Lisbon to obtain the Master degree in Electrical and Computer Engineering.

Supervisor: Prof. Luís Correia

Connection element to FCT/UNL: Prof. José Barata

Lisboa
2008

Acknowledgements

First I would like to thank my dissertation supervisor, Professor Luís Correia for his help and his constant support and motivation. I would like to thank all the former IntRoSys, S.A. R&D crew: Vasco Santos, Mário Salgueiro, Paulo Santos, Ivan Almeida, João Lisboa, Helder Monteiro and last but not least, to Pedro Santana for all his valuable support and motivation. None of this work would have been possible without the help of Prof. José Barata, who gave me the opportunity of working in this project. Special thanks also to the GruVA members for all the interesting discussions.

I also owe special words of appreciation to all of those who walked with me through all the lectures, sleepless nights and restless work since I joined the University, Baía, Feijão, Gonçalo, Greg, Lopes, Rute, Sérgio, Simão, Valente and all the others I may be forgetting.

None of this would have been possible without all the love and support provided by my parents Jaime and Isilda and my joyful sister, Mariana. This paragraph would not be complete without a word of appreciation to my amazing grandparents, Álvaro, Jaime and Celeste.

I would also like to underline the support provided by my friends Carol, Lorena, Paisana, Tânia, Hugo and Adolfo for your priceless friendship and for everything else words simply cannot describe.

Last but not least, I would like to also mention how important the love and support given by Lara and how important she was during the development of this dissertation.

What is important is to keep learning, to enjoy challenge, and to tolerate ambiguity.

In the end there are no certain answers.

Martina Horner

Sumário

Esta dissertação aborda o problema de teleoperar um robo móvel sobredeterminado a nível cinemático. Localizar e avaliar a condição de um robô são tarefas importantes num cenário de teleoperação. Apesar de estarem cada vez mais sofisticados, os robôs móveis ainda não são capazes de funcionar eficientemente sem intervenção humana.

O estado da arte em sistemas de localização ainda não é à prova de falhas. A odometria é robusta, contudo tende a gerar erros e torna-se complicada a sua utilização em robôs com cinemática sobredeterminada. Idealmente, todos os actuadores de tracção conduzem o robô para a mesma posição. Contudo, a rugosidade de alguns terrenos dificulta a tarefa aos sistemas de controlo cinemático que acabam por não conseguir manter as geometrias cinemáticas correctas. Isto significa que a odometria obtida a partir de cada actuador contribui de forma diferente para a estimativa de posição do robô. O modelo de odometria proposto considera individualmente a informação lida em cada roda e oferece um modelo de erros para posterior fusão sensorial.

A ligação entre o operador e um ambiente remoto está geralmente limitada a uma imagem de vídeo, o que reduz a capacidade do operador de avaliar o estado do robô e o ambiente em que este se encontra. Isto contribui para a sua desorientação e dificulta a compreensão do ambiente remoto. Esta dissertação aborda também o problema de controlar uma câmara de teleoperação, *pan-tilt*, oferecendo ao operador diversas alternativas de controlo expedito.

Resultados obtidos em ambientes reais e simulados confirmam as capacidades dos sistemas apresentados.

Abstract

This dissertation addresses the problem of improving the teleoperation experience of an over-constrained multiterrain mobile robot. Localising a robot and assessing its condition is cumbersome in a teleoperation scenario. Despite the fact that robots are becoming more sophisticated, they aren't still able to perform efficiently without human intervention.

Current state of the art localisation technology is not able to operate without failure. Odometry is robust, though it is error prone and complicated to use in over-constrained robots. Ideally, every actuator drives the robot to the same position, though, terrain roughness difficults the task of motion control systems which fail to guarantee perfect kinematic geometries. This means that each wheel contributes differently to the position estimation of the robot. The proposed odometry model considers the input provided by each wheel and provides an error model for further sensor fusion.

The operator's link to the remote environment is often influenced by a single video feed which decreases the operator's capability of assessing the robot's condition. This contributes to disorientation and complicates the comprehension of the remote environment. This dissertation tackles the problem of improving situation awareness in teleoperation by offering the user multiple alternatives to control a pan-tilt camera.

Results obtained in real and simulated environments demonstrate the capabilities of the presented systems.

Symbols and Notations

Symbol	Description
4WS	Four-Wheel-Steered
AKF	Augmented Kalman Filter
Az-El	Azimuth and Elevation
CCC	Cross Coupled Control
CMF	Camera Model Field
CMU	Cargenie Mellon University
CW	Clockwise
CCW	Counter-Clockwise
DGPS	Differential Global Positioning System
DOF	Degrees of Freedom
EKF	Extended Kalman Filter
ET	Encoder Trailer
FIDO	Field Integrated Design and Operations
FLEXnav	Fuzzy Logic and Expert rule-based navigation system
GPS	Global Positioning System
GruVA	Grupo de Vida Artificial (Artificial Life Group)
IPEC	Internal Position Error Correction
IRC	Instant Rotating Centre
JPL	Jet Propulsion Lab

Symbol	Description
LRV	Lunar Roving Vehicle
MDOF	Multiple Degree Of Freedom
PC	POSTECH CMU
PD	Proportional-Derivative
POI	Point of Interest
PTZ	Pan-Tilt-Zoom
RAVON	Robust Autonomous Vehicle for Offroad Navigation
S&R	Search and Rescue
SME	Small Medium Enterprise
STOC	Stereo On a Chip
UMBMark	University of Michigan Benchmark
E_O	Odometric orientation Error
E_C	Odometric contour Error
E_T	Odometric tracking error
l	Distance between the robot's front and rear axles
$2t$	Width of the robot's axle
TW	Distance between the kingping and the centre of wheel
δ_i	Steering angle of wheel i
d_i	Linear distance a wheel i travels between two iterations
$sgn\left(X\right)$	Signal of variable X
ρ_i	Turning radius of a wheel i
α_i	Angular distance travelled by a wheel i
r	Nominal radius of the wheels of the robot
u^w	3-DOF motion estimation of the robot
u_i^w	Robot's motion estimation induced by wheel i

Symbol	**Description**
ω_i	Weight of the contribution of wheel i for the global motion estimate
$\hat{\delta}_i$	Expected steering angle of wheel i
e_i	Difference between wheel i's current steering angle and the expected
$\delta_{<F,R,C>}$	Steering angle of the front and rear fictitious wheels
δ_{Ci}	Angle each wheel i induces in the fictitious wheel of the bicycle model
C_X	Input covariance matrix
F_X	Jacobian of the output variables of the odometry estimate
C_Y	Covariance of the estimate of the variation in the position of the robot
σ_A^2	Variance of variable A
σ_{AB}^2	Covariance between two variables A and B
k_{δ_i}	error constant of the steering angle of wheel i
k_{d_i}	error constant of the distance travelled by wheel i
ψ_{rov}	Heading angle of the rover
φ_{rov}	Pitch angle of the rover
θ_{rov}	Roll angle of the rover
P_{rov}	The rover's 4×4 posture matrix
$r_i(t)$	The desired Pan $(i = P)$ and Tilt $(i = T)$ angles of the camera
$p_i(t)$	The actual or estimated position of the camera's Pan $(i = P)$ and Tilt $(i = T)$ actuators
$u_i(t)$	The output of the camera controller for the Pan $(i = P)$ and Tilt $(i = T)$ actuators of the camera
$e_i(t)$	Reference error of the camera's Pan $(i = P)$ and Tilt $(i = T)$ actuators
K_{ip}	Proportional component of the camera's position controller
K_{id}	Derivative component of the camera's position controller
K_{ir}	Varying reference component of the camera's position controller

Symbol	**Description**
η	Azimuth between the world frame and the camera's line of sight
ε	Elevation between the world frame and the camera's line of sight
Ω	Matrix which defines the desired viewpoint (in Az-El)
S	Matrix which defines the values of the pan and tilt references to gaze at a certain Az-El
p	Position of the point of interest in the site frame when gazing at a GPS point
K	Constant which will affect the pan angle of the camera in the Assisted Navigation mode when the double-Ackerman geometry is being used
v_i	Velocity of wheel i

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation covers part of the author's work developed within the AMI-02 project which aimed at the development of a multi terrain service robot, the Ares robot. During this project the author was responsible for the development of all the design of the electrical installation of the several modules of AMI-02 project, hardware integration, the development of a Differential Global Positioning System (DGPS) interface, among others. This dissertation could never cover all the author's work developed during his involvement in this project, therefore, it covers part of the author's involvement in the last year of the AMI-02 project, focusing on the development of two mechanisms in which his involvement was fundamental and also with more relevance from the scientific point of view.

1.1 Motivation

The motivation behind this dissertation is the development of two mechanisms to support telerobotic activities, namely, the control system for a teleoperation camera and the odometry measurement mechanism to support the localisation of a mobile robot.

Getting mobile robots out of the semi-structured office and shop-floor environments has always been a goal of the robotics research community. In this context, substantial progress has been achieved in the last 10 years and with it, new interesting challenges are arising. The challenges of field robotics are enormous. Carlson and Murphy [Carlson and Murphy, 2005], go as far as to show that current state-of-the-art field robots are not able to operate without failures within an acceptable period of time.

Recovery of such failures often requires human intervention. Apart from this, human judgement displays a critical element of a robot's activity, specially in tasks that involve exploration, surveillance and reconnaissance.

Cameras are a common denominator in most teleoperation interfaces (for a summary on methods for interfacing humans and robots refer to [Fong and Thorpe, 2001]). The most common types of camera consist in spherical vision, fixed and pan-tilt cameras. Despite their ability to provide a panoramic view of the environment, spherical vision cameras are only able to offer low resolution [Schiehlen and Dickmanns, 1994] which makes them unsuitable for teleoperation purposes. Moreover, in what regards teleoperation, most of the peripheral image information is useless, thus it only contributes to overwhelm the operator with unnecessary information. On the other hand, fixed mounted cameras do not allow the operator to rapidly span its attention over the relevant aspects of a cluttered environment [Scholtz et al., 2004]. When provided with an adequate control system, pan-tilt cameras can allow the operator to freely and accurately inspect an environment, while simultaneously navigating the robot [Nielsen et al., 2007]. Hence, they are a better trade-off. Despite the advantages, several factors are important for a camera to be efficient: deficient camera placement, narrow field of view (*soda straw effect* [Voshell and Woods, 2005]), among others, can significantly harm the operator's conception of the environment, resulting in cognitive mistakes and disorientation [Hughes and Lewis, 2005] (i.e. reduced situation awareness).

A teleoperation task normally engages the teleoperator in the subtasks of navigation (i.e. moving the robot in the environment) and inspection (i.e. acquiring a viewpoint to a particular object) [Hughes et al., 2003]. Alternating between these two tasks, added to the aforementioned issues, robot teleoperation can be very demanding for the operator [Hughes and Lewis, 2005]. For example, in an uneven terrain, it requires the operator to constantly correct the camera's position to maintain visual contact with a desired target. This results in distracting the teleoperator from navigation, slowing down the task while inducing the teleoperator in error.

The complexity of this problem drastically increases when data latency is high (e.g. as in space exploration rovers). This problem is typically solved by providing the robot with some autonomy that can aid teleoperation [Powell et al., 2006]. Shared control of the viewpoint [Hughes, 2002], which consists in allowing an intermediate control of the viewpoint between the user and the rover, attentive navigation [Wernert and Hanson, 1999] and decoupled [Hughes et al., 2003] camera controls, are state of the art hypotheses to assist on teleoperation overcoming the problems associated with coupled camera controls [Hughes and Lewis, 2004]. Besides the importance of the aforementioned techniques, the camera control systems (e.g. position/speed control) to support its manipulation also display an important role in a teleoperation process.

Inspecting an environment through a camera can be quite confusing and it is therefore important to perform smooth movements with the goal of reducing image blur. For instance, latency (e.g. communications, processing, etc.) can be an obstacle in achieving such requirements. It is therefore necessary to develop controllers which achieve a balance between motion smoothness and motion accuracy (e.g. ability to track a reference provided by the user). Control systems are often crafted in order to cope with very specific and structured problems. The use of legacy systems (e.g. commercial pan-tilt cameras) instead of custom made tools, is being more and more recurrent in robotics, requiring control systems that are capable of adapting to the limitations such systems offer. For instance, it is often impossible to attach a controller directly to a legacy system's sensors/actuators which leads to the necessity of creating an interface to allow communication. This introduces limitations in the whole process (e.g. latency). Control systems must also be designed in order to cope with these cases.

Providing a mobile robot with an accurate localisation mechanism is also one step further in order to avoid failures in the field. For instance, if the localisation mechanism fails, the robot can not know where it is and it can be difficult to determine what to do next. The other side of the problem is that it is also difficult for an operator to understand where the robot is and therefore difficult to rescue the robot.

Several sensors and methods are known in order to localise a rover, though, most localisation sensors can not operate in every circumstance:

- DGPS systems can offer extremely accurate measurements in what concerns to global localisation (50 cm), though, they tend to fail or get extremely inaccurate while being used close to walls or under trees.
- Compasses can offer fairly good absolute measurements about orientation, though, they are easily affected by external magnetic fields.
- Gyroscopes can be extremely accurate but they have an associated drift and are heavily expensive.
- Visual odometry is extremely precise but it is extremely heavy to compute and its accuracy heavily degrades specially when the sun is low and facing directly the camera and in homogeneous environments, etc.

It is therefore important to have some redundancy with localisation sensors in a way that, when one or several devices fail, localisation is still available.

A basic, yet robust method to estimate a vehicle's motion consists in using knowledge about its wheels steering angle and induced velocities. This method is known as *odometry*. Odometry is the most widely used method for determining the momentary position of a mobile robot. Odometry provides good short-term accuracy, is inexpensive, and allows very high sampling rates. However, the idea behind odometry is the integration of incremental motion information over time, which inevitably leads to an unbounded accumulation of errors. Particularly, the accumulation of orientation errors causes large position errors which increase proportionally with the distance travelled by the robot. These errors increase on less-than-ideal ground conditions as odometry alone quickly accumulates large position and heading errors.

Despite such limitations, odometry is an important part of every robot localisation mechanism and it is used in almost all mobile robots, it can be easily fused with absolute position measurements to provide better and more reliable position estimation and it often happens that odometry is the only navigation information available (e.g. GPS dropout or lack of luminosity for visual odometry).

On this document a method for computing the odometry of an over-constrained (i.e. with more independent motors than degrees of freedom) mobile robot is proposed. Over-constrained mobile robots offer better mobility (the robot used in this dissertation offers four locomotion modes), though, due to the complex nature of their mechanical structure, odometric accuracy heavily degrades and it is therefore important to have an error model of the odometry estimate. Such error models are also necessary to fuse odometry with other localisation mechanisms.

1.2 Dissertation Statement

1.2.1 Problem Statement

As previously stated, the goal of this dissertation is to develop two mechanisms to support telerobotic activities.

First a control system for a teleoperation camera. As previously mentioned, mobile robots require human intervention when they fail to be autonomous or when they are supposed to be operated on missions like surveillance, scouting, etc. As it happens often, the utility of a video feed is overrated, resulting in the video feed being the single link between the operator and the remote environment. In the case of pan-tilt cameras it is common that the operator looses track of the position of the camera in relation with the robot (i.e. reduced situation awareness). This problem is often undesirable when the operator needs to track an object with the camera while simultaneously operating the robot, i.e. the operator is overwhelmed with the amount of commands he simultaneously has to send to the robot and the camera. Having this into consideration, the following requirements were established:

- Offer the operator more than one hypothesis of control of the camera;
- Assist the remote operator in understanding the condition of the robot;
- Allow the operator to abstract himself from the orientation of the camera.

The other issue this dissertation addresses is concerned with estimating the position of an over-constrained mobile robot recurring to wheel odometry. Over-constrained mobile robots have more actuators than degrees of freedom and it is therefore important to bear this in mind. It is possible that, in some cases, the actuators ”fight” against each other so it is necessary to develop a mechanism which takes this into consideration.

Odometry, due to its characteristics, is a widely used method for determining the position of a mobile robot. As in most localisation methods sensor fusion is used, it is necessary to have a minimum knowledge about the error of each method.

The following requirements were established for the odometry computation mechanism:

- Be able to cope with every locomotion method of the vehicle;
- Take advantage of redundant wheels;
- Have a coherent error model for further sensor fusion.

1.2.2 Solution Prospect

This dissertation presents two solutions, one for each of the presented problems.

In what concerns to the teleoperation camera, the control process is implemented from bottom to top. A position controller which is concerned with motion smoothness of the camera is implemented as well as a hardware architecture which will allow the operator to have an upper level control of the camera. This upper level control of the camera provides the operator with some pointing functionalities and automatic mechanisms to understand the condition of the robot. By assisting the operator with such functionalities, the user is able to offer more attention to more tasks inherent to a mission without degrading situation awareness.

Three techniques have been considered in order to assist the user in teleoperating the camera:

Pointing assistance: Manoeuvring a robot while simultaneously aligning a camera with an interest point is a complicated task, therefore, three mechanisms were developed for assisting the operator in the task of pointing the camera: (i) pointing in azimuth and elevation, (ii), pointing to a GPS point and (iii) pointing to the rover frame where the user specifies a position of the robot he wishes to look at, without having to be concerned with the position of the camera (i.e. its pan and tilt angles).

Assisted navigation: As the name indicates, this method facilitates the use of the camera when navigating the robot by suggesting the user the best viewpoint the camera can offer according to the desired trajectory to follow. This allows the user to focus on navigating and releasing him from the known problem of identifying the position of the camera relative to the robot and its displacement trajectory.

Self monitoring: This method offers the operator the possibility of identifying and analysing events in a rover in a more efficient way. As soon as an event is detected, the camera automatically points in the direction of the event. For instance, when a wheel of the robot gets stuck, the current in the traction motor rises in order to overcome the obstacle without the operator noticing it, this mechanisms leads the camera to automatically point at the wheel where the current rose to an abnormal level, allowing the user to assess the problem more efficiently.

In what concerns odometry, the problem of obtaining the odometry of an over-constrained vehicle is complex. These problems are usually tackled by exploring the fact that all the wheels are independent, this means that analysing the motion of different wheels in the same period of time, might lead to the obtention of different odometry results. The general technique to solve this problem consists in obtaining the best wheel or group of wheels according to a rule or set of rules which are empirically defined. Excluding the outliers might theorically be a good approach, though, in reality, an outlier wheel might harm the rover's desired motion, this is because wheels, by forcing an erroneous movement, end up fighting against the "correct" movement the remainder wheels are executing. In the model presented in this dissertation, all the wheels are taken into account and odometry is computed as a weighed average of the odometry computed for each wheel. These weights are computed according to the effect of the "rebel" wheels, the bigger the error induced by the wheel, smaller its weight will be in the weighed average of the vehicle's odometry.

1.3 Dissertation Outline

This chapter has provided the motivation behind this dissertation and outlined the objectives sought. The remainder of this document is organised as follows:

Chapter 2 reviews in the first section the state-of-the-art regarding odometry measurement mechanisms and methods for reducing odometry errors and in the second section, control techniques for teleoperation cameras are presented and discussed.

Chapter 3 presents the odometry model of the Ares robot.

Chapter 4 addresses the teleoperation camera control system. It first describes a controller for its pointing mechanism and in the remaining sections, the implemented functions to assist the operator on controlling the camera are explained.

Chapter 5 presents the experimental setups and corresponding results for the two mechanisms described on this dissertation.

Chapter 6 concludes this dissertation with a set of conclusions and contributions of this dissertation. Future research opportunities on the topics of this dissertation are also commented on this chapter.

1.4 Further readings

Some of the concepts covered in this dissertation have been published in [Santana et al., 2008a] and [Cândido et al., 2008]. The study which was elaborated in order to implement the odometry method presented in this dissertation was applied in a motion controller for compliant four-wheel-steering robots [Santana et al., 2006]. This dissertation is part of a project supported by the company IntRoSys, S.A.. This project aims at the development of affordable service robots for humanitarian demining. In order to find out more about this project refer to [Santos et al., 2007], [Santana et al., 2007], and [Santana et al., 2008a].

For further information on the work presented in this dissertation, feel free to contact the author: **carlos.p.candido[at]gmail.com**.

Chapter 2

State-of-the-Art

This chapter surveys the state-of-the-art on odometry error reduction methods and on camera control on mobile robots. Section 2.1 will foster the comprehension on the problem of odometric errors and techniques which allow the reduction of such errors. Section 2.2 makes an overview on techniques aimed at improving the usability of a teleoperation camera.

2.1 Odometry

This section introduces the reader to the problem of odometry and to the most relevant methods to improve odometry accuracy. The methods which allow for the improvement of odometry accuracy and their positive impact on localisation estimation mechanisms are well known. Nevertheless, a brief overview concerning odometry may endow us with a deeper understanding and characteristics of odometry measurement mechanisms.

2.1.1 Dead-Reckoning Sensors

In order to build an odometry system it is necessary to have a way of measuring the distances travelled by the traction motors as well as estimating the angular displacement of the steering motors. Since most mobile robots rely on some variation of wheeled locomotion, a basic understanding of sensors that accurately quantify angular position and velocity is an important

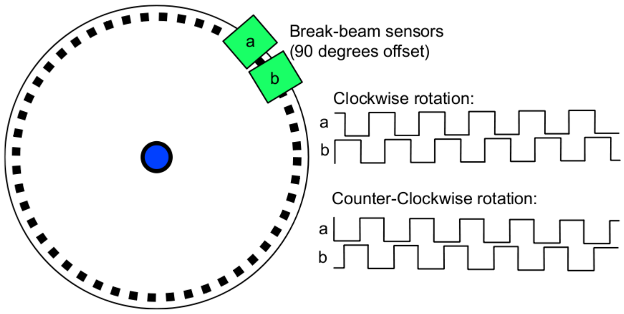


Figure 2.1: Quadrature phase encoder [Olson, 2006].

requisite to further discussion of odometry. There are a number of different types of rotational displacement and velocity sensors in use:

- Quadrature Phase Encoders (Optical). For a better understanding of such encoder, the mono phase encoder should be addressed first. In a mono phase encoder, a sensor is placed on the motor shaft or wheel such that when the shaft rotates, the circuitry generates alternating 1's and 0's. A simple way to do this is by attaching a perforated disk to the shaft, and using a breakbeam sensor to detect when a hole passes by. A monophase encoder can not determine the direction of motion. As an improvement upon mono phase encoders, quadrature phase encoders use two breakbeam sensors arranged so that they produce waveforms 90° out of phase. When the shaft rotates in one direction signal A "leads" signal B; when rotating in the opposite direction signal B "leads" signal A (see Fig. 2.1). With the correct controller, such encoders can be used to measure absolute positions. In mobile robot applications incremental optical encoders are the most popular.

- Tachometers. A tachometer is essentially a small motor attached to the output shaft of the main motor. It operates as a generator, where the output voltage is proportional to the angular velocity of the main motor.

- Potentiometers. A potentiometer is a three-terminal resistor with a sliding contact that forms an adjustable voltage divider [Breitfelder and Messina, 2000]. By changing the position of the sliding contact, different voltages are read and therefore, the position of the slider can be determined by reading them.

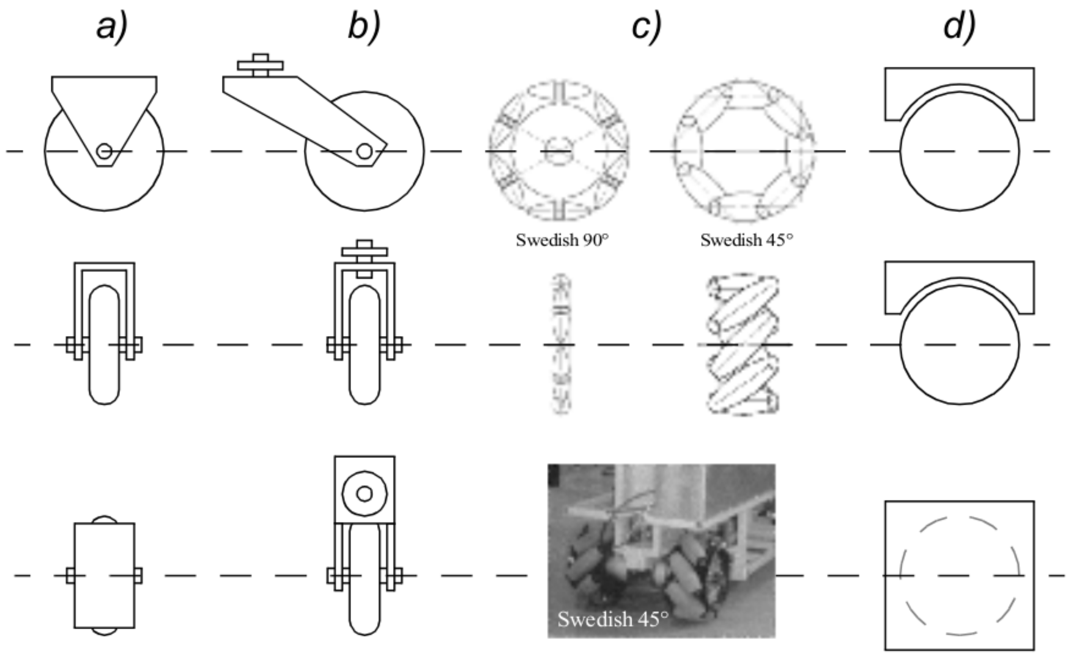


Figure 2.2: The four basic wheel types. (a) Standard wheel: two degrees of freedom; rotation around the (motorised) wheel axle and the contact point. (b) castor wheel: two degrees of freedom; rotation around an offset steering joint. (c) Swedish wheel: three degrees of freedom; rotation around the (motorised) wheel axle, around the rollers, and around the contact point. (d) Ball or spherical wheel: technically difficult to build [Siegwart and Nourbakhsh, 2004].

Other sensors exist, such as the brush encoders, synchros, resolvers as well as other type of encoders: magnetic, inductive or capacitive, though, because of their irrelevance on the scope of this project, they will not be addressed in this dissertation.

2.1.2 Wheel design

There are four major wheel classes, as shown in figure 2.2. They differ in their kinematics, and therefore the choice of wheel type has a large effect on the overall kinematics of a mobile robot [Siegwart and Nourbakhsh, 2004]. The standard wheel and the castor wheel have a primary axis of rotation and are thus highly directional. To move in a different direction, the wheel must be steered first along a vertical axis. The key difference between these two wheels is that the standard wheel can accomplish this steering motion with no side effects, as the centre of rotation passes through the contact patch with the ground, whereas the castor wheel rotates around an offset axis, causing a force to be imparted to the robot chassis during steering.

The Swedish wheel and the spherical wheel are both designs that are less constrained by directionality than the conventional standard wheel. The Swedish wheel functions as a normal wheel, but provides low resistance in another direction as well, sometimes perpendicular to the conventional direction. The small rollers attached around the circumference of the wheel are passive and the wheel's primary axis serves as the only actively powered joint. The key advantage of this design is that, although the wheel rotation is powered only along the one principal axis (through the axle), the wheel can kinematically move with very little friction along many possible trajectories.

The spherical wheel is a truly omnidirectional wheel, often designed so that it may be actively powered to spin along any direction. One mechanism for implementing this spherical design imitates the computer mouse, providing actively powered rollers that rest against the top surface of the sphere and impart rotational force.

2.1.3 Error Sources and Measurements

Odometry is based on the assumption that wheel revolutions can be translated into perfect displacements relative to the floor. This assumption is not totally correct. One simple example is related to wheel slippage: if one wheel steps on a slippery surface (e.g. thin sand), then the associated encoder would register wheel revolutions even though these revolutions did not translate themselves to displacement of the wheel, and thus, of the vehicle. Besides this case, there are several other causes for inaccuracies in the translation of wheel encoder readings into motion. Therefore, it is important to take these causes into account when trying to estimate the motion of a vehicle recurring to wheel odometry.

When trying to measure and improving odometry accuracy, it is important to make the distinction between systematic and non-systematic sources of error. In previous work, Borenstein and collaborators [Borenstein et al., 1996] investigated on possible sources of both types of errors. Systematic sources of error are inherent to the robot's kinematic or controller properties, regardless of the environment or the conditions in which the robot is operating. Non-systematic sources of error, on the other hand, are those which depend on the robot's environment. These two types of error affect mobile platforms differently and this is why it is important to distinguish between them. The remediation of both types of errors is different and most importantly, both types require different measuring techniques with the goal of obtaining comparable and useful data.

A common categorisation of odometry errors is as follows:

Systematic errors

- Unequal wheel diameters;
- Misalignment of wheels;
- Uncertainty about the effective wheelbase;
- Limited encoder resolution;
- Limited encoder sampling rate.

Non-Systematic errors

- Travel over uneven floors;
- Wheel slippage or wheel skid;
- External forces (e.g. interaction with external bodies);
- Internal forces;
- Variation of wheel contact with the floor.

Systematic errors can be reduced in conventional mobile robots by a careful mechanical design of the vehicle and by vehicle-specific calibration. However, systematic errors depend partially on changing factors (e.g. load distribution), therefore, they can not be fully eliminated.

On the other hand, non-systematic errors can neither be avoided or compensated for with conventional odometry, being often necessary to act directly over the robot's actuators in order to obtain better results, as we will see in the following section.

The gravest about systematic errors is the fact that they accumulate constantly, specially on most smooth indoor surfaces, where they heavily contribute to the odometry error drift when compared the non-systematic errors. However, on an outdoor environment, generally characterised by a rough surfaces with significant irregularities, non systematic errors are dominant.

Both systematic and non-systematic errors harm the estimate of the motion of a wheeled

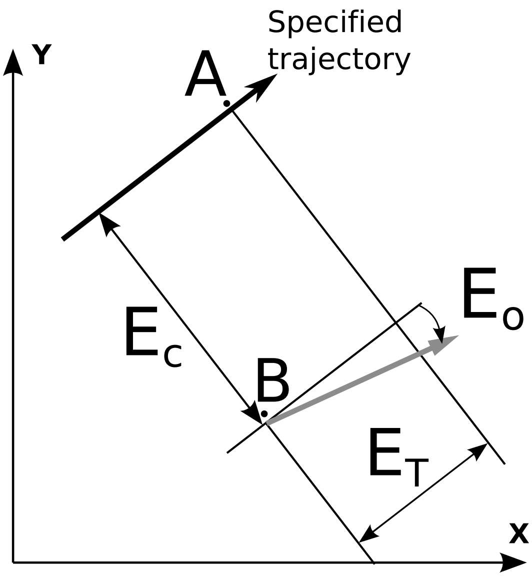


Figure 2.3: Motion error diagram (adapted from [Fan et al., 1995]).

vehicle and it is therefore important to characterise the impact of the sources of error on the motion of the vehicle. The errors on the estimation of odometry can be decomposed into orientation errors, contour errors and tracking errors [Fan et al., 1995]. As shown in figure 2.3, the vehicle is supposed to pass through point A, but due to the motion errors, it is actually passing through point B. The angular difference between the actual orientation (thick grey arrow) and the specified orientation (thick black arrow) defines the orientation error, E_O. The contour error E_C corresponds to the distance between the actual position B and the desired trajectory in the direction perpendicular to the direction of travel. The tracking error E_T is the distance between the actual position and the desired position in the direction of travel. In mobile robot motion control, the most significant motion error is concerned with orientation [Feng et al., 1993]. This is so because, once incurred, orientation errors grow without bound into lateral position errors.

2.1.4 Error Reduction Methods

The accuracy of odometric measurements in mobile robots partially depends on the kinematic design of the vehicles:

- Vehicles with a small wheelbase are more eager to orientation errors than vehicles with a larger wheelbase [Borenstein et al., 1996].

- Wheels used for odometry should be as thin as possible and not compressible. In practice, this design is not feasible for heavy duty vehicles because usually, the wheels used for odometry are also load-bearing drive wheels, which require a large ground contact surface [Borenstein et al., 1996].

- Castor wheels are likely to induce slippage when reversing direction, specially if they are supporting a significant portion of the overall weight [Carlisle, 1983].

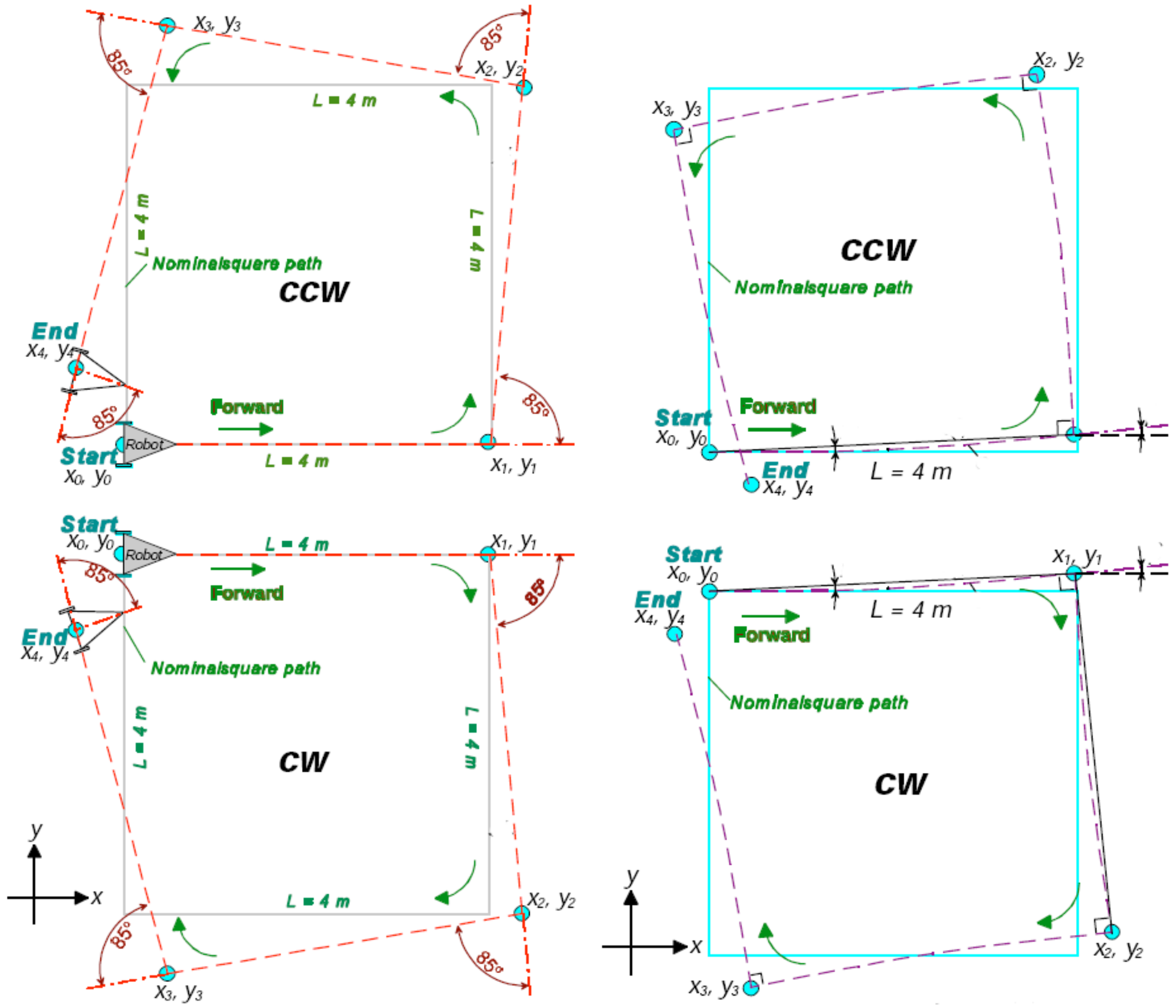
In the remainder of this section, specific methods for reducing both systematic and non-systematic odometry errors are presented.

Calibration techniques

The *University of Michigan Benchmark* (UMBmark) test [Borenstein and Feng, 1994] has been developed to calibrate for systematic errors of a mobile robot with a differential drive. This test consists in a set of runs in which the robot is programmed to traverse the four legs of a 4×4 m square path. Ideally, the path returns the vehicle to its starting point, though, due to odometry and controller errors, this is not true. The absolute position of the robot is measured before and after each run. These measurements are then compared to the position and orientation of the vehicle obtained from odometry. The experiment, as well as the measurements of the return positions associated with it are performed five times in clockwise (CW) and counter-clockwise (CCW) direction.

By performing this test, it is possible to identify two error characteristics, called Type A and Type B, which represent odometry errors in orientation. Type A errors are defined as orientation errors which reduce or increase the total amount of rotation of the robot in both CW and CCW direction. Type B errors are defined as orientation errors which reduce or increase the total amount of rotation of the robot during the square path experiment in one direction, but increases or reduces the rotation amount when going in the opposite direction.

Figure 2.4(a) displays the contribution of errors of Type A, which are mainly caused by the uncertainty about the effective wheelbase. In figure 2.4(b) the contribution of Type B errors



(a) (b)

Figure 2.4: Type A 2.4(a) and Type B 2.4(b) errors in clockwise and counter-clockwise direction [Borenstein and Feng, 1994].

is displayed. Such errors are mostly caused by the ratio between wheel diameters, which in normal conditions, should be equal to 1, resulting in over or under steering situations.

By analysing the behaviour of the robot during the UMBmark test, it is then possible to draw conclusions about which variables to adjust in order to improve odometry results. More information about these adjustments can be found on [Borenstein and Feng, 1994].

Results presented in [Borenstein and Feng, 1995] show that it is possible to dramatically improve odometry accuracy by compensating the effect of systematic errors. The main disadvantage relies on the fact that it requires a well controlled environment (flat surface) and changing the testing area may result in inaccurate results as this method does not allow the compensation of non-systematic errors. Another disadvantage is that it does not allow a robot to calibrate online, which is undesired for vehicles which need to operate for large periods of time and thus, eager to suffer unpredictable mechanical wear, generating new types of systematic errors during their lifetime.

In [Doh et al., 2003], the *POSTECH CMU* (PC) method is suggested as a procedure to calibrate an odometry system. The main idea behind this method consists in moving the robot in forward direction and then back along the same trajectory, followed by an obtention of the error parameters that minimise the error between both trajectories. This method is compared with the UMBmark and the results revealed to be similar, slightly in favour of the PC method. It is cumbersome to remember that, despite the similarity, small differences in error parameters induce large error in case of a long distance navigation. The main disadvantage when comparing this method with the UMBmark relies on the fact that it requires the vehicle to travel a straight line never inferior to $150\,\mathrm{m}$ in order to estimate the error parameters, while with the UMBmark a square with a $16\,\mathrm{m}$ perimeter is enough.

The main advantage of this method is that it provides a general error parameters and a three parameters covariance matrix estimation (typical in wheel odometry), thus, it can be used for all types of wheel odometry.

A less time consuming calibration procedure to compensate systematic errors is proposed in [Goel et al., 1999]. This method is more empirical than the previous one. With the robot suspended in the air, the actual velocities of the wheels are measured recurring to a tachometer (more precise) and are afterwards compared to the measurements taken from the encoders. The comparison between both the sources allows the establishment of a relation between the velocities returned by the encoders and by the tachometer and theoretically, it allows to compensate for the errors generated by the use of the encoders. As previously mentioned in section 2.1.3, the most significant motion error is related with orientation. In this case, no particular calibration procedure was taken for calibrating the yaw rate; in fact, the calibration of the yaw rate was done empirically. It was found that when the encoders predicted a $200\,^{\circ}$ turn, the robot actually turned $180\,^{\circ}$.

One of the main pitfalls of the wheel speed calibration relies precisely in the fact that it was done with the robot suspended in the air. After putting the robot in the ground, wheels immediately compress due to the weight of the robot which immediately affects the wheel geometry and therefore, compromises the calibration procedure. Once again, this method needs to be done offline. Another issue relies on the fact that it is relatively easy to calibrate a small vehicle (e.g. the Pioneer AT), though, when using a bigger and heavier vehicle, such calibration procedure dramatically increases in complexity.

It is known that robots change their physical properties over time. In mobile robotics, wear and tear can change the diameter of wheels, loosen belts, etc. Therefore it is important not to waste much time calibrating a mobile robot while it could actually be in use. Recently, some work has been carried out in what concerns to calibrate the odometry of a robot during operation [Larsen et al., 1998, Martinelli and Siegwart, 2003, Caltabiano et al., 2004] by recurring to additional (e.g. unrelated to odometry) sensors. In the three mentioned methods, an augmented Kalman filter (AKF) is used in order to estimate the configuration of the robot as well as the parameters characterising the systematic odometry errors. This filter uses encoder readings as inputs and the measurements coming from other sensors as observations. The main difference between the aforementioned works relies on the sensor used for observation: In [Larsen et al., 1998] a vision sensor is used, in [Martinelli and Siegwart, 2003] a laser rangefinder is adopted and in [Caltabiano et al., 2004] a DGPS system is used, being the only one validated in an outdoor environment.

These methods are particularly useful as they allow to change the robot's parameters (e.g. changing the wheels according to different situations) without being necessary to change its control variables as they are automatically estimated and adjusted during operation. These algorithms also allow skipping the calibration phase of offline calibration methods. Besides, they are more suitable for unstructured environments. In dynamic environments, static solutions are always temporary, therefore offline calibration can not be an adequate alternative for improving odometry accuracy of outdoor mobile robots.

In [Caltabiano et al., 2004], a quantitative comparison is made with the UMBmark and it is possible to observe that the results are more satisfactory by estimating the error parameters recurring to an external sensor, encoder readings and an AKF.

Some work has also been carried out on odometry errors from a theoretical point of view [Wang, 1988, Chong and Kleeman, 1997]. The non-systematic errors are analysed and the co-variance matrix Q for a given robot's trajectory is computed. These theoric models of odometry errors are important as they allow a proper integration with other localisation mechanisms using sensor fusion techniques (e.g. Kalman filters).

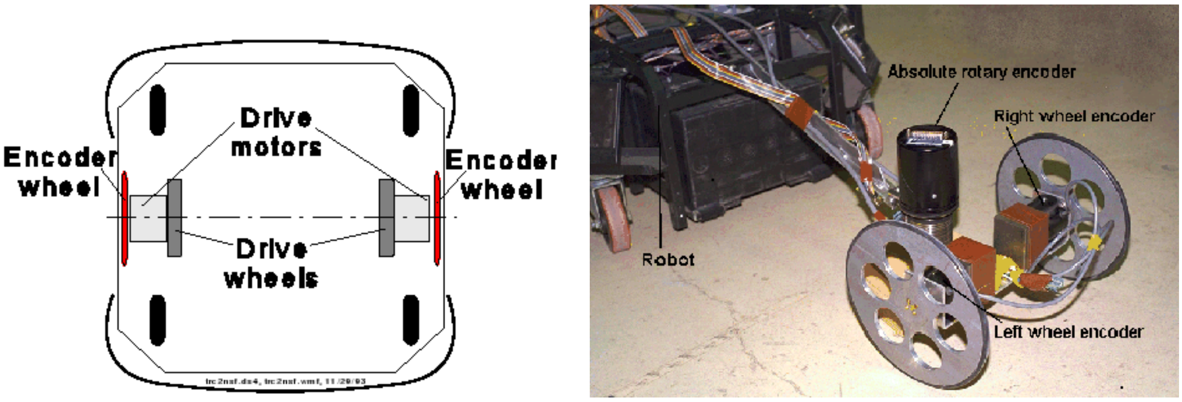
Auxiliary Wheels and the Encoder Trailer

It is generally possible to improve wheel odometry accuracy by adding a pair of thin edge, non-load-bearing encoder wheels. Since these wheels are not used for power transmission, they can be as thin as possible and use only a thin layer of rubber to avoid slippage.

This method was first proposed by [Hongo et al., 1987]. The goal consisted in improving the accuracy of a large differential drive mobile robot which weighed around $350\,\mathrm{kg}$. Tests with this method reported that, after travelling a distance of $50\,\mathrm{m}$, a position error inferior to $200\,\mathrm{mm}$ was detected which is a reasonable value for a vehicle of such dimensions.

An alternative approach is the Encoder Trailer (ET), proposed in [Fan et al., 1995]. The ET is a small two-wheel trailer that drags behind the main vehicle. Mounted in each wheel is an optical incremental encoder. A rotary joint allows the trailer to rotate horizontally around a fixed point behind the pulling vehicle. The trailer can rotate around this joint without limitation due to a slip ring that provides the electrical connections to the wheel encoders. When the pulling vehicle moves backwards or traverses uneven grounds or obstacles, the ET retracts. The argument for this part-time deployment of the encoder trailer is that in many applications, robots may travel *mostly* on reasonably smooth concrete floors and that it would thus benefit *most of the time* from the encoder trailer's odometry.

Wrapping up, the main idea of the ET consists in performing odometric measurements whenever the ground characteristics allow it. This is at the same time a major drawback, as it ignores two of the main reasons for using odometry: it is functional most of the time and it is



(a) (b)

Figure 2.5: Figure 2.5(a) shows a rough schematic of the implementation of the auxiliary wheels method [Borenstein et al., 1996]. In Figure 2.5(b) an encoder trailer, developed at the University of Michigan is shown [Borenstein, 1994].

important for redundancy on a localisation system with several mechanisms (e.g. GPS, visual odometry, etc.). The main advantage of this method is that it can be attached to a vehicle with no means of providing odometry measurements and instantly providing it with such ability.

Both the presented methods are unsuitable for outdoor environments, as their working principle relies on the fact that surfaces are usually flat and very well structured.

Internal Position Error Correction

A less common approach consists in using multiple robots to reference each other. A method is proposed in [Sugiyama, 1993], where two robots mutually measure their positions. When one of the robots moves to another place, the other remains still, observes the motion, and determines the first robot's new position. At any time one robot localises itself with reference to a fixed object: the standing robot. In [Kurazume et al., 1994], the results of a simulation with a similar approach are presented. Such approach uses multiple cooperating mobile robots divided into two groups. When the members of group A move, the members of group B remain stationary and provide positioning beacons that the group A robots can use for absolute positioning. After a while, group A robots stand still and group B robots move.

Though the aforementioned methods are not exclusively based on odometry, they inspired the development of the *Internal Position Error Correction* (IPEC) method [Borenstein, 1995b,

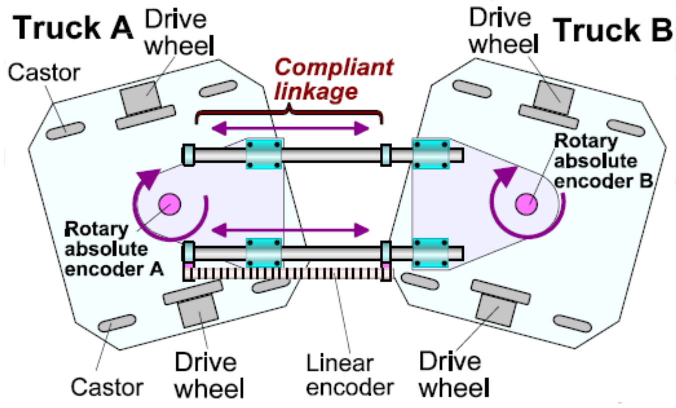


Figure 2.6: Essential components of the OmniMate [Borenstein, 1995a].

Borenstein, 1998]. This method allows two mobile robots to mutually correct their odometry errors, even while both robots are in continuous, fast motion, which is the great innovation relative to the methods presented above. To implement this method, it is required that both robots can measure their relative distance and bearing continuously and accurately. In order to achieve such requirement, the OmniMate robot was developed (See Fig. 2.6). This mobile robot consists in two differential-drive mobile robots (called ”trucks”), which are physically connected through a compliant linkage. In addition to the differential drive's encoders, this platform has one linear and two rotary *internal* encoders, which allow measurement of the relative distance and bearing between the two trucks. During operation, both mobile robots perform conventional dead-reckoning with their wheel encoders. In addition, the IPEC method uses information from the internal encoders to detect and correct odometry errors as soon as they occur.

The main advantage of this method, when compared to the localisation methods proposed in [Sugiyama, 1993] and [Kurazume et al., 1994], relies on the fact that it allows both robots (trucks) to mutually reference each other in continuous motion, overcoming the main drawback of the stop-and go effect of the initially proposed methods.

Tests with this method revealed that this method is capable of reducing most of both systematic and non-systematic errors efficiently. Though, due to the unique mechanical configuration required to use the IPEC method, other sources of errors are introduced (provenient from the *internal* encoder measurements). These errors can be reduced, though, this requires an accurate calibration of the vehicle.

One of the main drawbacks of this method, is the already mentioned mechanical characteristics a vehicle requires in order to use it. The sensorial apparatus required is not suitable for a robot with dimension restrictions. For instance, the OmniMate robot, weighs around 285 kg and is unable to cope with outdoor environments

Redundant Encoders

Rocky or Field Integrated Design and Operations (FIDO) type rovers have six drive wheels equipped with encoders. Since usually one or two encoders are needed for odometry (depending on the rover's locomotion mode), the six encoders on these vehicles are redundant. With redundant encoders it is possible to determine which encoder or set of encoders produce the more accurate readings in a given sample interval. Several methods exist to discriminate which wheels should be used for odometry measurements [Ojeda and Borenstein, 2004]:

The fewest pulses method: This method consists in the application of a simple rule: "*Use only the data from the encoder-pair that provided the least number of pulses during the last sampling interval.* The rationale behind this approach is that for most error-causing conditions encoders "over-count". That is, they provide more pulses than what corresponds to the actual linear distance travelled, when they encounter wheel slips or floor irregularities.

At low speeds and while driving at low traction surfaces (e.g. sand), it is possible to observe a significant wheel skid (instead of slippage), therefore, this method is ineffective and even counter-productive in some situations.

The Expert Rule method: This method aims at reducing odometry errors by differentiating between different situations recurring to a rule-based system and by offering a course of action for each of them. This method assumes that there is no absolutely correct solution, but it hypothesises that the rules provide an equal or better solution than the "Baseline Rule" which consists in calculating separately the readings of the encoders on the left and on the right side of a vehicle. In order to improve performance beyond the baseline rule, two sets of expert rules are applied during each sampling interval:

- Expert Rule # 1 (Average + Smallest): If the front and rear encoder on either side of the robot produce a similar number of encoder pulses in a sampling interval, the number of encoder pulses reported by both encoders on that side is assumed to be correct. In this case, the displacement of that side of the robot consists in the arithmetic average of the front and rear wheel encoder reading (identically to the baseline rule). Otherwise, if the difference in encoder pulses is greater than a threshold, the wheel with greater number of encoder pulses is assumed to be slipping and the wheel with lower count is selected, providing the readings for that sampling interval (it consists in the expert rule formulation of the fewest pulses method).

- Expert Rule # 2 (Average + Angle + Smallest): After applying rule # 1 and if only one side of the robot had correct encoder readings, the linear displacement on the correct side should be computed from the encoders, while the displacement on the side suspected of slippage should be estimated recurring to the angular velocity measured by an onboard gyro.

These methods show that it is important to see odometry as more than a theoretical problem. Tests with this method show that these rules are very effective in reducing odometry errors, particularly in errors caused by wheel slippages, which is the predominant source of error in outdoor environments, though, using heuristics to solve the problem of odometry errors, despite the fact that it shows good results, does not exclude the need for a theoretical approach. No theoretical models of the errors are presented which makes it impossible to use such models in an Extended Kalman Filter (EKF) for example.

Odometry error reduction by control approaches

In over-constrained vehicles the actual wheels and steering angles of the wheels have to match perfectly. Small discrepancies end up resulting in wheels fighting against each other and unavoidable odometry errors. Several methods exist in order to reduce such effects and associated wheel slippage in over-constrained mobile robots which will be briefly explained in the following lines.

Cross Coupled Control: This method was initially developed by Borenstein and Koren [Borenstein and Koren, 1987] and later refined by Feng [Feng et al., 1993]. It was initially designed to control the wheels of an over constrained differential drive robot. The Cross-Coupled Control (CCC) method continuously compares the actual travelled distance from the left and right wheel of a vehicle and issues corrective commands to the motors to slow down the motors which are faster and to speed up the slower motors. The overall effect is that the velocities of the left and right wheels of a robot are matched more tightly even in the presence of internal and external disturbances.

Another benefit of the CCC, though not odometry related, should be mentioned. In suspensionless robots, it often happens that one wheel looses contact with the ground while the vehicle traverses irregular surfaces. If each wheel is independently powered by a motor, the off-ground wheel and motor do not contribute with torque to displace the vehicle in the desired direction, which ends up in an unavoidable vehicle stall. The CCC partially overcomes this problem by increasing the speed of the "ground" wheel in order to equalise the speed of the two cross-coupled wheels on that side.

This method does not contemplate steering wheels which makes it unable to cope with some locomotion modes (e.g. Ackerman steering) unless another control layer is introduced in the system. A block diagram of the CCC is presented in Fig. 2.7.

Wheel odometry of the FIDO Rover: The FIDO rover is a vehicle with 6 independently driven and steered wheels and it is provided with a suspension that allows it to displace over rough terrains. As the rover drives and the suspension moulds itself to the topography of the terrain, each of the 6 wheels experience different forces which may cause wheels to "fight" against each other, resulting in increased power requirements, undesired wheel slippage/skid, among other issues. To minimise this effect a velocity synchronisation algorithm [Baumgartner et al., 2001] was developed for the FIDO rover which also enables odometry error reduction.

The first step of this algorithm consists in outlier detection that determines which wheels

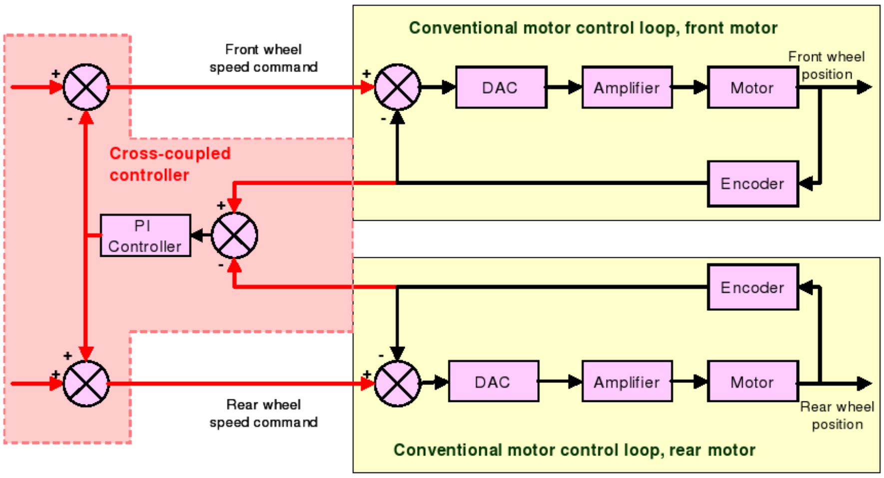


Figure 2.7: Block diagram of a Cross-Coupled Control loop, here shown for the two left-hand wheels of a 4WD differential drive robot [Ojeda and Borenstein, 2004].

of the rover are deviating significantly from the nominal velocity. All the wheels are given a score and after knowing the score each wheel is given, an average travelled wheel distance is voted and the required change in the commanded nominal velocity is computed. Finally, the resulting voted wheel distance is utilised in the odometry equations of the FIDO, as the means for determining the rover's position. This method of observing the behaviour of all wheels in order to build the estimate is interesting, though, excluding a wheel which is "fighting" against the movement of the remaining wheels may not be a good approach for the computation of odometry. This outlier wheel will always affect the motion of the vehicle and therefore will introduce errors (which won't be accounted for) on the vehicle's trajectory estimation.

Experiments with this control method revealed a slippage minimisation. In what concerns to odometry, this method is similar to the fewest pulses method [Ojeda and Borenstein, 2004], with the main difference that it does not discriminate the left or the right side of the vehicle when it is necessary to determine the outliers which intuitively results in smoother and more accurate results.

Biologically Inspired Locomotion Control: In the context of the AMI-02 project, we gave a step further and developed a controller which aims at maintaining a correct posture of the wheels. In [Santana et al., 2006] a novel behaviour-based approach for the locomotion control of the Ares robot was proposed. In such approach, each wheel has an independent controller composed of a set of behaviours (i.e. units linking perception to action) generating force vectors according to a given criterion. In order to handle exceptional situations, an enhanced version of the method was proposed in [Santana et al., 2008a].

A reference control behaviour seeks to steer each wheel towards the angle that allows the robot to turn towards the desired direction. The transient control behaviour adapts the output of the reference control behaviour so as to maintain the correct locomotion geometry. Upon reaching an upper threshold of the current in the wheel steering actuator, the stiffness control behaviour becomes active and it suppresses the output resulting from the fusion of both reference control and transient control.

As soon as the stiffness control behaviour becomes active, the wheel stops, and as a result the current applied to the steering motor drops. An active mechanism was implemented to confirm the stiffness control behaviour. When active, it suppresses other behaviours output and asks the actuator to turn in the direction it was turning before the activation of the stiffness control behaviour for a while, and then to turn in the opposite direction for the same amount of time. If the obstacle is still present, the current will rise and the stiffness control behaviour will trigger. Otherwise the wheel will turn in one direction, then in the opposite one, staying roughly in the same place before the stiffness confirmation behaviour was activated. Then, the stiffness control behaviour will timeout and the operations are resumed normally. This swing behaviour of the wheel does not induce other wheels to follow it in order to maintain the correct geometry.

This locomotion control is focused on maintaining a correct geometry of the robot during motion by making the robot compliant with the terrain where it is deployed, therefore, slippage, skidding, internal and external forces are avoided which aims at improving odometry results.

Discussion

In this section we have analysed previous odometry reduction methods. Offline calibration techniques are unable to cope with dynamic environments and besides, robots change their physical properties over time, requiring periodic calibrations. In order to solve this problem, online calibration methods were developed. Though, online calibration requires the presence of additional sensors, which may not be feasible in some applications. The use of auxiliary wheels or the encoder trailer showed good results but are unable to cope with the demands of rough outdoor environments. Other methods like the IPEC require additional robots and a complicated sensorial apparatus. The presented methods based on heuristics which approach the problem by considering the encoders to be redundant showed good results fighting non-systematic sources of error, which are the biggest source of errors in unstructured environments though, they lack a theoretical approach of the problem which enables the possibility of mathematical treatment of the results. Besides, these methods also tend to consider some wheels as outliers when in fact, they affect the motion of the vehicle, therefore, every wheel should be taken into account when developing the odometry model. Developing control approaches bearing in mind the reduction of odometry errors also proved to be fruitful in the reduction of odometry errors.

2.2 Camera Control for Teleoperation

Video streams are often the only perceptual link between the human and a remote vehicle. A teleoperation task normally engages the teleoperator in the subtasks of navigation and inspection. Switching between the aforementioned subtasks complicates the teleoperation task by reducing situational awareness. The tools and mechanisms provided to the teleoperator for manipulating the video stream (e.g. manipulating a camera) will have a dramatic influence on the teleoperation experience, therefore, it is important to invest some time in studying camera configurations and control strategies which aim at reducing the disorientation problems.

2.2.1 Robotic Viewpoint Control Issues

Before addressing the state of the art in improving the control of a teleoperation camera some elements should be addressed first. In [Hughes and Lewis, 2005] some key factors are identified as being relevant in the design of a vision platform:

Control Mapping Strategies

Handling a Multiple Degree of Freedom (MDOF) device can be overwhelming for an average user, specially when most of the familiar input devices (e.g. joysticks, mice, etc.) do not offer enough inputs in order to fulfil the degrees of freedom such devices can offer. This leads to several alternative strategies for control mappings.

- Overloading: Extra degrees of freedom are achieved by modal operation of the device.
- Constraining: Consists in limiting the movement of the viewpoint to certain operations. A common constraint consists in eliminating the roll in the viewpoint of a ground vehicle.
- Coupling: This approach consists in binding one or more attributes of the viewpoint to the state of the others.
- Offloading: This method consists in delegating control of certain operations to an external source, which can be a collaborative operator or an autonomous controller.

Camera configuration strategies

In [McGovern, 1993] several camera configuration strategies are proposed and deeper analysis on this topic is presented in [Hughes et al., 2003]. Three camera configuration strategies are known:

- Coupled Camera Controls. Also known as *gaze-directed steering* [Bowman et al., 1997], such configuration implies the camera orientation to be fixed, hence, the gaze direction is directly dependent on the direction of travel. Such coupled camera controls offer a strong bias towards navigation tasks in detriment of inspection. In order to navigate, the operator only needs to be concerned with the orientation and the travelling velocity of the robot, while inspecting an object is much more difficult. For example, in order to look at a specific object, it might be necessary to stop the robot, pivot the robot to acquire a good perspective of the object, and then reacquire the initial trajectory to proceed with the navigation task. This type of configuration ends up in excessive maneuvering of the robot, hence increasing the probability of getting the robot stuck. Another issue of this configuration is the fact that it is not energetically efficient. This is due to the amount of energy which is wasted to repeatedly pivot the robot in order to acquire visual contact with a desired object.

- Decoupled Camera Controls. This configuration allows the camera to be controlled independently of the direction of travel. Though being apparently useful and senseful, the extra degrees of freedom offered to the controller increase the complexity of the teleoperation task. Decoupling the camera facilitates inspection by allowing the operator to keep interesting objects in view, though, navigation becomes more difficult.

- Multiple Cameras. The prospect of applying multiple cameras on a robotic system provides a vast set of hypothesis: stereo vision, marsupial teams of robots, where a second robot provides an exocentric view of the main robot, which can be particularly useful for disambiguating obstacles that may have blocked the first robot's movement, providing different perspectives of the environment around a single robot, etc.

Selecting an appropriate camera configuration strategy requires an analysis of the situations in which a robotic camera system is to be used. For instance, coupled camera controls are more suitable for situations where navigation is more important than inspection. Using decoupled camera controls on the other hand has the reverse effect, i.e. inspection is more important and navigation less important. Theorically, a multiple camera approach may be used to solve the aforementioned problem of simultaneously navigating and inspecting by using a fixed camera for teleoperation purposes and a camera with more degrees of freedom for inspection [Hughes and Lewis, 2004], though, using more than one camera inevitably results in a larger occupation of bandwith for image transmitting. Bandwith is a precious resource in teleoperation activities and, increasing teleoperation experience at the cost of bandwith might not be a good trade-off, though, technological advances suggest that this problem can be overcome in short term. Another problem concerned with the use of multiple cameras resides on the fact that, despite offering the possibility of having multiple perspectives of an environment, it may require one monitor per camera and therefore overwhelm an operator with information, therefore, in order to obtain the best performance of such type of configuration, it may require more than one operator to execute a task. A camera which allows decoupling its gazing direction regardless of the vehicle's position (e.g. a Pan-Tilt-Zoom (PTZ) camera), not only it is similar to the behaviour of the humans (as they are able to pan and tilt their head) and hence, more intuitive, but with an appropriate control system as a backbone, despite the fact that it is more suitable for inspection purposes, it can also be extremely useful for an operator on a navigation task.

Camera Posture Assessment

When gaze-travel decoupling is permitted, situational awareness may degrade if the operator cannot quickly assess the status of the viewpoint. Several methods exist in order to assist the operator in what concerns to spatial cognition, as the two-handed compass which allows the viewer to detect misalignment between the orientation of the robot and the orientation of the camera, or by using peripheral fixed references, i.e. using the robot's body.

2.2.2 Improving the Control of a Teleoperation Camera

Offloading some of the viewpoint controls to an automated system can mandate effective navigation strategies while simultaneously reducing the control burden. The goal consists in delegating the responsibility of accomplishing certain tasks to a machine. While the notion of a fully autonomous entity replacing human presence is appealing, human observation and supervision are critical elements of robotic activity.

This section covers the state of the art in what concerns to viewpoint control issues, focusing on task automation techniques.

Attentive Interaction Techniques

The attentive navigation technique [Hughes and Lewis, 2000] is based on previous work presented in [Hanson and Wernert, 1997] and [Hanson et al., 1999]. The navigation space of a display is divided into a constraint space, which is a set of positional values a viewpoint can assume and a Camera Model Field (CMF), which is a collection of vectors that provide ”ideal” information for every point on the constraint surface. Attentive navigation proposes storing two vectors at each sampled point: the ideal gaze direction and the ideal ”next motion” direction. The ideal gaze direction will be used to provide a suggested orientation for the viewpoint; the direction that the virtual camera should be facing given a certain location. The ideal motion vector can be used to suggest the next positional value as the virtual camera moves through the viewpoints. Thus, Attentive Navigation serves as a recommendation engine capable of instructing the viewer to the ideal viewing orientation as well as the suggested motion from the current position.

Another attentive interaction technique is the attentive camera [Hughes and Lewis, 2000]. This method is similar to the one described in the previous paragraph, but in this case, the viewer is allowed to freely explore locations in an environment. Here, the system suggests an optimal viewpoint corresponding to the viewer's position. This is accomplished by augmenting the environment with a collection of optimal viewing vectors. In figure 2.8 a sample environment map with the ideal gaze vectors overlaid is shown. In such example, two points of interest are present on opposite sides of a room that capture nearby vectors. The ideal gaze vectors can be interpolated to ensure smooth transitions as the viewer moves through the environment. The attentive camera technique is characterised by the system continuously aligning the viewpoint with the ideal gaze vector as the viewer moves through the environment.

The attentive camera technique is effective at focusing the user's attention on significant elements in the environment, maximising their exposure, and increasing the viewer's perception.

In [Thorndyke and Hayes-Roth, 1982], it is argued that it is important for an operator to have different perspectives of a single point of interest to ease its identification. This method is of great help regarding this issue, as, instead of merely passing in and out of the visual field, an object or a point of interest becomes the focal point of the scene. As the user walks/drives by, this has the effect of the environment rotating around the object, naturally increasing the number of different perspectives.

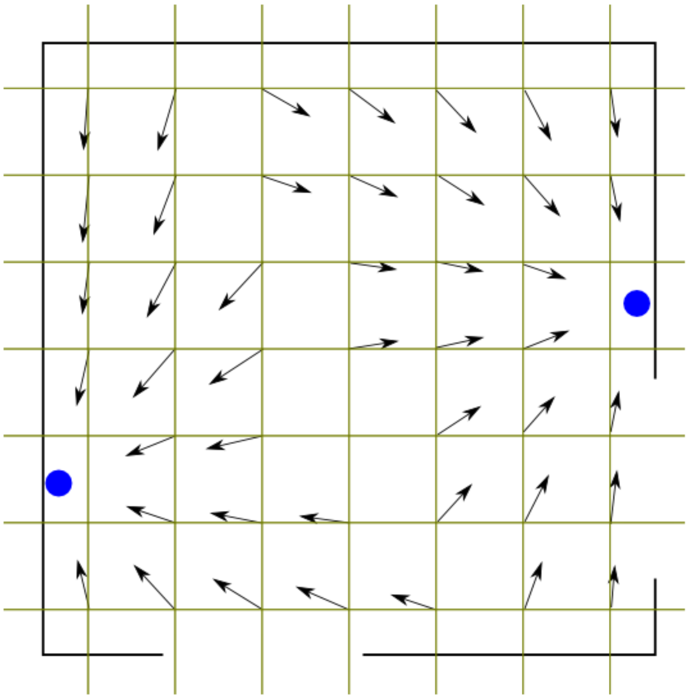


Figure 2.8: Ideal gaze vectors. Image adapted from [Hughes and Lewis, 2002].

Another technique is known as the Attentive Flashlight [Hughes and Lewis, 2002]. In this case, the ideal gaze vector is used to compute the direction in which to shine a spotlight. As the viewer wanders through the environment, the flashlight fixates on objects of interest. Initial tests revealed that it is useful for the viewer to keep track of the spotlight, therefore, if the object of interest is outside the viewer's field of vision, the spotlight is set to remain in the peripheral edge of the image. This alerts the viewer to relevant information which is currently outside of the field of view. Figure 2.9 shows a snapshot of the Attentive Flashlight being used in a virtual environment.

Shared Viewpoint Control for Information Visualisation

Shared Viewpoint Control [Hughes, 2002], consists in an improvement to the previously mentioned attentive techniques. Here, instead of offering suggestions to the user, the operator shares the control of the viewpoint with the system. The Shared Viewpoint Control can be implemented at several levels depending on how the information in the CMF is used:

- Complete Guidance: With this method, the user is limited to observe as the computer

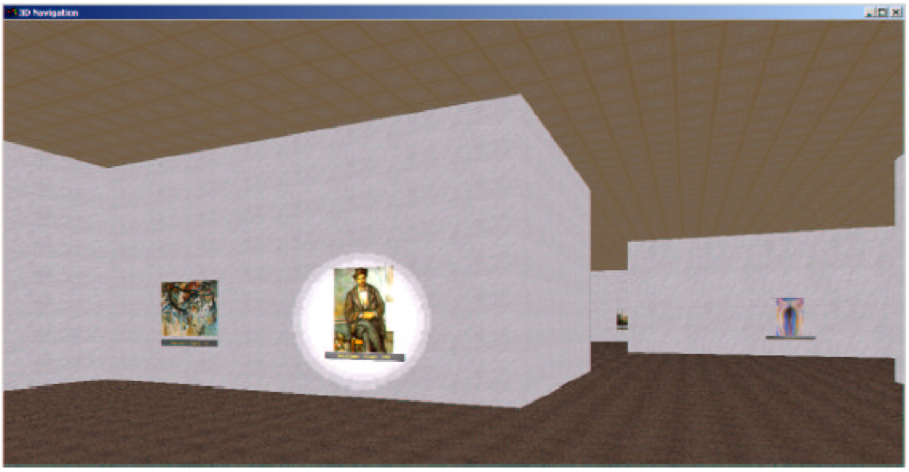


Figure 2.9: Attentive Flashlight [Hughes and Lewis, 2002].

determines the appropriate viewpoint changes required to understand the environment. This is accomplished by using the motion vectors from the CMF to move the viewer to a new location. The CMF is given the current location and responds with the ”ideal” next step. After the position has been updated, a similar operation is done with the gaze CMF to align the orientation of the camera with the ideal viewing location. This process can be repeated, leading to continuous motion through the display.

- Restricted: This method allows partial control of the viewpoint to be assumed by the viewer. Either the camera orientation or the motion is determined by the viewer while the other is obtained recurring to the CMF. If the system determines the motion and the viewer the orientation of the camera, the situation is analogous to riding in a tour bus, where the bus follows a predetermined path while riders are free to choose where they wish to look at. As a person moves through the environment, their gaze is automatically redirected to focus on objects of interest. By using this type of navigation, the viewer gets a preview of all the main features in an environment (see Fig. 2.10).

Both the previously presented methods restrict the user’s options. It is therefore necessary to establish the restrictions accurately. For instance, if the viewer is restricted from acquiring a viewpoint which is thought to be necessary, it will quickly lead to viewer frustration.

- Facilitated: This approach consists in extending the viewer’s control. The role of the

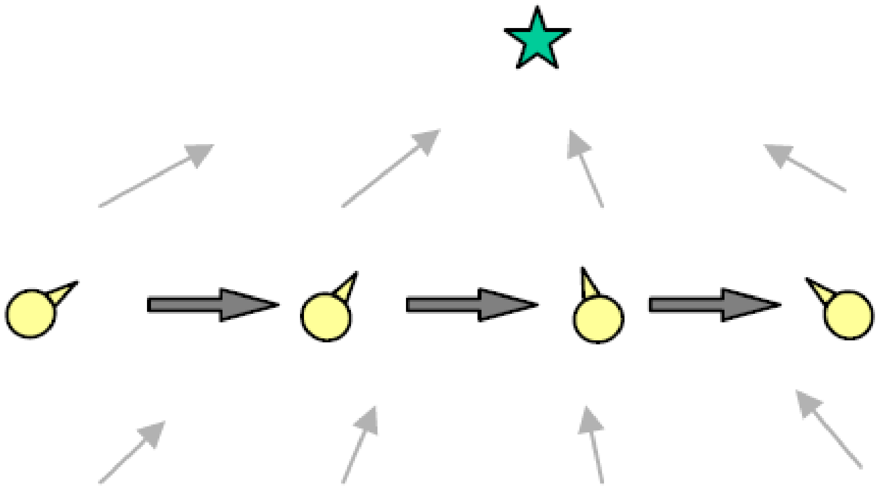


Figure 2.10: Restricted control, where the motion is controlled by the viewer and the orientation is controlled by the system, results in automatic gaze redirection to fixate on significant objects [Hughes, 2002].

CMF in this method lies in suggesting a direction of motion or gaze, that means, as an addition to the restricted method, now the viewer is able to override the suggestion into a certain point. For example, a system with facilitated control may be designed to keep the viewer from straying too far from the plan, and therefore limit the gaze to within 45° of the ideal.

- Requested: The viewer is granted full autonomy for this level of control. However, the viewer is able to request the system (e.g. the CMF) for help. The viewer is able to explore the scene on his own but at any time, he can recognise that help is necessary and is able to ask the computer to intervene and show the ideal direction of gaze or align the camera with the pointing direction.

- Annotated: In this method, the CMF vectors are used to power additional imagery in order to supplement the standard visualisation. At this point, the computer is not capable of directly intervening in the control of the viewpoint, being up to the operator to control it in a consistent manner with what they perceive the machine suggests (see Fig. 2.11).

The power to adjust the amount of control is not necessarily a design level decision. The appropriate level of assistance is instead determined by the information needs of the viewer. For

Figure 2.11: Annotated control: Arrows are being used to highlight critical elements of the display [Hughes, 2002].

instance, the interaction with someone seeking specific information is likely to be very different from someone who wishes to learn about the environment at large. Therefore, collaboration with the system may vary from user to user and from use to use. In some cases, it might be necessary for the user to be in complete control, with the machine offering no assistance.

The greater issue is the lack of capacity of the aforementioned techniques (attentive techniques and shared control) to function on unpredictable environments as they require the CMF to be defined before the system is launched.

Augmented reality

As previously mentioned, pan-tilt errors can lead to operator confusion and error when the camera is off-centre and the operator is navigating the robot. In [Nielsen et al., 2005], a method which aims at facilitating the use of a pan-tilt camera is presented. For this purpose, a 3D augmented reality display is used to render the video stream at an angle that corresponds to the orientation of the real camera. This provides an intuitive representation of the pan and tilt angles of the camera as they relate to the orientation of the robot.

By presenting the orientation of the camera as it relates to the robot's orientation in an intuitive, integrated display, it is possible to increase the operator's anticipation of how the robot responds to commands.

Tests with this method suggest that a 3D interface facilitates the use of a pan-tilt camera and including the navigation process, helping on the avoidance of walls and required less mental workload of the operator, though, they do not offload any commands from the operator's hands.

Discussion

We have already seen that cameras are a common denominator in most teleoperation interfaces despite the limitations they offer. In this section we have analysed some methods which aim at facilitating the use of a camera for robotic remote exploration. The first thing that should be mentioned is that most of the work done on this field, with exception to the ”augmented reality” method presented in this section, is limited to experiments in virtual environments and few work has been published using real models. Despite the fact that they offer interesting solutions for sharing the control of a camera, some of the concepts used are too focused on virtual environments, leaving out the concept of GPS positions, Azimuth and Elevation and the problems inherent to teleoperation in the real world, i.e. latency, and errors in communication.

Chapter 3

Odometry Modelling of a Four-Wheel-Steered (4WS) Vehicle

In this chapter an odometry model for over-constrained 4WS vehicles is proposed. Odometry is known to be a popular localisation method in the robotics community and 4WS vehicles, though being widely used, due to their unique mechanical characteristics are known to be troublesome in what is concerned with odometry obtention. The most popular methods to obtain the odometry of such vehicles are based on heuristics, though, they do not offer an error model, which is important if the odometry information is to be fused with other localisation mechanisms. In section 3.1 an overview on the characteristics of 4WS vehicles is presented. Section 3.2 presents a method for obtaining the odometry of a 4WS vehicle followed by a description of its error model in section 3.3.

3.1 Kinematic Aspects of a 4WS Vehicle

Four wheel vehicles have always been explored by the robotics community. The Lunar Roving Vehicle (LRV) [Williamson, 2002] made a lunar excursion. This rover had four-wheel traction and was provided with double-Ackerman steering. More recently, Jet Propulsion Lab's (JPL) Rocky series of micro-rovers [Miller, 2008], which lead to the development of the Sojourner

[Dubov, 1996], which, though being a six wheeled rover, used four-wheels with independent steering. The spatial rover communities have developed many innovative vehicle systems, the most widely used being the Russian designed ”Marsokhod” vehicle.

The Nomad vehicle [Bapna et al., 1998b], developed at Carnegie Mellon University (CMU) stands out as one of the most innovative of all recent vehicle designs. Its main innovation is concerned with locomotion, which design features four-wheel drive locomotion, a reconfigurable chassis, electronically coordinated steering, among others. Another interesting feature is concerned with its steering system, which enables both skid and explicit steering.

More recently, the Robust Autonomous Vehicle for Offroad Navigation (RAVON) appeared [Schäfer and Berns, 2006], with four wheel drive and independent motors, the front and rear axles steer independently, supporting manoeuvres like double Ackerman and parallel steering.

On independent steering vehicles, each wheel assembly is explicitly steered. Synchronised or all-wheel explicit steering schemes can emulate any rigid-chassis steering. The heading change is achieved by modulating the steering angle and direction by which the heading of the wheel changes. This scheme is greatly advantageous from the manoeuvrability point of view, specially in the case of vehicles which are expected to operate in rough terrains, though, actuation complexity and accuracy control can become a problem. This characteristic of high mobility enables low friction quasi-holonomic motions. In some case, like Search and Rescue (S&R) [Carlson and Murphy, 2005] or landmine detection [Santana et al., 2008b], where ground disturbance is undesired this sort of mobility has an important part. In addition to this, high manoeuvrability is also essential for the robot to move in highly cluttered environments.

The most common locomotion schemas for 4WS rovers are the following:

Ackerman: This refers to a car-like locomotion method, this steering schema, despite the fact that it can be available, it is easily supplanted by the double-Ackerman mode, where the rear wheels are steered as a function of the deflection of the front-wheels. Generally, the adopted strategy for such locomotion mode consists in turning the front and rear wheels with the same angles, continuously maintaining the Ackerman geometry (see Fig. 3.1 (a)).

Linear Displacement: In this mode, the four wheels are aligned to produce linear trajecto-

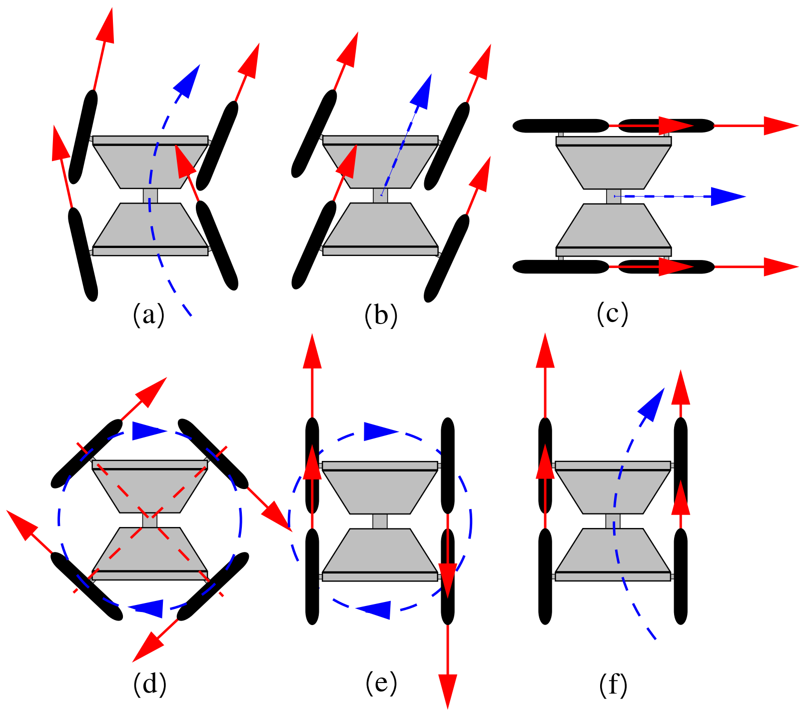


Figure 3.1: The most common locomotion schemas for 4WS rovers: (a) double-Ackerman, (b) Linear Displacement, (c) Lateral Displacement, (d) Turning-Point, (e) Skid Steering Point Turn and (f) Differential Skid Steering (Image adapted from [Shamah, 1999]).

ries without varying the vehicle's heading (see Fig. 3.1 (b)).

Lateral Displacement: Also known as crab displacement, this is a special instance of the previous mode, in which the four wheels are aligned and perpendicular to the main axis of the robot, allowing the robot to move sideways (see Fig. 3.1 (c)). This mode is considered separately from the linear displacement as the wheels, despite the fact that they are aligned, have opposite angles.

Turning-Point: In this mode the robot is able to rotate around its own geometrical centre without lateral slippage (see Fig. 3.1 (d)).

Skid Steering Point Turn: This mode is similar to the previous one, though, the turn is executed with lateral slippage. This mode is more suitable for differential drive robots than 4WS ones (see Fig. 3.1 (e)).

Differential Skid Steering: This mode consists in maintaining the wheel angles aligned with the longitudinal axis of the vehicle at all times. The difference on the velocities from the left and right side of the vehicles enables the vehicle to vary its heading while moving forward at the same time (see Fig. 3.1 (f)).

As this dissertation is related with a demining robot, where ground disturbance is highly undesired, skid steering locomotion schemas will not be further analysed.

3.2 Odometry Model of a 4WS Vehicle

The goal behind the odometry model described in the remainder of this chapter, consisted in developing a model which would receive encoder readings providing the angles and the distance travelled by each wheel and return an estimate on the variation of the position of the rover as well as the error of the estimated variation. These estimated variations and the error will afterwards ”feed” a filter which will propagate the position of the rover, as well as the odometry error and make the fusion between all localisation sensors/mechanisms present on the rover.

Figure 3.2 shows a draft of a 4WS mobile robot, where l corresponds to the distance between the front and rear axles, $2t$ is the width of both front and rear axles, TW is the gap between the kingpin and the centre of the circumference which describes the wheel, δ_i is the steering angle, d_i the distance travelled and ρ_i the turning radius. The subindex i represents one wheel. The vehicle motion has a centre called ”instant rotating centre” (IRC) which, in case of parallel steering of all the wheels (e.g. linear displacement and lateral mode), can reach to infinity. The IRC of the vehicle motion is defined by the steering angle of the vehicle's wheels.

Generally, the reference frames for obtaining wheel odometry are placed at the geometric centre of the vehicle. Though, in our case the reference frame for obtaining the wheel odometry of the 4WS vehicle is placed at the centre of its front axle (see Fig. 3.2). This is due to the fact that, in the robot used in the AMI-02 project, the localisation sensors, were placed on top of the front axle and aligned with its centre. For the sake of coherence with the remaining localisation sensors, it was decided to establish the centre of the front axle as the reference frame for obtaining wheel odometry.

Encoders generally return angular distances. In order to convert the angular distance α_i travelled by a wheel i to the linear distance d_i it is only necessary to apply the following equation:

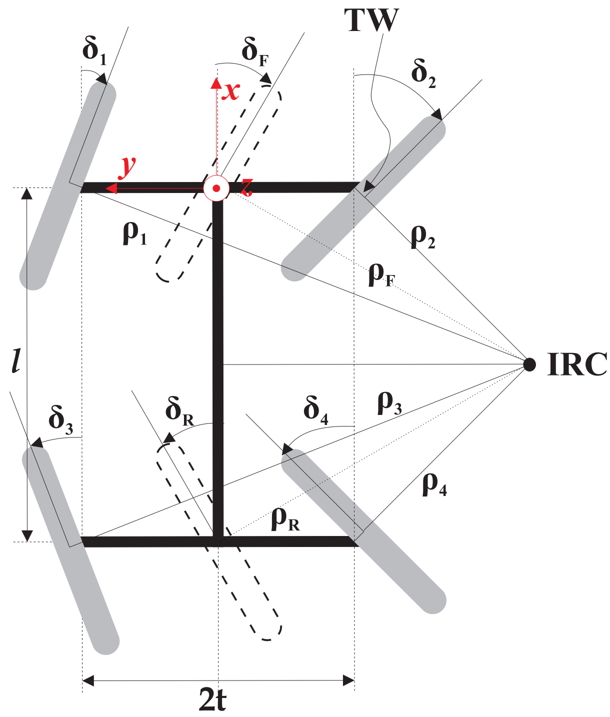


Figure 3.2: Model demonstrating a 4WS robot turning to its right in Double-Ackerman steering mode. Note that the front wheels' steering angles increase when turning CW, while the rear wheels' steering angles increase when turning CCW.

$$d_i = \alpha_i r \tag{3.1}$$

where r corresponds to the nominal radius of the wheels of the robot (it is assumed that all wheels have the same dimensions).

According to wheel i and based on the constraints of the locomotion mode in question, the robot's displacement and heading variation, u_i^{w}[1] (3.2) can be estimated from the wheel's steering angle and travelled distance. A wheel odometer has been implemented in order to provide estimation of 3-DOF motion in body frame u^w between two consecutive iterations n and $n+1$,

[1]The super-index w allows the distinction between wheel odometry and visual odometry, denoted with super-index v, which was also used in the AMI-02 project.

$$u_i^w = \begin{bmatrix} u_{xi}^w & u_{yi}^w & u_{\phi i}^w \end{bmatrix}^T \tag{3.2}$$

The redundancy introduced by the possibility of calculating wheel odometry based on a single wheel requires a decision on which wheel to consider. We have already seen in this document several methods that exploit the redundant wheels paradigm. The *fewest pulses* and the *expert rules* methods [Ojeda and Borenstein, 2004] or the more complex case of the FIDO rover [Baumgartner et al., 2001] provide mechanisms which intend to distinguish the best wheel or set of wheels which theorically offer the best odometry results. Observations on the dynamical behaviour of the robot suggest that all wheels strongly affect each other, meaning that wheel odometry computation benefits from considering all wheels rather than just a set of wheels or in the worst case, a single one.

Hence, each wheel's estimate is brought together, as a weighed average to build up a global estimate, u^w (3.3):

$$u^w = \begin{bmatrix} \sum_{i=1}^4 \omega_i u_x^w & \sum_{i=1}^4 \omega_i u_y^w & \sum_{i=1}^4 \omega_i u_\phi^w \end{bmatrix}^T \tag{3.3}$$

where the weight of each contribution, ω_i, is,

$$\omega_i = \frac{1 - e_i/\Sigma e_i}{\Sigma(1 - e_i/\Sigma e_i)}, \qquad \sum_{i=1}^4 \omega_i = 1 \tag{3.4}$$

where,

$$e_i = |\hat{\delta}_i - \delta_i| \tag{3.5}$$

which corresponds to the difference between the expected angle for wheel i, $\hat{\delta}_i$, and its current steering angle, δ_i. $\hat{\delta}_i = \frac{1}{3}\sum_{j \neq i} \Phi(\delta_j)$, where $\Phi(\delta_j)$ returns the steering angle wheel i should have in order to be coherent (according to the locomotion mode in question) with the steering angles of wheels j.

Intuitively, the odometry computed based on each wheel is weighed for the global estimate according to a function of its estimated error, e_i , normalised with the other wheels estimated errors. Since it is impossible to determine which wheel is failing to meet the kinematic constraints of the locomotion mode, the error of each wheel is computed in relative terms by considering that all other wheels are in the correct position.

On the following lines the equations that will allow the obtention of odometric results for each locomotion mode will be described. The result u^w will be given relatively to the body-frame, where the x-axis points forward, the y-axis points left and the z-axis points upward.

Double Ackerman

For the obtention of the double Ackerman odometry, the following hypothesis are made:

• Hypothesis 1: the path described by the wheels between two consecutive sampling times is an arc of circle (see Fig. 3.3); this allows the assumption that the steering command of the robot freezes between sampling periods; and

• Hypothesis 2: the sampling rate of the state and controls is very small compared to their rate of change; this ensures that linearising errors and curvature discontinuities at elementary path ends are small.

The calculations behind the double-Ackerman geometry are complex, therefore, to simplify, the bicycle (also known as single track) model abstraction is used [Kim et al., 2004], [Proetzsch et al., 2005]. This model bases itself on curve trajectories of the robot, therefore, straight lines must be considered as a particular case of the linear displacement mode, where $\delta_1 = \delta_2 = \delta_3 = \delta_4 = 0$.

According to the double Ackerman geometry constraints we have that $\delta_1 = \delta_3$, $\delta_2 = \delta_4$ and $\delta_F = \delta_R$, where δ_F and δ_R correspond to the front and rear fictitious wheel angles. As $\delta_F = \delta_R$ from now on they will only be referred as δ_C

In [Spentzas et al., 2001] the wheel-angle relation is described by the following expression:

$$\frac{1}{tan\delta_1 + tan\delta_3} - \frac{1}{tan\delta_2 + tan\delta_4} = \frac{2t}{l} \tag{3.6}$$

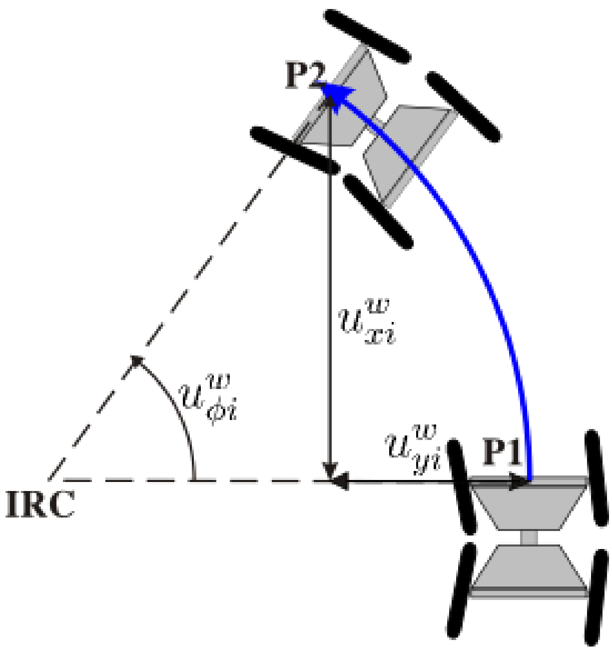


Figure 3.3: Double Ackerman geometry. Over a small time period, the robot's motion between points P1 and P2 can be approximated by an arc. In the figure the robot is moving in CCW direction.

From (3.6) it is possible to calculate the angle δ_C each wheel i induces in the bicycle model. This value will be referred as δ_{Ci}:

$$\delta_{Ci} = atan\left(\frac{1}{\frac{1}{tan\delta_i} + (-1)^i \cdot \frac{2t}{l}}\right) \tag{3.7}$$

The turning radius of each wheel, i.e. the distance of wheel i to the IRC, [Kim et al., 2004], concretely to the double Ackerman geometry, can be obtained by,

$$\rho_i = \frac{l}{2sin\delta_i}, \quad \rho_{Ci} = \frac{l}{2sin\delta_{Ci}} \tag{3.8}$$

The distance between the kingpin and the wheel, TW, ideally should be included in the computation of ρ_i, though, it requires a particular case for every wheel, whether the wheel is turning right, ($\delta_i > 0$) or turning left, ($\delta_i < 0$). As it does not introduce significant changes in the final results, for the sake of simplicity, in this particular locomotion mode, TW is therefore ignored.

The description of the variation of Ares' position, offered by each wheel i in double-

Ackerman displacement mode is given by:

$$u^w_{\phi i} = -\frac{d_i}{\rho_i} \tag{3.9}$$

$$u^w_{xi} = |\rho_{Ci}|sin(sgn(\delta_{Ci})u^w_{\phi i}) \tag{3.10}$$

$$u^w_{yi} = -\rho_{Ci}(1 - cos(u^w_{\phi i})) \tag{3.11}$$

On this mode the expected angles $\hat{\delta}_i$ are given with respect to the bicycle model:

$$\hat{\delta}_i = \frac{\sum_{j=1}^{4} |\delta_{Cj}|}{3}, \qquad j \neq i \tag{3.12}$$

Taking this into account, expression 3.5 needs to be adapted to this particular mode as the wheel angles of the bicycle model are being compared. Therefore the operand δ_i is replaced by δ_{Ci} allowing the comparison between the correspondent bicycle angle, δ_{Ci} one wheel induces and $\hat{\delta}_i$, which in this case, represents the bicycle angle induced by the remainder wheels:

$$e_i = ||\hat{\delta}_i| - |\delta_{Ci}|| \tag{3.13}$$

Linear Displacement

This locomotion mode bases itself on the following constraints:

$$\begin{cases} \delta_1 = \delta_2 = -\delta_3 = -\delta_4 \\ d_1 = d_2 = d_3 = d_4 \end{cases} \tag{3.14}$$

This means that the robot moves without presenting any variation on its heading. Having this in consideration, the odometry results can be obtained recurring to:

$$u^w_{\phi i} = 0 \tag{3.15}$$

$$u^w_{xi} = d_i \cdot cos(\delta_i) \tag{3.16}$$

$$u^w_{yi} = \begin{cases} d_i \cdot sin(\delta_i), & i = 1, 2 \\ d_i \cdot sin(-\delta_i), & i = 3, 4 \end{cases} \tag{3.17}$$

On this mode the expected angle $\hat{\delta}_i$ is computed as follows:

$$\hat{\delta}_i = \frac{\sum_{j=1}^{4} |\delta_j|}{3}, \qquad j \neq i \tag{3.18}$$

Lateral Displacement

This locomotion mode bases itself on the following constraints:

$$\begin{cases} \delta_{1,3} = \pi/2 \\ \delta_{2,4} = -\pi/2 \\ d_1 = -d_2 = -d_3 = d_4 \end{cases} \tag{3.19}$$

which leads the robot to present variations only in the y-axis:

$$u^w_{\phi i} = 0 \tag{3.20}$$

$$u^w_{xi} = 0 \tag{3.21}$$

$$u^w_{yi} = \begin{cases} d_i, & i = 1, 4 \\ -d_i, & i = 2, 3 \end{cases} \tag{3.22}$$

On this mode the expected angles $\hat{\delta}_i$ are static as indicated in equation (3.20), therefore we have:

$$\hat{\delta}_i = \frac{\pi}{2} \tag{3.23}$$

Turning Point

Theoretically, on this mode there would not be any variation on the values u_x or u_y, though, as the GPS antenna and the stereo-head are placed in the front of the robot, the "odometric-centre" should also be considered on this same point.

$$u^w_{\phi i} = (-1)^i \frac{d_i}{\sqrt{\left(\frac{l}{2}\right)^2 + t^2 + TW}} \quad (3.24)$$

$$u^w_{xi} = -\frac{l}{2}\left(1 - cos(u^w_{\phi i})\right) \quad (3.25)$$

$$u^w_{yi} = \frac{l}{2} sin(-u^w_{\phi i}) \quad (3.26)$$

On this mode the expected value, $\hat{\delta}$, is constant, only depending on the structure of the robot and also the constraints of this locomotion mode, which leads to the following expression,

$$\hat{\delta}_i = cos\left(\frac{t}{\sqrt{t^2 + \left(\frac{l}{2}\right)^2 + TW}}\right), \quad \hat{\delta}_1 = \hat{\delta}_2 = \hat{\delta}_3 = \hat{\delta}_4 \quad (3.27)$$

3.3 The Error Model

As previously mentioned, odometric position updates can only provide a very rough estimate of the actual position. Due to integration errors of the uncertainties of the motion estimation u^w and the motion errors due to the linear distance, d_i travelled by each wheel and its steering angle δ_i, the position error based on odometry integration grows with time. Here, we are only concerned with the error of odometry estimates. As previously mentioned, the propagation of the error is executed by a filter whose description is outside the scope of this dissertation.

Here, an error model for the position variation u^w which will allow the obtention of the covariance matrix C_Y of the odometric position estimate will be established. The estimate on the variation of the position of the robot is assumed to be normally distributed.

The input covariance matrix C_{X_i} contains all variances and covariances of the input random variables. Each wheel is computed independently, therefore, for the motion variation of wheel i the following covariance matrix is obtained:

$$C_{X_i} = \begin{bmatrix} \sigma_{v_i}^2 & \sigma_{v_i\delta_i} \\ \sigma_{\delta_i v_i} & \sigma_{\delta_i}^2 \end{bmatrix} \tag{3.28}$$

where $\sigma_{v_i}^2$ and $\sigma_{\delta_i}^2$ are, respectively, the variances of the read velocities and the steering angles of wheel i and $\sigma_{\delta_i v_i}$, $\sigma_{v_i\delta_i}$ the covariances between these input variables.

The following assumptions are made:

- The errors of both input variables are independent;
- The variance of the error of the travelled distance, $\sigma_{d_i}^2$, as well as the variance of the error of the steering angle of a wheel $\sigma_{\delta_i}^2$ varies from wheel to wheel.

If the input variables are independent, all σ_{ij} with $i \neq j$ disappear and C_{X_i} is a diagonal matrix, therefore, for the motion variation of wheel i the following covariance matrix C_{X_i} is assumed:

$$C_{X_i} = \begin{bmatrix} k_{d_i} & 0 \\ 0 & k_{\delta_i} \end{bmatrix} \tag{3.29}$$

where k_{d_i} and k_{δ_i} are, respectively, error constants representing the nondeterministic parameters as the characteristics of the motor-drive (e.g. condition of the mechanical connections).

These assumptions are far from being perfect, though, they are suitable and will thus be used for the further development of the error model. The motion errors are due to imprecise movement because of wheel deformations, slippage, floor characteristics, errors in encoders and so on. The values of k_{d_i} and k_{δ_i} depend on each wheel of the robot and the environment. They should also be experimentally established by performing and analysing the behaviour of the system in such environments. The constants k_{δ_i} and k_{d_i} should be considered to be different for each wheel i as their mechanical constraints are different. For instance wheel 4 can be looser than wheel 1 on the steering or on the traction actuator.

The next step consists in applying the error propagation law [Arras, 1998]:

$$C_{Y_i} = F_{X_i} C_{X_i} F_{X_i}^T \tag{3.30}$$

where C_{X_i} is a 2×2 (due to the two variables each wheel provides), C_{Y_i} a 3×3 (due to the three parameters of the odometry estimate each wheel provides) covariance matrix and F_{X_i} a matrix of dimension 3×2. The Jacobian F_{X_i} is the transpose of the gradient of the position variation vector of wheel i, u_i^w:

$$F_{X_i} = \left[\nabla_{d_i,\delta_i} \cdot u_i^{w\,T}\right]^T = \begin{bmatrix} \frac{\partial u_{x_i}^w}{\partial d_i} & \frac{\partial u_{xi}^w}{\partial \delta_i} \\ \frac{\partial u_{y_i}^w}{\partial d_i} & \frac{\partial u_{y_i}^w}{\partial \delta_i} \\ \frac{\partial u_{\phi_i}^w}{\partial d_i} & \frac{\partial u_{\phi_i}^w}{\partial \delta_i} \end{bmatrix} \tag{3.31}$$

Due to the two input variables and three outputs of the system, matrix C_{Y_i} is on its final form:

$$C_{Y_i} = \begin{bmatrix} \sigma^2_{u_{x_i}^w} & \sigma^2_{u_{x_i}^w u_{y_i}^w} & \sigma^2_{u_{x_i}^w u_{\phi_i}^w} \\ \sigma^2_{u_{y_i}^w u_{x_i}^w} & \sigma^2_{u_{y_i}^w} & \sigma^2_{u_{y_i}^w u_{\phi_i}^w} \\ \sigma^2_{u_{\phi_i}^w u_{x_i}^w} & \sigma^2_{u_{\phi_i}^w u_{y_i}^w} & \sigma^2_{u_{\phi_i}^w} \end{bmatrix} \tag{3.32}$$

As previously mentioned, an odometry estimate is assumed to have a normal distribution. From the properties of the normal distribution, we know that:

1. If $X \sim N(\mu, \sigma^2)$ and a and b are real numbers, then $aX + b \sim N\left((a\mu + b, (a\sigma)^2\right)$.

2. If $X \sim N(\mu_X, \sigma_X^2)$ and $Y \sim N(\mu_Y, \sigma_Y^2)$ are independent normal random variables, then their sum is normally distributed with $U = X + Y \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$.

On the previous section it has been seen that the global estimate is built as a weighed average between each wheel's estimate (see equation 3.3), therefore, the overall error is given by matrix C_Y which has the form.

$$C_Y = \begin{bmatrix} \sigma^2_{u_x^w} & \sigma^2_{u_x^w u_y^w} & \sigma^2_{u_x^w u_\phi^w} \\ \sigma^2_{u_y^w u_x^w} & \sigma^2_{u_y^w} & \sigma^2_{u_y^w u_\phi^w} \\ \sigma^2_{u_\phi^w u_x^w} & \sigma^2_{u_\phi^w u_y^w} & \sigma^2_{u_\phi^w} \end{bmatrix} \tag{3.33}$$

By applying the properties of the normal distribution, the global covariance matrix for the position estimation of the robot is given by:

$$C_Y = \sum_{i=1}^{4} \left(\omega_i^2 \cdot C_{Y_i} \right) \tag{3.34}$$

Having the error model established, the error parameters k_{v_i} and k_{δ_i} (see equation 3.29) must be obtained. As we have already seen in this document, it is possible to properly compensate for systematic errors with proper calibration of the robot, however, the error parameters specifying the non-systematic errors can only be quantified and compensated with repetitive measurements, or on-the fly as we have already seen in chapter 2.

Chapter 4

Control System of a Mobile Robot's Teleoperation Camera

This chapter presents a control system for a teleoperation camera. This system has been developed from bottom up: the development of a position controller for the actuators of the camera is presented in section 4.2. As previously mentioned, when an operator has to navigate a robot and at the same time control a camera, he can rapidly become overwhelmed, therefore it is important to have other mechanisms that help the operator in achieving these tasks. In section 4.3 a set of mechanisms are proposed to help the operator in the task of controlling a MDOF teleoperation camera.

4.1 A Position Controller for a PTZ Camera

Fig. 4.1 depicts the camera controller's block diagram. Let us assume the following reference frames: *World frame* $\{X_w, Y_w, Z_w\}$, *Site frame* $\{X_s, Y_s, Z_s\}$, *Rover frame* $\{X_{rov}, Y_{rov}, Z_{rov}\}$ and the *Camera frame* $\{X_{cam}, Y_{cam}, Z_{cam}\}$.

The world frame is fixed with regard to earth and its X-axis points North, Y-axis points west and Z-axis is parallel but opposite in sign to the local gravity vector, its origin is at the intersection between the Greenwich meridian and the equator. The site frame is the surface-relative

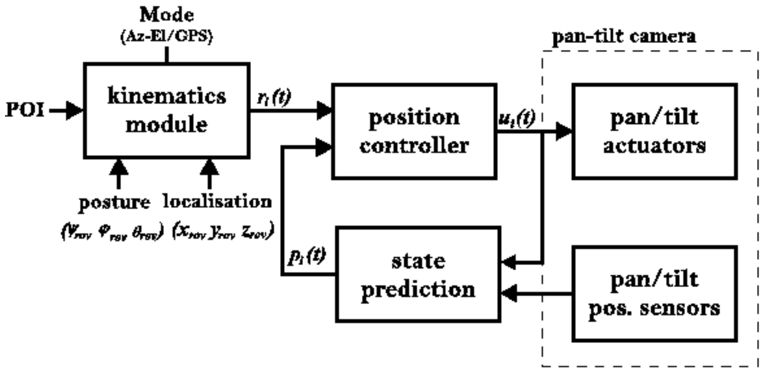


Figure 4.1: Overview of the camera controller.

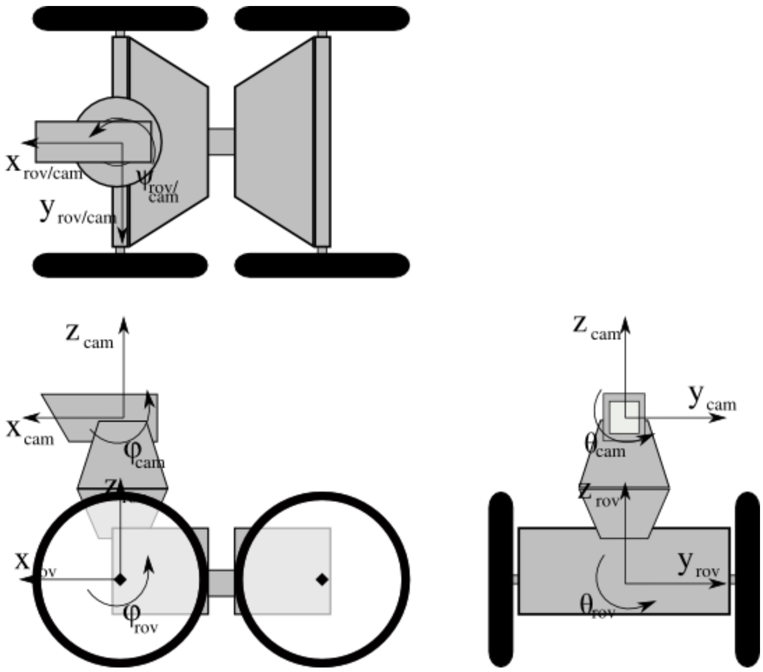


Figure 4.2: Rover and camera coordinate frames.

frame of reference for all targeting and localisation operations. It has the same orientation of the global reference frame, though, its origin is defined by the user. The frame which describes the rover is defined as follows: X-forward, Y-left and Z-up and it is placed (as explained in the previous chapter) in the geometric centre of the front axle. The camera frame is identical to the rover frame, except that its origin is located in the camera's Tilt axis (see figure 4.2) (e.g. with regard to the rover frame, the camera frame is translated a T_z distance on the Z-axis).

Heading (ψ_{rov}), pitch (φ_{rov}) and roll (θ_{rov}) represent the angle values relative to Z_{rov}, Y_{rov} and X_{rov}, respectively, as depicted in figure 4.2, and they are used to define the robot's absolute attitude, measured against a fixed reference (e.g. the horizontal plane).

The rover's attitude can be represented by a product of the three aforementioned rotations. The well known Tate-Bryant rotation sequence convention is used to represent the rover's 4×4 posture matrix, $P_{rov}(\psi_{rov}, \varphi_{rov}, \theta_{rov})$, where the first rotation is applied around the Z-axis, the second around the Y-axis and finally around the X-axis,

$$P_{rov}(\psi_{rov}, \varphi_{rov}, \theta_{rov}) = R_z(\psi_{rov}) R_y(\varphi_{rov}) R_x(\theta_{rov})$$

where,

$$R_z(\psi_{rov}) = \begin{bmatrix} cos(\psi_{rov}) & -sin(\psi_{rov}) & 0 & 0 \\ sin(\psi_{rov}) & cos(\psi_{rov}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$R_y(\varphi_{rov}) = \begin{bmatrix} cos(\varphi_{rov}) & 0 & sin(\varphi_{rov}) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\varphi_{rov}) & 0 & cos(\varphi_{rov}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$R_x(\theta_{rov}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\theta_{rov}) & -sin(\theta_{rov}) & 0 \\ 0 & sin(\theta_{rov}) & cos(\theta_{rov}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The kinematics module allows the user to position the camera according to one of the two proposed methods for shared control. The inputs of this module are "mode" and the point of interest (POI), given in azimuth and elevation, in GPS or rover coordinates according to the mode in use, provided by the user, allowing the selection of pointing in azimuth/elevation, GPS or rover pointing. In order to compute the correct angles to point at the desired objective, this module must also receive the posture and the localisation of the robot, represented by matrix,

$$rov = \begin{bmatrix} x_{rov} & y_{rov} & z_{rov} & \psi_{rov} & \varphi_{rov} & \theta_{rov} \end{bmatrix}^T .$$

The output of this module consists in the reference Pan/Tilt angles which will define the input of the position controller.

The position controller compares the desired Pan and Tilt angles r_i (where $i = P$ if it is referred to the pan or $i = T$ if it is the tilt axis) provided by the kinematics module with the position of the camera, p_i (an actual or a predicted value). The output of this block consists in the velocities of the pan and tilt actuators which will lead the camera to move into the desired position. This block as well as the state prediction block are explained with more detail in Section 4.2.

4.2 Camera Pan/Tilt Position Control

As previously mentioned, inspecting an environment through a camera can be a daunting task. One way of reducing the effects of reducing this problem consists in performing smooth transitions between perspectives as it allows an operator to maintain situational context as the perspective changes [Nielsen et al., 2007]. This section focuses on a controller to smoothly position a teleoperation camera.

A legacy surveillance Pan-Tilt-Zoom (PTZ) camera was used. The impossibility of introducing the software controller into the camera, lead to the necessity of having the controller in a remote processor, requiring a communication interface that could be easily integrated between both. Several problems were identified: Sluggish and error prone communication and excessive number of queries necessary to retrieve little information (e.g. pan/tilt positions).

Latency is a common problem in different types of applications (e.g. interplanetary communication, image processing, etc.). In this case, it is due to the fact that the controller is placed remotely from the camera and therefore required a communication interface constrained to the legacy camera. Both the identified problems with the camera consist in the introduction of delays in the control system. Neglecting such delays results in systems with slow responses as the gains need to be conservative to reach stability. The only way to improve performance at this level consists in recurring to prediction to compensate time delays [Murray et al., 1993]. A necessary requirement to implement prediction is to have knowledge of the dynamic model of the system, in this case, the PTZ camera.

Ideally, at each cycle, the controller should have access to the pan and tilt angle values, which results in two queries. The predictive algorithm consists in estimating, either the pan or the tilt angles of the camera each cycle based on the period of the cycle and on the induced speed, either on the pan or on the tilt actuators. Then, in each cycle, one single query is made (thus obtaining alternately the Pan or the Tilt angles), being the second (and remainder) query replaced by a predicted value. As every two cycles, the information about the angle value of each axis is updated with actual sensorial values, errors do not accumulate over time.

By taking out one of the queries, this algorithm allows the reduction of the period of a cycle in approximately $25\,\%$. By reducing the computation time it is possible to increase the gains of the controller without leading the system to undesired overshooting situations.

The core of the proposed controller consists of two Proportional-Derivative (PD) controllers, one per each camera actuator. The PD controller allows the obtention of satisfactory results for a static reference, though, it proved to be insufficient for a dynamic one (e.g. representing an irregular terrain). Therefore, the variation of the reference is also taken into account, significantly improving the results in dynamic situations.

The output of the controller $u_i(t)$ consists of the velocities to set the pan or tilt actuators in order to reduce the reference error, $e_i(t) = r_i(t) - p_i(t)$, where $p_i(t)$ refers to the position (true or the estimated) of axis i (see Fig. 4.3),

$$u_i(t) = e_i(t)K_{ip} + \frac{de_i(t)}{dt}K_{id} + \frac{dr_i(t)}{dt}K_{ir} \tag{4.1}$$

where K_{ip}, K_{id} and K_{ir} are the proportional, the derivative and the varying reference gains, respectively and: $p_P = -\psi_{rov}$ and $p_T = -\varphi_{rov}$.

A gain scheduling approach was used in order for the controller to cope with the non-linear

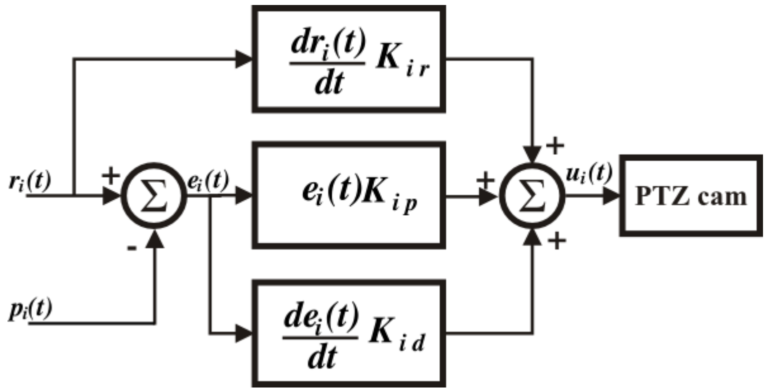


Figure 4.3: Block diagram of the position controller

behaviour of the camera in some situations [Leith and Leithead, 2000]. The scheduling variables used to determine the operating regions which enable the different gains are the position errors between the camera's actual position and the desired reference. Thus, three scheduling levels were established for the both camera actuators, leading the camera to present different behaviours in different situations.

4.3 Control strategies for shared control

Teleoperation experience can be improved significantly recurring to shared-control approaches [Adams and Skubic, 2005, Conway et al., 1990], which consists in mixing input provided by the teleoperator with information from the robot's sensors.

In this context, the controller described in section 4.2 was used to implement several mechanisms which are intended to assist teleoperation. First, three pointing mechanisms to assist on teleoperation are presented in this dissertation: (i) Pointing in Azimuth and Elevation (Az-El), (ii) Pointing to GPS point and (iii) Pointing at the Rover frame: The goal is to provide the operator with mechanisms to define the desired orientation or a location to gaze at, regardless of the robot's posture and position. With this feature, it is possible to fix on a desired point of interest without effort for the operator. Secondly, a method to assist on navigation was also developed which takes into account the desired motion direction. And last, a self-monitoring mechanism which enables the user to rapidly account for an event generated within the rover.

Each of these mechanisms is available for the user, the interface in which the user switches between these methods is not addressed in this thesis.

4.3.1 Three pointing mechanisms

An Approach to Azimuth/Elevation Following

The goal of this mechanism consists in maintaining the camera aligned within a user defined Az-El. Azimuth and elevation are angles used to define the position of an object, relative to a reference plane. The azimuth angle corresponds to the compass bearing, relative to geographic north, of a point on the horizon. Elevation is the angle measured between the horizon and the desired object, measured from the reference frame of the observer, i.e. the location of the robot.

Obtaining the Pan and Tilt camera joints in order to aim at a desired Az-El can be seen as a reverse kinematics problem. Let us use a unitary viewpoint vector represented on the earth frame by matrix Ω, to define the desired viewpoint,

$$\Omega = \begin{bmatrix} 1 & 0 & 0 & cos(-\eta)cos(\varepsilon) \\ 0 & 1 & 0 & sin(-\eta)cos(\varepsilon) \\ 0 & 0 & 1 & sin(\varepsilon) \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where, η corresponds to the desired azimuth angle between the world frame and the camera's line of sight and ε to the desired elevation angle between the world frame and the camera's viewpoint vector.

The position of the pan and tilt actuators matrix, S, can be determined from P_{rov}, and Ω as follows,

$$S = \Omega P_{rov} \tag{4.2}$$

This matrix represents the position of the pan and tilt camera actuators for a given Az-El. Matrix S is in the following form,

$$S = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \alpha \\ a_{21} & a_{22} & a_{23} & \beta \\ a_{31} & a_{32} & a_{33} & \gamma \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Before extracting the desired Pan and Tilt angles, it is necessary to transform matrix S into its row echelon form,

$$S' = \begin{bmatrix} 1 & 0 & 0 & \alpha' \\ 0 & 1 & 0 & \beta' \\ 0 & 0 & 1 & \gamma' \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The desired Pan, r_P and Tilt r_T angles are consequently given by,

$$\begin{cases} r_P(t) = atan\left(\dfrac{\beta'}{\alpha'}\right) \\ r_T(t) = asin\left(\gamma'\right) \end{cases} \tag{4.3}$$

Pointing at a GPS Point

As previously mentioned, the goal of this mechanism consists in offering the user the possibility of gazing to a user defined position in a site frame, recurring to GPS. This mechanism can be particularly useful for surveillance purposes.

The first step of the GPS pointing mechanism consists in obtaining the camera's desired Az-El between the robot's position and the GPS point to inspect. Having this information, the method explained in Section 4.3.1 is applied.

The position of the point of interest, p, in the site frame is represented in a similar way, though the information about orientation can be dropped,

$$p = \begin{bmatrix} x_p & y_p & z_p & 0 \end{bmatrix}^T.$$

Having the desired point identified, it is possible to extract its Az-El relative to the robot's position as follows,

$$\begin{cases} \eta = atan\left(\dfrac{x_{rov} - x_p}{y_{rov} - y_p}\right) \\ \varepsilon = asin\left(\dfrac{z_p - z_{rov}}{\sqrt{(x_p - x_{rov})^2 + (y_p - y_{rov})^2}}\right) \end{cases} \tag{4.4}$$

It is now necessary to determine the pan and tilt camera angles to maintain the just computed azimuth angle, η, and elevation angle, ε. For that purpose the pointing in azimuth and elevation angle proposed in Section 4.3.1 is applied.

Pointing at the rover frame

This pointing method allows the user to point the camera to a user defined coordinate (x_d, y_d, z_d) of the rover.

The rover is considered as a 3D cartesian coordinate system where the camera's pan actuator rotation centre is mounted on a point with coordinate $(x_{0\ cam}, y_{0\ cam}, z_{0\ cam})$.

The desired Pan angle, r_P is given by an arctangent, defined as follows:

$$r_P(t) = \begin{cases} atan\left(\dfrac{y}{x}\right) & x > 0 \\ \pi + atan\left(\dfrac{y}{x}\right) & y \geq 0,\ x < 0 \\ -\pi + atan\left(\dfrac{y}{x}\right) & y < 0,\ x < 0 \\ \pi/2 & y > 0,\ x = 0 \\ -\pi/2 & y < 0,\ x = 0 \\ undefined & y = 0,\ x = 0 \end{cases} \tag{4.5}$$

And the desired tilt angle $r_T(t)$ is given by:

$$r_T(t) = -atan\left(\frac{T_z}{\sqrt{x^2 + y^2}}\right) \tag{4.6}$$

where,

$$\begin{cases} x = x_d - x_{0\ cam} \\ y = y_d - y_{0\ cam} \end{cases}.$$

Here we assume a more general case, where the reference camera and the robot reference frames are displaced.

Though being relatively simple to compute, this type of pointing is extremely powerful as it enables the user to observe a coordinate of the robot with a click on a Graphical User Interface.

4.3.2 Assisted Navigation

The *gaze-directed steering* camera configuration proposed in [Bowman et al., 1997], implies the camera to be fixed (i.e. solidary with the rover's chassis), which, apart from heavily degrading inspection, is not suitable for robots with several degrees of mobility as described in chapter 3. For instance, a fixed camera is useless when a robot displaces in a direction outside the camera's field of view. To overcome this issue, a mechanism was developed which helps the user to navigate by adjusting the camera direction in the direction of travel. This method intends to take *gaze-directed steering* one step furher.

In order to release the task of positioning the camera from the operator, this method aims at positioning the camera according to the locomotion mode being used. The goal is to offload the burden of controlling the viewpoint in tasks which are more navigation oriented, where the user is more focused on following a path, instead of a search task. For instance, during a mission, a user may require to just take the robot from point A to point B and thus, it is required for the camera to focus on the path to travel on. In the same way that a driver focuses on the left side of the road when he tries to turn left or when he turns his face to the rear of the vehicle when the idea is to drive the vehicle backwards, the same principle applies to a PTZ camera: the operator attempts to position the camera in order to take a better perspective of the path to follow.

Situational awareness is known to be degraded if the operator cannot quickly assess the camera's angular magnitude of displacement. Ecological cues, such as peripheral references, two-handed compasses, among others may provide the operator with some insight to the camera orientation relative to the robot's heading. However, it is unclear if these approaches are sufficient [Hughes and Lewis, 2005]. Situational awareness degrades even more when the operator also needs to assess the current locomotion mode of the rover and adapt its visual needs of the locomotion characteristics of the selected form of controlling the robot. In order to overcome this issue, the task of positioning the camera in order to provide a better perspective of the path to follow is once again delegated on the rover. Hence, the rover freely positions the camera in function of the locomotion mode being used and the travel direction.

Double-Ackerman mode: In this mode, the camera positions itself in a way that it is able to cover the direction of travel of the fictitious wheel of the bicycle model explained in chapter 3, hence, its pan angle is a function of the front and rear fictitious angles (δ_F and δ_R) considered in the Ackerman mode. In order for the user to observe the ground, an angle equal or inferior to zero is considered for the tilt axis. This is due to the fact that the camera should point in a lower angle in order to have a better perspective of the floor the robot will step. Thus, the camera's pan, $r_P(t)$ and tilt angles $r_T(t)$ are positioned as follows:

$$\begin{cases} r_P(t) = K \cdot \delta_{F,R} \\ r_T(t) \leq 0 \end{cases} \tag{4.7}$$

Variable K can be defined according to the user's needs. For instance if it is desired for the camera to pan more or less than the actual angles of the fictitious wheels.

Linear Displacement: In this mode, the camera's pan angle should cover the trajectory of the linear movement described by the rover. Therefore, the pan angle $r_P(t)$ can be given by a simple average of the angles described by the four wheels of the rover. In what concerns the tilt angle $r_T(t)$, the same principle of the above locomotion mode can be applied; therefore we have:

$$\begin{cases} r_P(t) = \dfrac{\delta_1 + \delta_2 - \delta_3 - \delta_4}{4} \\ r_T(t) \leq 0 \end{cases} \tag{4.8}$$

Lateral Displacement: The same principle of the linear displacement locomotion mode is applied here, though, as the wheel's position is static during the whole time, the direction of motion of the robot only depends on the velocity induced on the robot (i.e. inducing the robot to displace towards its left or towards its right). Hence, the signal of the mean velocity of the four wheels is considered, therefore we have:

$$\begin{cases} r_P(t) = sgn\left(\dfrac{v_1 - v_2 - v_3 + v_4}{4}\right) \cdot \dfrac{\pi}{2} \\ r_T(t) \leq 0 \end{cases} \tag{4.9}$$

Turning-Point: In this mode, the robot turns around its geometric centre. Bearing this in mind, the camera should position itself in order to have the best possible overview of the terrain the rover will be stepping on. Ideally, the camera should have a lower tilt angle in order to capture the largest possible area where the rover steps on and it should be set according to the rover's mechanical structure.

4.3.3 Self-monitoring

As previously mentioned, teleoperation of a mobile robot is a rather complex task. The operator tends to focus in executing the mission and often ignore the robot's signals of alarm or malfunctioning. Generally control interfaces provide information about such signals, though it is common for an operator not to notice them or simply ignore them. In order to overcome this issue, a self-monitoring mechanism is here proposed.

This mechanism observes significant features of the robot (e.g. impact bumpers, pitch/roll alarms, etc.) and provides the camera with the ability of taking an active role in the monitorisation of the robot's condition by automatically positioning it in order to allow the operator to faster assess the robot's condition, evaluate the generated alarm signal and take action.

Chapter 5

Prototyping

The previous chapters proposed an odometry model for a 4WS mobile robot as well as a control system for a teleoperation camera. This chapter presents the current model's implementation.

This work is part of a wider project aiming the development of affordable service robots [Santana et al., 2007]. All the work presented in this dissertation has been tested in a real robot which will be briefly described in the following lines.

5.1 Test platform: The Ares robot

The test platform used to examine the implemented mechanisms is an all-terrain vehicle with four independently steered wheels, the Ares robot (see Fig. 5.1). This robot was developed to be sustainable and disposable, which requires high mobility, ground adaptability, reduced size, low energy consumption and low-cost [Santana et al., 2007]. In the following lines a short description of the platform is performed.

5.1.1 Mechanical Structure

Traction is a very important factor on an all-terrain robot. The robot must be able to adapt itself to the unevenness of the terrains and at the same time, be agile enough to dodge obstacles. Another important factor is the sensitivity of the terrain in some domains as landmine triggering

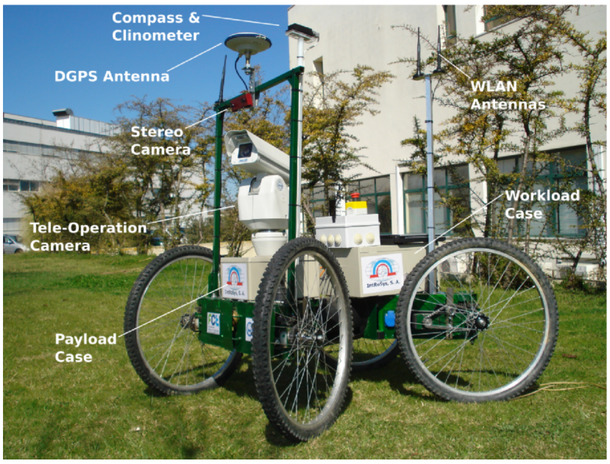
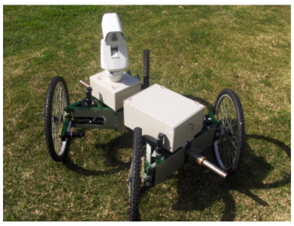


Figure 5.1: The four independently steered wheeled Ares robot.

(a) Double Ackerman

(b) Turning Point



(c) Linear

(d) Lateral

Figure 5.2: The Ares Robot prototype and its locomotion modes.

or collapse of a fragile structure are always an imminent danger in domains like humanitarian demining or S&R.

The degrees of freedom Ares possesses allow it to displace in several modes (Fig. 5.2): Double Ackerman (Fig. 5.2(a)), Point Turning (Fig. 5.2(b)), Linear Steering (Fig. 5.2(c)) and Lateral Mode (Fig. 5.2(d)).

To comply with the unevenness of some terrains, a passive spinal axle, which allows the front and rear wheels to rotate around it was introduced (see Fig. 5.3).

The dimensions of this vehicle are (See Fig. 3.2.):

Table 5.1: Dimensions of the Ares Robot.

Parameter	Size (cm)
2t	72
TW	9
l	86
r	33

Figure 5.3: Torsion caused by the spinal axle.

5.1.2 Control Hardware

The control hardware of the Ares robot was built taking into account the necessity of some modularity and flexibility in what concerns module exchange between robots. Therefore, a division between *workload* and *payload* was made. The workload module contains all the essential components for the Ares robot to be autonomous. The payload will carry all the elements that are not exclusive for Ares' operation, though essential for the purpose of its mission.

Workload

The workload is composed of the Motor Control sub-system and a set of sensors and actuators.

- **The Processing Unit** is based on a PC-104 stack (Diamond Systems Hercules EBX) running Linux (Slackware version). Communications are established recurring to a conventional wireless access point.
- **Motor Control** is performed by four roboteQ AX3500 boards (one per wheel, controlling both steering and traction), either in speed or position mode (the former was chosen).
- **Workload Actuators:** Eight maxon 150W @ 24V motors were mounted, being four for traction and the remainder for steering.

- **Localisation and Navigation Sensors:** A conventional DGPS system (Novatel OEMV-1) is used for global localisation purposes. In what concerns relative localisation, optical encoders are used for a wheel odometer and a Videre Design STOC camera for visual odometry [Santana and Correia, 2008]. For the purpose of assessing the robot's posture, a magnetic compass with clinometers was used (a Honeywell HMR3000) and with a potentiometer, Ares' spinal torsion is obtained.

  Apart from its utility in localisation purposes, the stereo-vision camera is used for other tasks like environments characterisation [Konolige et al., 2006].

Payload

- **Teleoperation Sensors:** The used camera (a Pelco Esprit) is a stock PTZ surveillance camera whose base version offers encoders for measuring the attitude of its Pan and Tilt actuators. The Zoom is not considered for this work.

  The Esprit camera communicates with the processing unit recurring to an RS-422 connection, using the following configuration:

  2400 bits/s, 8-bit data, 1 Start bit, 1 Stop bit, No Parity.

  The communication between the camera and the processing unit is based on Pelco D protocol. Esprit commands and queries are composed of a series of six bytes followed by a checksum byte. If the checksum is correct, the camera will reply with six bytes, also followed by a Checksum byte. Detailed information on this protocol can be found on [pel, 2004].

- **Mission Tools:** The robot can extend its sensorial structure in order to allow mine-detection and removal, handling objects, etc. through teleoperation (e.g. a robotic arm).

5.1.3 Control System

The aim of this dissertation does not consist in explaining the control system of the Ares robot, though, for the sake of completeness, a short description of it is presented in the following lines.

Biologically Inspired Locomotion Control

An intuitive and biologically plausible approach was developed for the control of the Ares robot. The main goal of this controller consisted in assuring that all the joints of the robot are well coordinated with each other and allow a good interaction between the robot and the terrain. For further information about this locomotion control refer to [Santana et al., 2006].

Morphological Computation for Affordable All-Terrain Piloting

This architecture exploits interactions between properly distributed simple sensors in key locations of the Ares' body and the environment in order to simplify the control structure and reduce the sensory requirements. Refer to [Cruz et al., 2005, Santana, 2005] for more information.

Mapping and Obstacle Avoidance

An obstacle avoidance algorithm for stereo vision based on [Manduchi et al., 2005] updates an Occupancy Grid [Elfes, 1987] featuring an aging mechanism. In what concerns obstacle avoidance, a set of arcs of trajectory is considered when selecting the next action to perform and each arc's utility is computed by using a set of criteria. The speed of travel is selected according to several factors: (i) the terrain's roughness, (ii) closest obstacle to the robot, (iii) desired speed and (iv) dynamical constraints of the robot. For further details on this control system of the Ares robot, refer to [Santana et al., 2008a].

5.2 Odometry in the Ares Robot

This section provides experimental results obtained with the odometry model. A draft of the hardware setup for this test can be seen on Fig. 5.4.

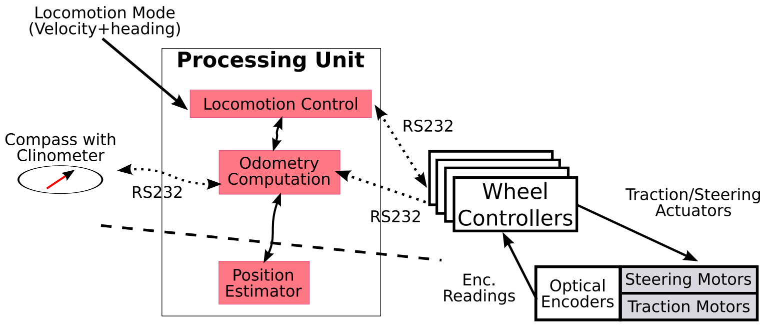


Figure 5.4: Schema of the hardware setup used for implementing the presented odometry mechanism.

As optical incremental encoders were used to determine the wheel's steering angle, it is impossible to determine this value at start up, therefore, a method for calibrating the "position 0" of the wheels' steering angles had to be implemented. This method consists in first turning the wheels' steering actuators i until they reach the $(-1)^i 45°$ position. These positions are marked with a switch which is activated as soon as the wheels reach it. As soon as all the wheels reach this position they stop and the internal variables of the wheel controller board which mark the wheels' steering angles are set with the value $\delta_i = (-1)^i 45°$. This stage is completed as soon as all the wheels reach this position and the registers are set. The next step consists in aligning the wheels' steering angles at $\delta_i = 0$. As soon as all the wheels reach this position, the reset process is complete and the rover is ready to run using odometry measurements. No further calibration methods were explored due to time constraints of this project.

The frequency at which odometry is calculated is at 10Hz. A low frequency is not reliable as the wheel angles as well as their velocities can vary and the system does not realize such changes. On the other hand, at higher frequencies an excessive number of readings is made to the encoder. Due to the nature of the optical encoders, every reading has an associated error, thus, a bigger number of readings results in larger accumulation of errors. The 10Hz value was determined by trial and error, revealing good results.

Initial tests immediately revealed the presence and the effect of the already mentioned orientation errors present in odometry. This error resulted in a constant widening of the desired trajectories, therefore, a compass was used for global convergence of the heading measured from odometry readings. The compass could not be used alone as its readings revealed high-frequency noise when the rover was moving. Therefore a combination of both heading results was used. In [Ojeda et al., 2006] a similar approach which reveals remarkable results is used, though, instead of a magnetic compass, a single axis gyro was used.

Fig. 5.5 illustrates an experiment with the robot leaving a starting point S executing a path on a terrain composed of rough grass and obstacles and return to the exact starting point. Such terrain allowed to test the ability of the presented wheel odometry method to handle a terrain where wheel friction is high, and consequently the wheels geometry is harder to maintain. The selected locomotion schema is the double-Ackerman mode.

This test allowed us to notice the importance of external magnetic fields. Until reaching point ”A”, the rover described a straight line, though, the map illustrates a deformation (marked with a thicker blue segment) which results from a deformation on the earth’s magnetic field in that same point. The reader may suggest that this does not happen if the compass is replaced by a gyroscope, though, gyroscopes have a bias drift, which results in ever-increasing orientation errors, besides here, the main goal, is global convergence.

In point A, it is possible to observe a small deformation on the trajectory. Here the rover got to close to an obstacle and in order to avoid it, it had to move back and align itself into a direction in which it would not collide with the obstacle.

After travelling approximately $75\,\mathrm{m}$, the rover stopped slightly less than $2\,\mathrm{m}$ before the initial point, i.e. it showed an odometry error around $2.6\,\%$.

Comparing the results of different odometry measurement methods is not an easy task as the results depend directly on several variables, as the characteristics of the vehicle, roughness of the terrain, among others. It would make no sense to compare the results of the method presented in this dissertation with the UMBMark (which showed an average error of $1.75\,\%$) or the IPEC, where even the ”disturbances” introduced were perfectly structured and obtained better results with disturbances than in smooth concrete floors (5-fold improvements were achieved). It was not an easy task to look for tests on grass and the most similar tests found were carried out to test the Fuzzy Logic and Expert rule-based navigation system (FLEXnav)

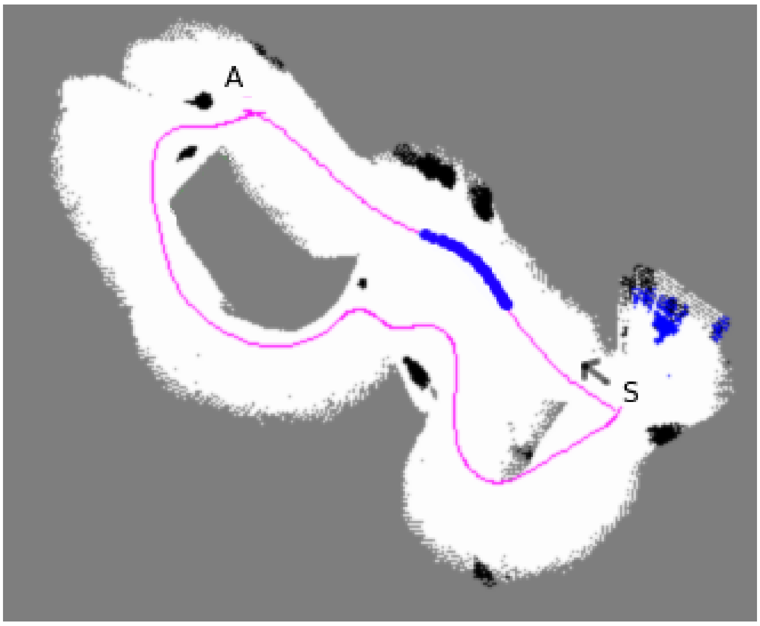


Figure 5.5: Resulting 33 m $\times$ 27 m map produced by the robot while autonomously following a path. Both starting an ending points are in location S. The travel direction is represented by the arrow. Black, white and dark grey pixels represent obstacles, non-obstacle regions and unmapped regions respectively. The closed-loop line represents the estimated location of the robot during the run. The thicker segment of the followed path illustrates the effect of an external magnetic field on the compass.

[Ojeda and Borenstein, 2002]. The FLEXnav alies the expert rules concept with the CCC explained in chapter 2 on a 6 wheel-drive robot, a kinematic equivalent to JPL's Rocky-8 and FIDO planetary rovers. The tests were carried out on a terrain with small slopes, sand and rocks and achieved an error of $14\,\%$ traveling in straight line. After adding a single axis (yaw) gyro, the robot traversed a square path and registered an average error around $2.5\,\%$. Tests on the pure sand terrain were excluded as its characteristics differ too much from grass. These results allow to compare the behaviour of the odometry model presented in this dissertation with a similar one.

Despite the accumulation of errors, odometry revealed extremely interesting results when fusing its results with a GPS system. GPS systems, though being precise, are very sensitive which leads them to sometimes show ”jumps”. Due to the smoothness of the results of the presented odometry method, the ”jump” effect of the GPS is attenuated.

Due to time constraints of the AMI-02 project, the error model could not be deeply tested but

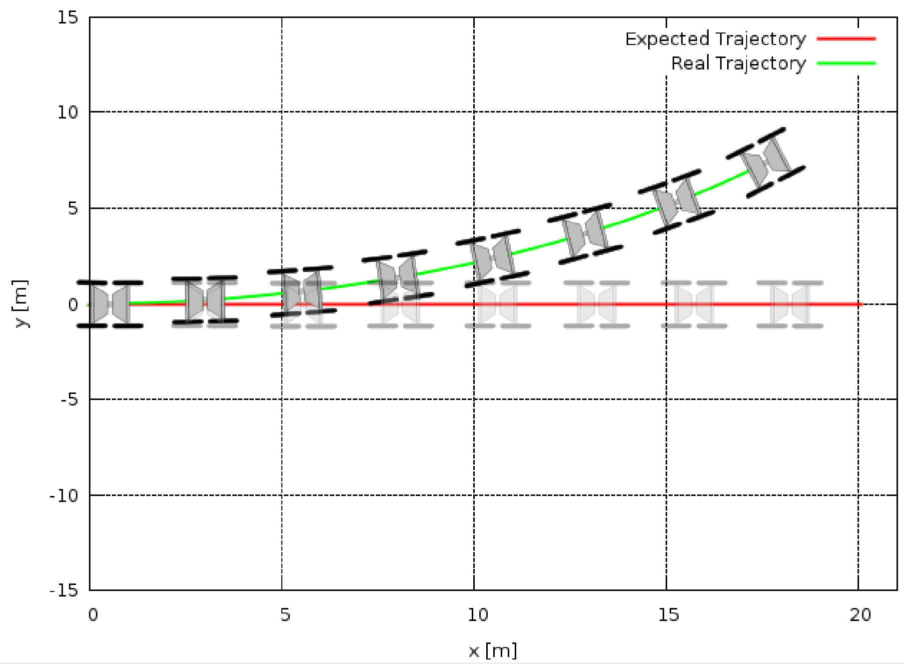


Figure 5.6: Estimated motion Vs. Expected motion.

it is wise to have some considerations about the results it would retrieve. Let us consider that the rover is commanded to travel at a velocity of $0.3\ m \cdot s^{-1}$ in a straight line in double-Ackerman mode, though, due to the errors in the encoders which allow for the obtention of the steering angle of each wheel, wheel i has a steering angle of -1^{o}. Figure 5.6 depicts the estimated motion versus the expected. Here it is possible to observe the influence of heading errors in the overall odometry estimation. It is possible to deduce that small errors in heading result in a bigger accumulation of errors in the axis Y_{rov}, when compared to X_{rov}, i.e. the uncertainty perpendicular to the movement grows much faster than in the direction of movement.

5.3 Teleoperation Camera Pointing in the Ares Robot

As previously mentioned, simultaneously operating a robot and aiming a camera on a target can be a daunting task. Therefore, to analyse the performance of the proposed pointing mechanism an experiment in all-terrain was carried out. The test consisted in defining a GPS point for the camera to target while freely manoeuvring the Ares robot in an all terrain environment (see Fig. 5.8). This test allowed us to evaluate both the position controller, the performance of the GPS

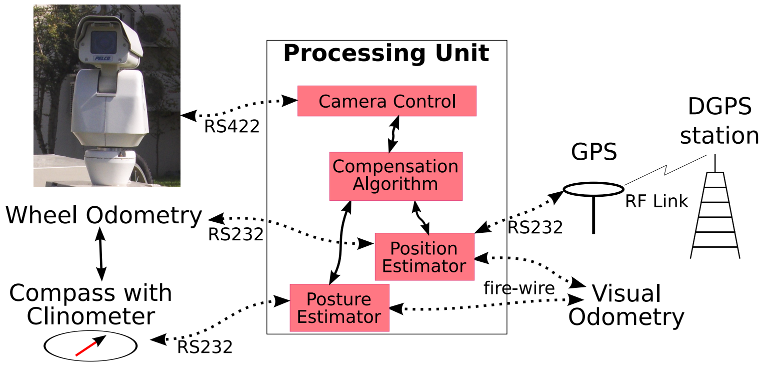


Figure 5.7: Schema of the hardware setup used for testing the pointing mechanism.

pointing and, as the GPS pointing mechanism also encompasses the Az-El pointing mechanism, it was possible to evaluate it also. The schema of the hardware setup for this test can be seen on Fig. 5.7.

The gains for each scheduling interval were tuned empirically and set as follows, for the pan controller,

$$\begin{cases} K_{Pp} = 0.4, K_{Pd} = 0.3, K_{Pr} = 1 \;\; if \;\; |e_P(t)| < 15^\circ \\ K_{Pp} = 0.7, K_{Pd} = 0.6, K_{Pr} = 1 \;\; if \;\; |e_P(t)| > 30^\circ \\ K_{Pp} = 1, K_{Pd} = 0.9, K_{Pr} = 1 \end{cases}$$

and for the tilt controller,

$$\begin{cases} K_{Tp} = 0.7, K_{Td} = 0.5, K_{Tr} = 1 \;\; if \;\; |e_T(t)| < 15^\circ \\ K_{Tp} = 1.1, K_{Td} = 0.8, K_{Tr} = 1 \;\; if \;\; |e_T(t)| > 30^\circ \\ K_{Tp} = 1.5, K_{Td} = 1.2, K_{Tr} = 1 \end{cases}$$

The results of the performance of the position controller are depicted in Figure 5.9 for the pan and in Figure 5.10 for the tilt actuator.

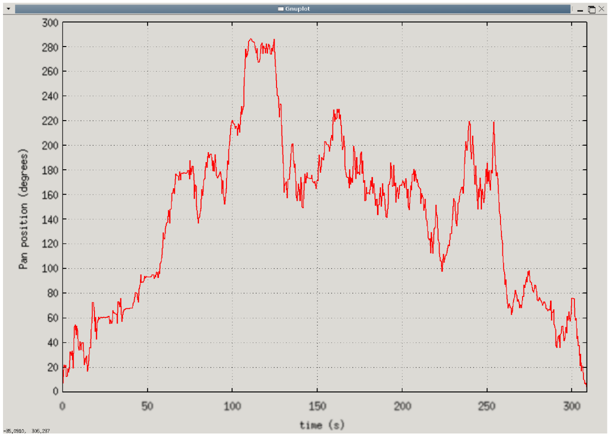
The average value for the error measured in the pan axis (see fig. 5.9(c)) consisted in a value of 5.19° with 4.98° of standard deviation. For the tilt axis (see fig. 5.10(c)) an average error of 2.40° with 3.20° of standard deviation was measured.

Figure 5.8: Aspect of the environment used for testing. The testing field is rugged, sloped and partially vegetated.

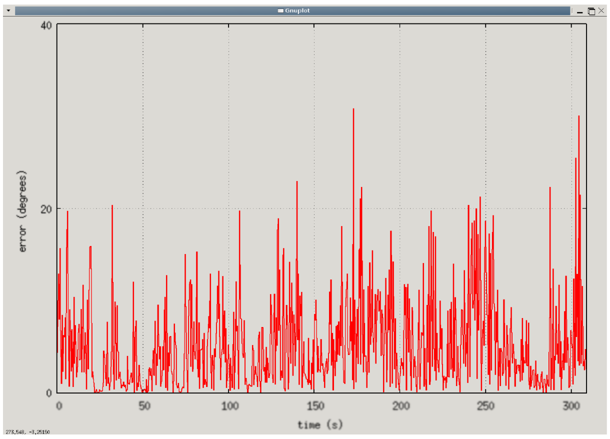
As it would be expected, results were better in a stationary environment. The first tests for the position control of the camera were elaborated in simulation. For this test, each actuator of the camera had to follow a smooth varying reference. The pan axis had to follow a 50 degree amplitude sinusoidal movement with a frequency of $74\,$mHz and the tilt axis a $63\,$mHz sinusoid with $30°$ of amplitude. The average error measured in the pan axis consisted in a value of $2.18°$ with $0.71°$ of standard deviation. For the tilt axis an average error of $1.22°$ with $0.33°$ of standard deviation was measured.

In what concerns the remaining methods, the assisted teleoperation mode revealed to be a good tool for the operator, mainly because it reduced the problem of the narrow field of view offered by the camera. The self-monitoring tool was also tested. The main disadvantage is that the operator has to stop controlling the camera, as soon as an event is deployed, or override the control of the camera in order to keep looking at the point of interest. The biggest challenge here consists in finding the trade-off between *how* to assist the operator and *when* to assist him, though, this problem will not be addressed in this dissertation.

(a) Reference for the pan actuator to follow.

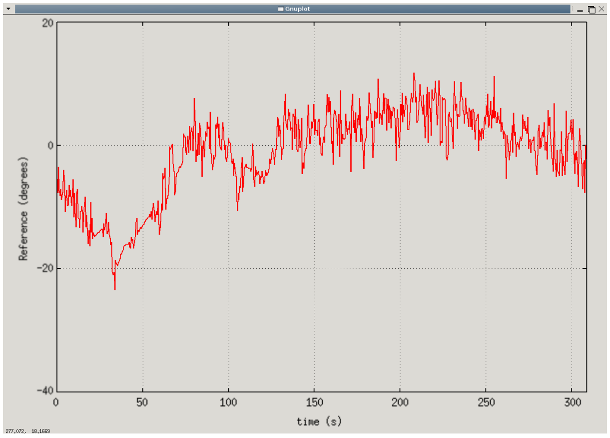


(b) Actual position of the pan actuator.

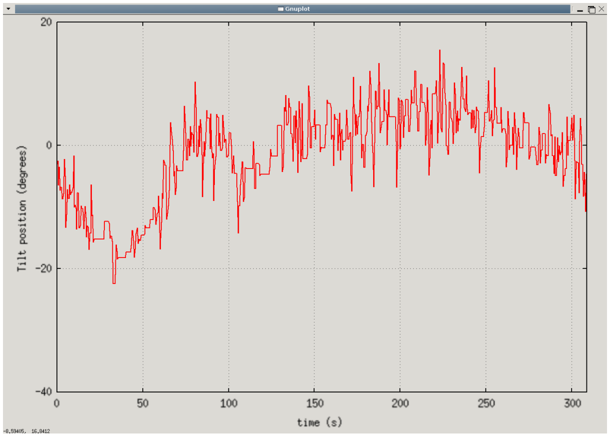


(c) Error between the reference and the actual position of the pan actuator.

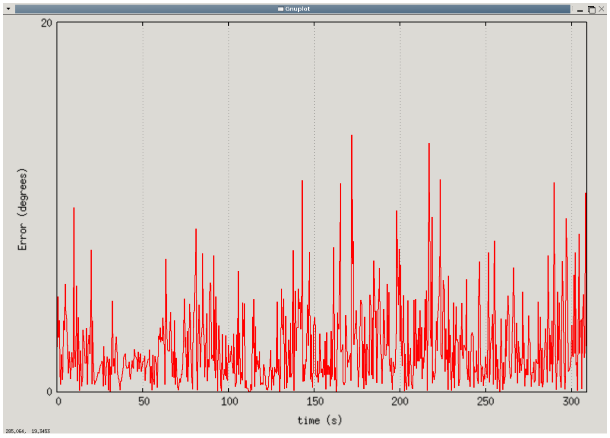
Figure 5.9: Results of the test for the position controller's accuracy in all-terrain for the pan axis.



(a) Reference for the tilt actuator to follow.



(b) Actual position of the tilt actuator.



(c) Error between the reference and the actual position of the tilt actuator.

Figure 5.10: Results of the test for the position controller's accuracy in all-terrain for the tilt axis.

Chapter 6

Conclusions and Future Work

This chapter wraps-up this dissertation providing a set of conclusions and contributions, and pointing out some directions for future research as well.

6.1 Conclusions

A wheel odometer for a 4WS mobile robot was proposed for integration in a localisation mechanism. The proposed model takes advantage of the redundancy of the 4 wheels of the robot and its final result is a weighed average given by the error (determined based on heuristics) each wheel is inducing in the motion of the robot.

The main contribution of the odometry system presented in this dissertation is the heuristic employed to reduce wheel odometry errors, which exploits in a simple way the kinematic and dynamical constraints of the robot.

Odometry, as expected, revealed to be faultier when terrains become more irregular, though, our heuristic method revealed to be robust in a complicated terrain (grass) which is extremely rough for the motors and therefore for the robot's mechanical structure.

The obtained results are quite interesting when compared to other similar wheel odometry methods [Ojeda and Borenstein, 2002] and due to its simplicity it proves to be quite affordable.

A position controller was proposed as backbone to a technique for pointing a PTZ camera according to azimuth and elevation references, followed by a mechanism to track a GPS coordinate. Typically, to tackle this type of problem, custom made cameras and controllers are used. In this case, a legacy PTZ camera was selected and the presented mechanisms are specifically designed to cope with its limitations: sluggish and error prone communication. Despite the fact that the limitations presented by this camera do not allow the exploration of other interesting visual movements (e.g. saccadic movement), the proposed controller was capable of dealing with the limitations offered by the camera.

By taking out one of the queries, the employed state prediction method allowed for the reduction of the period of a cycle in approximately 25%, approximately 200ms.

Despite having used a PTZ camera, the pointing mechanisms, as well as the developed controller, could be used to solve similar problems (e.g. antenna pointing [Bapna et al., 1998a]).

The three pointing mechanisms have shown to provide significant help to the teleoperator, though, their impact in increasing situation awareness must be further studied. The assisted navigation method revealed to be extremely useful, specially in the linear and lateral displacement modes where the camera needs to be panned more often in order for the user to have a better perspective of the trajectory of the robot. The self monitoring mechanism also revealed to be helpful to the user, specially in situations of robot collision or wheels getting stuck.

6.2 Future Work

Below, some research opportunities based on this dissertation are mentioned:

In what concerns odometry, the calibration and adaptation of wheel odometry parameters gives plenty of room for future work perspectives. Further studies can be made in order to identify parameters and the behaviours of the odometry results in each wheel, providing a way of calibrating odometry and parametrising it. Another method to improve odometry through calibration consists in doing it online, i.e. finding heuristics to associate visual odometry to wheel odometry behaviours, thus identifying the situations where odometry tends to fail, and improving it.

Another future work direction consists in finding heuristics and study their impact on their improvement of wheel odometry. For instance, finding heuristics which allow for the tuning of wheel odometry parameters according to the robot's posture.

By making a rough analysis it was possible to identify that, despite being error prone, wheel odometry is an excellent complement to visual odometry. For instance, homogeneous terrains are not proper for visual odometry but they allow for the obtention of better results with wheel odometry, though, visual odometry works better on irregular terrains, where wheel odometry tends to fail; Visual odometry fails when facing direct light, while wheel odometry functions at all times, etc. Further studies can be developed in order to study these situations and develop a proprioceptive method for locally estimating a vehicle's motion.

In what concerns the control system of the camera, it has already been mentioned that most of the studies about the control strategies presented were done recurring to virtual models. Here, several strategies were implemented in a real robot, though, no deeper analysis of the impact of the control strategies was made. An opportunity is provided here to allow the study of the presented control strategies in a real model with the hurdles it brings with it.

By doing these studies, and finding the limitations in the mechanisms presented, other methods for assisting on the operation of a remote camera can be sought in an easier way.

The impact of the presented controller can also be tested with a camera with more degrees of freedom, as well as in other fields, like high gain antenna pointing for example.

Bibliography

[pel, 2004] (2004). *"D" Protocol Manual*. Pelco, 4th edition.

[Adams and Skubic, 2005] Adams, J. and Skubic, M. (2005). Introduction to the special issue on human-robot interaction. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 35(No. 4):pages 433–437.

[Arras, 1998] Arras, K. (1998). An introduction to error propagation: Derivation, meaning and examples of equation cy= fx cx fx'. *Autonomous Systems Lab, Institute of Robotic Systems, Swiss Federal Institute of Technology Lausanne, Tech. Rep. EPFL-ASL-TR-98-01 R*, Vol. 3.

[Bapna et al., 1998a] Bapna, D., Rollins, E., Foessel, A., and Whittaker, R. (1998a). Antenna pointing for high bandwidth communications from mobile robots. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, volume 4, pages 3468–3473.

[Bapna et al., 1998b] Bapna, D., Rollins, E., Murphy, J., Maimone, E., Whittaker, W., and Wettergreen, D. (1998b). The Atacama Desert Trek: Outcomes. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, volume 1, pages 597–604.

[Baumgartner et al., 2001] Baumgartner, E., Aghazarian, H., and Trebi-Ollennu, A. (2001). Rover Localization Results for the FIDO Rover. *SPIE Photonics East Conference*, pages 28–29.

[Borenstein, 1994] Borenstein, J. (1994). Internal correction of dead-reckoning errors with the smart encoder trailer. In *Proceedings of the IEEE/RSJ/GI International Conference on*

Intelligent Robots and Systems.'Advanced Robotic Systems and the Real World', IROS'94, volume 1.

[Borenstein, 1995a] Borenstein, J. (1995a). The CLAPPER: A dual-drive mobile robot with internal correction of dead-reckoning errors. In *Proceedings of the IEEE International Conference on Robotics and Automation, 1995*, volume 3.

[Borenstein, 1995b] Borenstein, J. (1995b). Internal Correction of Dead-Reckoning Errors with a Dual-Drive Compliant Linkage Mobile Robot. *Journal of Robotic Systems*, volume 12(No. 4):pages 257–273.

[Borenstein, 1998] Borenstein, J. (1998). Experimental results from internal odometry error correction with the OmniMate mobile robot. *IEEE Transactions on Robotics and Automation*, Vol. 14(No. 6):pages 963–969.

[Borenstein et al., 1996] Borenstein, J., Everett, H., and Feng, L. (1996). Where am I? Sensors and Methods for Mobile Robot Positioning. *University of Michigan*.

[Borenstein and Feng, 1994] Borenstein, J. and Feng, L. (1994). UMBmark: a method for measuring, comparing, and correcting dead-reckoning errors in mobile robots. *Tech. report*.

[Borenstein and Feng, 1995] Borenstein, J. and Feng, L. (1995). Correction of systematic odometry errors in mobile robots. In *Proceedings of the International Conference on Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots'*, volume 3.

[Borenstein and Koren, 1987] Borenstein, J. and Koren, Y. (1987). Motion Control Analysis of a Mobile Robot. *Transactions of ASME, Journal of Dynamics, Measurement and Control*, Vol. 109(No. 2):pages 73–79.

[Bowman et al., 1997] Bowman, D., Koller, D., and Hodges, L. (1997). Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*.

[Breitfelder and Messina, 2000] Breitfelder, K. and Messina, D. (2000). The Authoritative Dictionary of IEEE Standards Terms.

[Caltabiano et al., 2004] Caltabiano, D., Muscato, G., and Russo, F. (2004). Localization and self-calibration of a robot for volcano exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'04*, volume 1.

[Cândido et al., 2008] Cândido, C., Santana, P., Correia, L., and Barata, J. (2008). Shared Control of a Pan-Tilt Camera on an All-terrain Mobile Robot. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA'08*.

[Carlisle, 1983] Carlisle, B. (1983). An Omni-directional mobile robot. *Developments in Robotics*, pages 79–88.

[Carlson and Murphy, 2005] Carlson, J. and Murphy, R. (2005). How UGVs physically fail in the field. *IEEE Transactions on Robotics [see also IEEE Transactions on Robotics and Automation]*, Vol. 21(No. 3):pages 423–437.

[Chong and Kleeman, 1997] Chong, K. and Kleeman, L. (1997). Accurate odometry and error modelling for a mobile robot. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, volume 4.

[Conway et al., 1990] Conway, L., Volz, R., and Walker, M. (1990). Teleautonomous systems: projecting and coordinating intelligent action at a distance. *IEEE Transactions on Robotics and Automation*, Vol. 6(No. 2):pages 146–158.

[Cruz et al., 2005] Cruz, H., Lisboa, J., Santana, P., Maltez, R., Barata, J., and Flores, L. (2005). Two sustainable and compliant robots for humanitarian demining. In *Proceedings of the IARP International Workshop on Robotics and Mechanical Assistance in Humanitarian Demining (HUDEM2005)*, pages 64–69.

[Doh et al., 2003] Doh, N., Choset, H., and Chung, W. (2003). Accurate relative localization using odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2.

[Dubov, 1996] Dubov, D. (1996). *Sojourner Rover Home Page [Web Site].* Retrieved Aug 2008 from the World Wide Web: http://mars.jpl.nasa.gov/MPF/rover/sojourner.html.

[Elfes, 1987] Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation [legacy, pre-1988]*, Vol. 3(No. 3):pages 249–265.

[Fan et al., 1995] Fan, Z., Borenstein, J., Wehe, D., and Koren, Y. (1995). Experimental Evaluation of an Encoder Trailer for Dead-reckoning in Tracked Mobile Robots. In *Proceedings of the 1995 IEEE International Symposium on Intelligent Control*, pages 571–576.

[Feng et al., 1993] Feng, L., Koren, Y., and Borenstein, J. (1993). Cross-coupling motion controller for mobile robots. *Control Systems Magazine, IEEE*, Vol. 13(No. 6):pages 35–43.

[Fong and Thorpe, 2001] Fong, T. and Thorpe, C. (2001). Vehicle Teleoperation Interfaces. *Autonomous Robots*, Vol. 11(No. 1):pages 9–18.

[Goel et al., 1999] Goel, P., Roumeliotis, S., and Sukhatme, G. (1999). Robust localization using relative and absolute position estimates. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999. IROS'99*, volume 2.

[Hanson and Wernert, 1997] Hanson, A. and Wernert, E. (1997). Constrained 3D navigation with 2D controllers. In *Proceedings of Visualization '97*, pages 175–182.

[Hanson et al., 1999] Hanson, A., Wernert, E., and Hughes, S. (1999). Constrained Navigation Environments. *Scientific Visualization: Dagstuhl*, Vol. 97:pages 95–104.

[Hongo et al., 1987] Hongo, T., Arakawa, H., Sugimoto, G., Tange, K., and Yamamoto, Y. (1987). An automatic guidance system of a self-controlled vehicle. *IEEE Transactions on Industrial Electronics*, Vol. IE-34(No. 1):pages 5–10.

[Hughes, 2002] Hughes, S. (2002). Shared Viewpoint Control for Information Visualization. *Workshop on Visual Computing at Distributed Multimedia Systems*.

[Hughes and Lewis, 2000] Hughes, S. and Lewis, M. (2000). Attentive camera navigation in virtual environments. *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 2.

[Hughes and Lewis, 2002] Hughes, S. and Lewis, M. (2002). Attentive Interaction Techniques for Searching Virtual Environments. *Human Factors and Ergonomics Society's 46th Annual Meeting*.

[Hughes and Lewis, 2004] Hughes, S. and Lewis, M. (2004). Robotic camera control for remote exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 511–517. ACM New York, NY, USA.

[Hughes and Lewis, 2005] Hughes, S. and Lewis, M. (2005). Task-driven camera operations for robotic exploration. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 35(No. 4):pages 513–522.

[Hughes et al., 2003] Hughes, S., Manojlovich, J., Lewis, M., and Gennari, J. (2003). Camera control and decoupled motion for teleoperation. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2003*, volume 2, pages 1339–1344.

[Kim et al., 2004] Kim, H., Choi, W., Kang, H., and Lee, K. (2004). State estimation for autonomous guided vehicle using the extended kalman filter. In *Proceedings of the 5th Asian Control Conference*.

[Konolige et al., 2006] Konolige, K., Agrawal, M., Bolles, R. C., Cowan, C., Fischler, M., and Gerkey, B. (2006). Outdoor mapping and navigation using stereo vision. In *Proceedings of the International Symposium on Experimental Robotics, Brazil*.

[Kurazume et al., 1994] Kurazume, R., Nagata, S., and Hirose, S. (1994). Cooperative positioning with multiple robots. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 1250–1257 Vol.2.

[Larsen et al., 1998] Larsen, T., Bak, M., Andersen, N., and Ravn, O. (1998). Location Estimation for an Autonomously Guided Vehicle using an Augmented Kalman Filter to Autocalibrate the Odometry. *First International Conference on Multisource-Multisensor Information Fusion 1998, FUSION'98*.

[Leith and Leithead, 2000] Leith, D. and Leithead, W. (2000). Survey of gain-scheduling analysis & design. *International Journal of Control*, Vol. 73(No. 11):pages 1001–1025.

[Manduchi et al., 2005] Manduchi, R., Castano, A., Talukder, A., and Matthies, L. (2005). Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation. *Autonomous Robots*, Vol. 18,(No. 1):pages 81–102.

[Martinelli and Siegwart, 2003] Martinelli, A. and Siegwart, R. (2003). Estimating the Odometry Error of a Mobile Robot during Navigation. In *Proceedings of European Conference on Mobile Robots, ECMR'03*, volume 1.

[McGovern, 1993] McGovern, D. (1993). Experience and results in teleoperation of land vehicles. *Pictorial communication in virtual and real environments*, pages 182–195.

[Miller, 2008] Miller, D. (2008). *The Rocky Micro-Rovers [Web Site]*. Retrieved Aug 2008 from the World Wide Web: http://www.kipr.org/robots/rocky.html.

[Murray et al., 1993] Murray, D., McLauchlan, P., Reid, I., and Sharkey, P. (1993). Reactions to peripheral image motion using a head/eye platform. In *Proceedings of the Fourth International Conference on Computer Vision*, pages 403–411.

[Nielsen et al., 2007] Nielsen, C., Goodrich, M., and Ricks, R. (2007). Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics [see also IEEE Transactions on Robotics and Automation]*, Vol. 23(No. 5):pages 927–941.

[Nielsen et al., 2005] Nielsen, C., Goodrich, M., and Rupper, R. (2005). Towards facilitating the use of a pan-tilt camera on a mobile robot. *IEEE International Workshop on Robot and Human Interactive Communication, 2005. ROMAN'05*, pages 568–573.

[Ojeda and Borenstein, 2002] Ojeda, L. and Borenstein, J. (2002). FLEXnav: fuzzy logic expert rule-based position estimation for mobile robots on rugged terrain. *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA '02*, 1:317–322 vol.1.

[Ojeda and Borenstein, 2004] Ojeda, L. and Borenstein, J. (2004). Methods for the Reduction of Odometry Errors in Over-Constrained Mobile Robots. *Autonomous Robots*, Vol. 16(No. 3):pages 273–286.

[Ojeda et al., 2006] Ojeda, L., Reina, G., Cruz, D., and Borenstein, J. (2006). The FLEXnav precision dead-reckoning system. *International Journal of Vehicle Autonomous Systems*, Vol. 4(No. 2):pages 173–195.

[Olson, 2006] Olson, E. (2006). A primer on odometry and motor control. Technical report, Massachusetts Institute of Technology.

[Powell et al., 2006] Powell, M., Crockett, T., Fox, J., Joswig, J., Norris, J., and Rabe, K. (2006). Targeting and localization for mars rover operations. In *Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration*, pages 23–27.

[Proetzsch et al., 2005] Proetzsch, M., Luksch, T., and Berns, K. (2005). Fault-tolerant behaviour-based motion control for offroad navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4697–4702.

[Santana, 2005] Santana, P. (2005). Survival Kit: A bottom layer for robot navigation. Master's thesis, Computer Science Department, Faculty of Sciences and Technology, New University of Lisbon, Portugal.

[Santana et al., 2007] Santana, P., Barata, J., and Correia, L. (2007). Sustainable robots for humanitarian demining. *International Journal of Advanced Robotics Systems*, Vol. 4(No. 2).

[Santana et al., 2008a] Santana, P., Cândido, C., Santos, P., Almeida, L., Correia, L., and Barata, J. (2008a). The Ares Robot: Case Study of an Affordable Service Robot. In *Proceedings of the 2nd European Robotics Symposium (EUROS 2008)*. Springer.

[Santana et al., 2006] Santana, P., Cândido, C., Santos, V., and Barata, J. (2006). A Motion Controller for Compliant Four-Wheel-Steering Robots. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO'06), Kunming, China*.

[Santana and Correia, 2008] Santana, P. and Correia, L. (2008). Improving visual odometry by removing outliers in optic flow. In *Proceedings of the 8th Conference on Autonomous Robot Systems and Competitions*, Aveiro.

[Santana et al., 2008b] Santana, P., Correia, L., and Barata, J. (2008b). Developments on an affordable robotic system for humanitarian demining. In *Humanitarian Demining: Innovative Solutions and the Challenges of Technology*. Advanced Robotic Systems.

[Santos et al., 2007] Santos, V., Cândido, C., Santana, P., Correia, L., and Barata, J. (2007). Developments on a system for human-robot teams. In *Proceedings of Scientific Conference of the 7th Edition of the National Robotics Festival (Robótica '07)*.

[Schäfer and Berns, 2006] Schäfer, H. and Berns, K. (2006). RAVON - An autonomous Vehicle for Risky Intervention and Surveillance. In *International Workshop on Robotics for risky intervention and environmental surveillance-RISE*.

[Schiehlen and Dickmanns, 1994] Schiehlen, J. and Dickmanns, E. (1994). Design and control of a camera platform for machine vision. In *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94*, volume 3, pages 2058–2063.

[Scholtz et al., 2004] Scholtz, J., Young, J., Drury, J., and Yanco, H. (2004). Evaluation of human-robot interaction awareness in search and rescue. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2004*, volume 3, pages 2327–2332.

[Shamah, 1999] Shamah, B. (1999). Experimental comparison of skid steering vs. explicit steering for a wheeled mobile robot. Master's thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

[Siegwart and Nourbakhsh, 2004] Siegwart, R. and Nourbakhsh, I. (2004). *Introduction to Autonomous Mobile Robots*. MIT Press.

[Spentzas et al., 2001] Spentzas, K., Alkhazali, I., and Demic, M. (2001). kinematics of four-wheel-steering vehicles. *Forschung im Ingenieurwesen*, pages 211–216.

[Sugiyama, 1993] Sugiyama, H. (1993). A method for an autonomous mobile robot to recognize its position in the global coordinate system when building a map. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems '93, IROS'93*, volume 3, pages 2186–2191.

[Thorndyke and Hayes-Roth, 1982] Thorndyke, P. and Hayes-Roth, B. (1982). Differences in spatial knowledge acquired from maps and navigation. *Cognitive Psychology*, Vol. 14(No. 4):pages 560–89.

[Voshell and Woods, 2005] Voshell, M. G. and Woods, D. D. (2005). Breaking the keyhole in human-robot coordination: Method and evaluation. Technical report, Ohio State Univ.

[Wang, 1988] Wang, C. (1988). Location estimation and uncertainty analysis for mobile robots. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pages 1231–1235.

[Wernert and Hanson, 1999] Wernert, E. A. and Hanson, A. J. (1999). A framework for assisted exploration with collaboration. In Ebert, D., Gross, M., and Hamann, B., editors, *IEEE Visualization '99*, pages 241–248, San Francisco.

[Williamson, 2002] Williamson, M. (2002). Man on the moon: the technology of lunar exploration. *Engineering Science and Education Journal*, Vol. 11(No. 6):pages 217–226.