CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY IN PRAGUE

MASTER'S THESIS

Data Fusion for a Mobile Exploratory Robot

Vladimír Kubelka

vladimir.kubelka@fel.cvut.cz

May 9, 2013

Thesis Supervisor: Michal Reinštein

ETH Advisors: Francis Colas, François Pomerleau

This thesis was supported by the European research project FP7-ICT-247870 NIFTi

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Data Fusion for a Mobile Exploratory Robot

Vladimír Kubelka

May 9, 2013

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická
Katedra měření

Akademický rok **2012-2013**

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	**Bc. Vladimír Kubelka**
Studijní program: Obor:	**Kybernetika a robotika** **Letecké a kosmické systémy**
Název tématu česky:	**Fúze dat pro mobilního průzkumného robota**
Název tématu anglicky:	**Data Fusion for a Mobile Exploratory Robot**

Pokyny pro vypracování:

Navrhněte, implementujte a experimentálně ověřte algoritmus pro fúzi navigačních dat získaných z následujících zdrojů: inerciální měřicí jednotka Xsens Mti-G, odometrie pro pásového robota, ICP algoritmus zpracovávající 3D laserové skeny měřené pomocí rotujícího SICK LMS-151 a algoritmus vizuální odometrie využívající panoramatické snímky ze všesměrové kamery Ladybug 3. Všechny jmenované senzory jsou umístěny na mobilním průzkumném robotu vyvíjeném v rámci projektu NIFTi (http://www.nifti.eu/). Pro řešení použijte multi-model EKF s rozhodováním. Výsledná implementace je požadována pro Robotic Operation System (http://www.ros.org/wiki/) v reálném čase a v jazyce C++. K ověření matematických modelů je doporučeno prostředí MATLAB. Experimentální ověření proběhne jak ve strukturovaném venkovním prostředí, tak i s využitím systému Vicon (ASL Lab, ETH Zurich). Téma je vypsáno ve spolupráci s laboratoří ASL, ETH Zurich v rámci projektu NIFTi.

Seznam odborné literatury:

[1] Oswald L.: Robust Localization for Search & Rescue Robots; Semester-Thesis; ETH Zurich; 2012

[2] Trawny, N.: Roumeliotis, S. I., Indirect Kalman Filter for 3D Attitude Estimation, University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, March 2005

[3] Weiss, S.: Siegwart, R.: Real-time metric state estimation for modular vision-inertial systems, IEEE International Conference on Robotics and Automation (ICRA), 2011, pp.4531-4537, May 2011

Vedoucí diplomové práce:	Ing. Michal Reinštein, Ph.D. (K 13133)
Datum zadání diplomové práce:	29. listopadu 2012
Platnost zadání do[1]:	30. června 2014

Haasz

Prof. Ing. Vladimír Haasz, CSc.
vedoucí katedry

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
L.S.
-4-

v.z. M. Demlová

Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 29. 11. 2012

[1] Platnost zadání je omezena na dobu tří následujících semestrů.

Acknowledgment

I would like to thank to my thesis supervisor Michal Reinštein and to my ETH advisors Francis Colas and François Pomerleau for their invaluable advices concerning the optimal estimation issues and for their support during my stay at the ETH. I would also like to thank to Tomáš Svoboda for helping me with the formal matters concerning the stay and also to my colleague at ETH, Lorenz Oswald for assisting me during the experiments making the work flow much faster. Last but not least, I thank to my dear parents and to my beautiful and smart sister because with their support, I could focus on things I enjoyed doing past 6 years.

Declaration

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis.

Abstract

A robust localization subsystem is a vital part of a mobile robot system; many high-level functionalities depend on it (mapping, autonomous navigation, task planning, sensor data postprocessing etc.) We propose a localization system for an Urban Search&Rescue robot being developed as a part of the European research project NIFTi. We are aiming for a higher grade of accuracy, fusing several sensor modalities to combine their strong points. This fusion is done by means of an error state Extended Kalman Filter and by advanced measurement preprocessing to ensure suppression of the drift of the sensor modalities world coordinate frames. The proposed algorithm has been extensively tested both by indoor and outdoor experiments (over 4 kilometers traveled by the robot in demanding 3D environments with a high-precision reference). Finally, to discover the true limits of the sensor modalities under realistic failure conditions, several fail-case experiments have been performed and analyzed.

Abstrakt

Robustní lokalizace je jeden ze základních systémů každého mobilního robota, mnoho dalších funkcí na ní závisí (mapování, navigace, plánování, pokročilé zpracování senzorických dat . . .) V této práci jsme navrhli systém lokalizace pro průzkumného a záchranného mobilního robota vyvíjeného jako součást evropského výzkumného projektu NIFTi. Tento systém usiluje o vyšší míru přesnosti fúzí dat z různých senzorů, využívaje jejich silné stránky. Tato fúze je realizována pomocí algoritmu Extended Kalman Filter, pracujícího s chybovým modelem systému, a pomocí pokročilého zpracování měření, které potlačuje drift souřadných systémů jednotlivých senzorických modalit. Algoritmus byl ověřen rozsáhlým experimentováním ve vnitřních i venkovních prostorech, robot urazil více než 4 kilometry v náročných podmínkách, to vše s vysoce přesnou referencí. Nakonec bylo pro zjištění skutečných možností jednotlivých senzorických modalit provedeno několik experimentů simulujících podmínky, které způsobovaly jejich selhání.

Abbreviations

These are abbreviations used thorough this document:

R	Robot coordinate frame
I	Inertial Measurement Unit coordinate frame
N	Navigation coordinate frame
O	Odometry coordinate frame
ROS	The Robot Operating System (`www.ros.org`)
SO(3)	3D rotation group
JPL	Jet Propulsion Laboratory (USA)
ICP	Iterative Closest Point
ODOM	(Caterpillar Tracks) Odometry
VODOM	Visual Odometry
IMU	Inertial Measurement Unit
USAR	Urban Search and Rescue (missions)

Notation Convention

Basic notation conventions used to distinguish scalars and vectors and to notate rotations and various multiplications:

x	Scalar x
$\mathbf{x}$	Vector $\mathbf{x} = [x_1, x_2, x_3]^T$
$\hat{\mathbf{x}}$	Estimate of $\mathbf{x}$
$\dot{\mathbf{x}}$	Time derivative of $\mathbf{x}$
$\Delta\mathbf{x}$	Error associated with the vector $\mathbf{x}$
$\lvert\lvert\mathbf{x}\rvert\rvert$	Norm of the vector $\mathbf{x}$
$\mathbf{q}_A^B$	Quaternion $\mathbf{q} = [\vec{q}^T, w]^T = [x, y, z, w]^T$ expressing a rotation from coordinate frame A to frame B
$(\mathbf{q}_A^B)^{-1}$	Quaternion inversion of $\mathbf{q}_A^B$ defined in the appendix B
$\Re(\mathbf{q}_A^B)$	A real part of the quaternion
$C_{(\mathbf{q})}$	3×3 Matrix $C \in \mathrm{SO}(3)$ expressing the same rotation the quaternion $\mathbf{q}$ does, see eq. (A.10)
C_A^B	3×3 Matrix $C \in \mathrm{SO}(3)$ expressing a rotation from the coordinate frame A to the coordinate frame B
$b_{x,A}$	Bias of the value x expressed in the A coordinate frame
$s_{x,A}$	Scale of the value x expressed in the A coordinate frame
×	Vector multiplication
$\lfloor\mathbf{x}\rfloor$	Skew form of the vector $\mathbf{x}$ substituting vector multiplication: $\lfloor\mathbf{x}\rfloor\mathbf{y} = \left(\begin{smallmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{smallmatrix}\right)\mathbf{y} = \mathbf{x} \times \mathbf{y}$
$\otimes$	Quaternion multiplication following the JPL quaternion multiplication proposal [1], definition in the appendix B
$\Omega(\omega)$	Matrix used in the quaternion derivative [2]: $\Omega(\omega) = \left(\begin{smallmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{smallmatrix}\right)$
$\emptyset_N$	$N \times N$ zero matrix
$\emptyset_{N \times M}$	$N \times M$ zero matrix
I_N	$N \times N$ identity matrix

List of Symbols

These are symbols frequently used in equations thorough this document:

$\mathbf{p}_N$, $\Delta\mathbf{p}_N$	Position expressed in the N coordinate frame and its corresponding error
$\mathbf{q}_N^R$	Rotation quaternion expressing current attitude as a rotation from the N frame to the R frame
$\delta\theta$	Error in attitude expressed as a rotation vector
$\mathbf{v}_R$, $\Delta\mathbf{v}_R$	Velocity expressed in the R coordinate frame and its corresponding error
ω_R, $\Delta\omega_R$	Angular rate expressed in the R coordinate frame and its corresponding error
$\mathbf{f}_R$, $\Delta\mathbf{f}_R$	Specific force [3] measured in the I coordinate frame and expressed in the R coordinate frame, its corresponding error
λ, $\Delta\lambda$	Visual odometry scale and its corresponding error
$\mathbf{b}_{\omega,I}$, $\Delta\mathbf{b}_{\omega,I}$	Bias of the angular rate sensors expressed in the I coordinate frame and its corresponding error
$\mathbf{b}_{f,I}$, $\Delta\mathbf{b}_{f,I}$	Bias of the accelerometers expressed in the I coordinate frame and its corresponding error
β, $\Delta\beta$	Odometry frame pitch (negative rotation around the y axis in the North-West-Up convention) correction angle and its corresponding error
$\mathbf{g}_N$	Gravitational acceleration expressed in the N frame; following our coordinate frame axes convention (North-West-Up), it is $\mathbf{g}_N = [0, 0, g]^T$ where g is approximately 9.81ms^{-2}, depending on latitude and longitude
F_c	Continuous time state transition matrix
F_d	Discrete time state transition matrix
G_c	Noise coupling matrix
Q_c	System noise covariance matrix (continuous time form)
Q_d	System noise covariance matrix (discrete time form)
$\mathbf{n}_{(\cdot)}$	Continuous time noise term (a random variable with normal distribution and zero mean), associated with a state denoted in the subscript
$\sigma_{(\cdot)}$	Standard deviation of the noise term associated with the state denoted in the subscript
$\mathbf{y}_{(\cdot)}$,$\hat{\mathbf{y}}_{(\cdot)}$,$\Delta\mathbf{y}_{(\cdot)}$	Measurement vector, predicted measurement vector and innovation (or *measurement residual*). All associated with measured value denoted in the subscript
$h(\mathbf{x})$,$h'(\Delta\mathbf{x})$	Measurement and innovation functions $\mathbf{y} = h(\mathbf{x})$, $\Delta\mathbf{y} \approx h'(\Delta\mathbf{x})$
H	Innovation matrix
$\mathbf{m}_{(\cdot)}$	Continuous time measurement noise term (a random variable with normal distribution and zero mean), associated with a measured value denoted in the subscript

Contents

1. Introduction

The aim of this work was to design, implement and experimentally test a navigation data fusion algorithm for an Urban Search&Rescue (USAR) robotic platform being developed as a part of the European research project NIFTi (see section 1.2). The purpose of the fusion system is a robust localization of the NIFTi robot in harsh environments the Search&Rescue missions typically take place in.

Robust localization and attitude determination with a sufficiently high rate is a crucial component of the robot system since many high-level functionalities rely on it (autonomous navigation or automatic return to the initial position, real-time 3D map creation, obstacle traversing etc.) The current state of the art in localization and navigation algorithms in the mobile robotics exploit many sensors available; these sensors offer rich scale of information about the robot state (interoceptive sensors - angular rate encoders attached to wheels or tracks, sensors monitoring states of joints in the robot's body, force and pressure sensors, inertial and magnetic sensors etc.) and about the robot's surroundings (exterioceptive sensors - various cameras: omni-directional, stereo, IR, . . . ; laser range finders, sonars, radars etc.). Advanced miniaturization, increasing computational performance and low power consumption of current CPUs supported by parallelization offered by GPUs enables all the sensor data to be processed by the robot's system yet still leaving space for higher-level tasks. That opens a new area of research concerning the fusion of all sensor data in such a way that would be robust, multi-modal and exploiting strong points of the sensors.

Our system aims for higher order of accuracy while ensuring correct operation in rough terrains including overcoming obstacles by fusing inertial measurements, odometric data, visual odometry (3.4) and laser scanner data processed by the ICP algorithm (3.3). The core of the fusion is realised by an error state Extended Kalman Filter (EKF) inspired by [4, 5]. Since the NIFTi project robot was designed to operate in unstructured environments such as collapsed buildings or tunnels blocked by mass car accidents, localization *in all three dimensions* is vital for any realistic application. The fusion system should be capable to optimally fuse the selected localization information sources enhancing the overall accuracy. These requirements have been extensively experimentally verified (4 kilometers of distance traveled in total with a high-precision reference).

This document is structured as follows: in the three following sections, the state of the art in the localization sources is discussed more in detail (1.1) and the NIFTi robotic platform is introduced (1.2) as well as the reference systems used (1.3) during the experiments. In the following chapters, mathematical models describe the robot's dynamics (2) and the measurement preprocessing system (3). Then, the error state EKF fusion algorithm is described (4) and tested on a large dataset of experiments (5). Finally, the results are summed up in the last chapter (6).

1.1. State of the Art

The basic approach to localization in mobile robotics is a combination of wheel or track velocity sensors with inertial measurements (angular rates and specific forces [3]). This approach produces localization data with a high rate and it is relatively computationally undemanding. However, as a dead-reckoning approach, it is prone to drift in position and the heading angle (only the roll and pitch angles are observable [6] under the condition of no retentively accelerated motion). In the case of skid-steered robots, the situation is complicated by systematic inaccuracy in the velocity measurements [7, 8] caused by slippage of the wheel or tracks. From these reasons, it is desirable to combine the dead-reckoning approach with an exteroceptive sensors to suppress these drifts.

Precision of the localization can be improved by introducing camera and computer vision algorithms resulting into visual odometry [9, 10] - an algorithm estimating consecutive camera poses in space based on image features (e.g. corners) co-observed by the camera from these poses. The visual odometry algorithm possibilities have been extensively investigated since the Spirit and Opportunity Mars rover mission. The camera types vary from a single lens cameras to multi-lens omni-directional cameras. Determined by the geometry of the task, the visual odometry exploiting images captured by camera assembly with only one central point (concerns the NIFTi robot omni-camera as well, Fig. 3.11) can determine positions of the camera only up-to-scale. To acquire metric information, at least stereo-camera or a combination of a normal camera and an IMU must be used. Independent on the camera type, visual odometry is not drift-free, yet the state-of-the-art solutions reach accuracies below 1% of the final position error relatively to the distance traveled. As mentioned earlier, applying these algorithms in the mobile robotics was enabled by advance in computer technology, nevertheless, more sophisticated variations as simultaneous localization and mapping using computer vision is still difficult to implement on-board complying the weight and power consumption requirements laid by mobile robotics.

Other sensors providing exteroceptive measurements are 3D scanners, whose miniaturization enables them to replace sonars offering incomparably greater accuracy. Commonly used 3D scanners can be divided into two main groups. The first group (e.g. Microsoft Kinect) projects structured light on the observed scene and combined with computer vision techniques as depth from focus or depth from stereo, it provides standard color images enhanced with known depth of each pixel. An advantage of this approach is a possibility of capturing a whole scene in one shot and therefore, the only factor limiting attainable scanning frequency is the computational hardware performance. The main drawback is the maximal range of the sensor, which is limited by reach of the structured light emitter. Also, since the structured light operates at IR wavelengths, it gets easily saturated by sunshine.

The second group - the 3D scanners estimate depth from the time a laser beam travels between the scanner and the scanned object. These scanners show better performance outdoors and their operational range is superior compared to the previous group. Their drawback is a limited number of laser beams that are used in one time instant. Typically, it is not more than 64 (Velodyne® HDL-64E, Fig. 1.1 and 1.2), the *NIFTi* robot is equipped with a *SICK LMS-151* laser scanner (Fig. 1.4) emitting only one beam. To obtain an omni-directional 3D scan, the laser beam is swept around by a high-velocity revolving mirror (inner functionality of the sensor) scanning 3D points lying in a plane, and the whole scanner has to be rotated in a perpendicular sense to the mirror rotation to obtain a full 3D scan of the surrounding space (Fig. 1.5). Due to the physical limitations of the servo which rotates the whole 3D sensor, the minimal time period between two consecutive scans is in the order of seconds (3 seconds for the NIFTi robot 3D scanner assembly). Laser scanners emitting several beams achieve higher scanning frequencies, yet, the scanned point cloud is composed of several planes corresponding to each laser beam. To create a denser 3D scan, the whole 3D scanner can be periodically tilted achieving similar results as the one-laser-beam scanners, but with higher frequency.

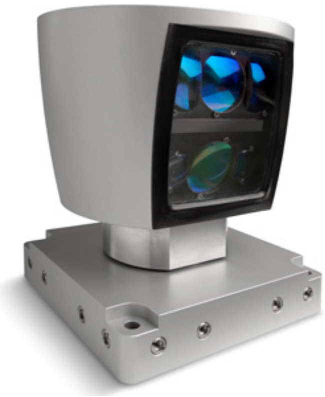


Figure 1.1. The Velodyne® HDL-64E laser radar (*lidar*). The sensor head emits and receives 64 laser beams while rotating about its axis permitting up to 15 complete 3D scans of its surroundings per second. Source: http://velodynelidar.com/

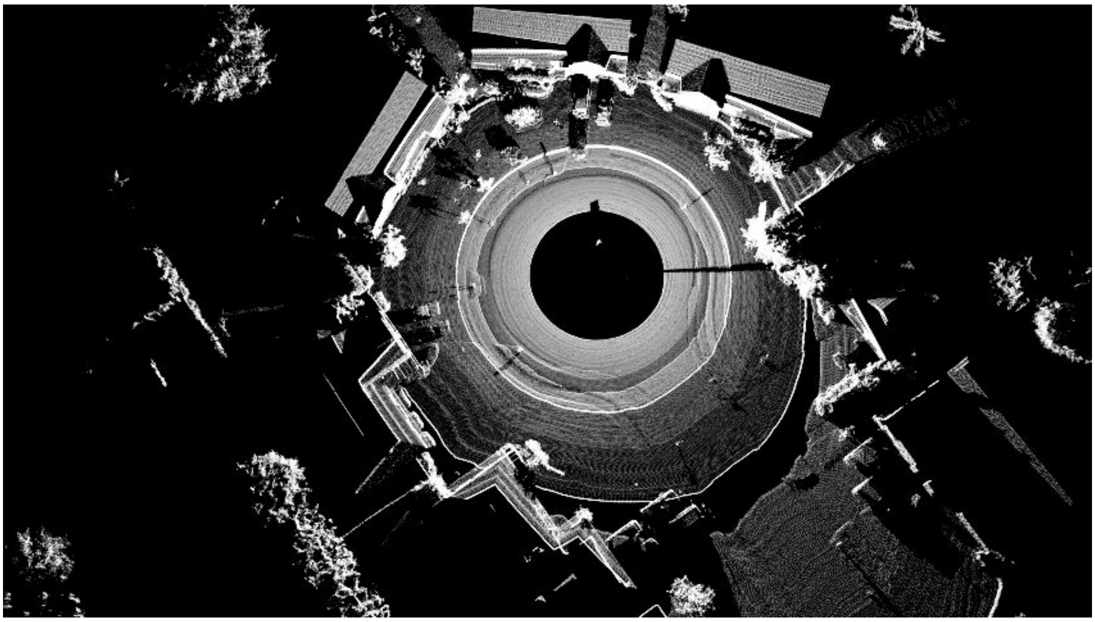


Figure 1.2. A Velodyne® HDL-64E lidar single 3D scan of a suburb street. Source: velodynelidar.com

The algorithm processing the 3D data is called ICP (Iterative Closest Point) odometry [11]. The ICP odometry iteratively matches new 3D pointclouds on the previous ones creating a 3D map of the robot surroundings. Apart from the (monocular) visual odometry, the resulting position estimates include metric scale. Correct function of the algorithm is conditioned by sufficient amount of 3D features reachable by the 3D scanner (i.e. large flat areas pose a problem), however, the NIFTi robot deployment scenarios comply this requirement.

The output of the ICP algorithm is position relative to its inner 3D pointcloud map. The main issue observed with the NIFTi robot is that it does not necessarily create a *correct* map - while locally correct, the map as a whole tends to slightly twist or bend.

That causes a problem: the position and the attitude estimated by the ICP odometry collide with other position information sources (visual odometry, track or wheel odometry etc.). For aerial robots, Weiss [4] brings an idea of augmenting a system state held by the fusion system with rigid transformations between these maps (coordinate frames), but according to observability study he carried out, additional sensors are needed to make these rigid transformations observable. Also, as our experiments have shown, maps created by different localization systems do not necessarily differ only by a rigid transformation, more complex deformations were observed (e.g. bending and twisting) and it is not clear how would a local approximation by rigid transformation affect the overall performance. From that reason, we have proposed to treat the ICP and visual odometries as velocity rather than absolute position estimating algorithms (3.3).

1.2. NIFTi Search&Rescue Robotic Platform

The robotic platform (Fig. 1.3) designed by a Swiss company BlueBotics has been developed as a part of the European USAR research project NIFTi (www.nifti.eu). The goal of the project is to investigate and develop means of a human-robot cooperation; the robotic platform is used to test these new approaches experimentally. It was designed to search at disaster or accident sites for victims and threads to the rescue personnel. It cooperates with a UAV (unmanned aerial vehicle) communicating, among others, each others position and thus, the localization subsystem is a vital part of the whole system.

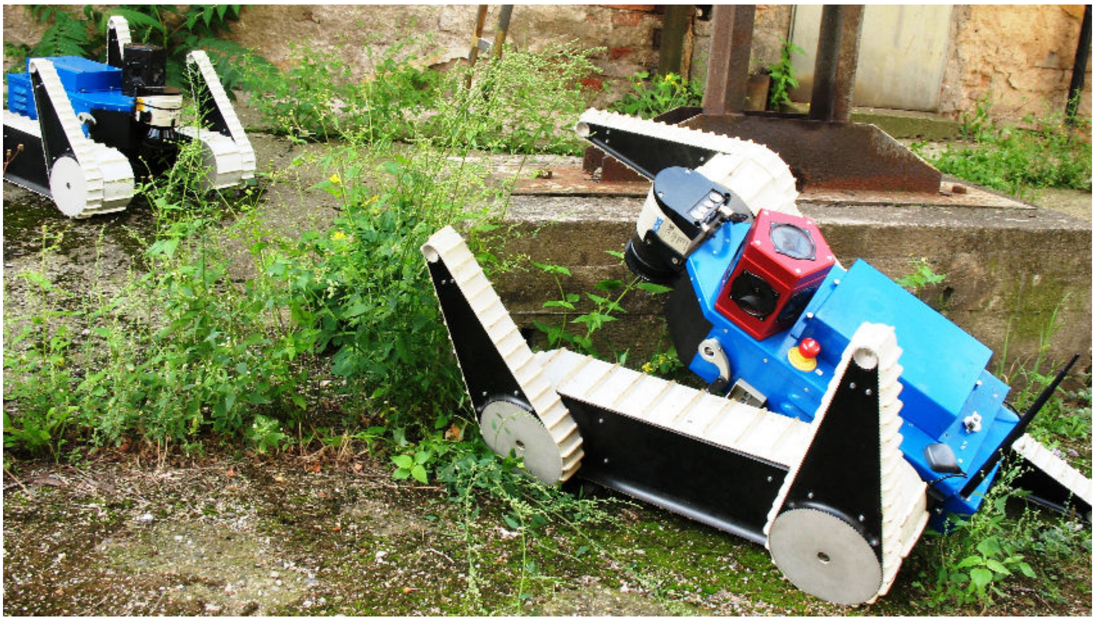


Figure 1.3. A prototype of the robotic platform Absolem; designed by BlueBotics (www.bluebotics.ch) and developed by the NIFTi project.

The robot is equipped with the SICK LMS-151 laser scanner (Fig. 1.4,1.5), the Point-Grey Ladybug®3 omni-directional camera (Fig. 3.11), the X-Sens MTI-G inertial measurement unit (IMU) and various internal state sensors (motor velocities sensors, angle sensors measuring posture of joints, temperature sensors, . . .). The computational power is delivered by the Intel® Core$^{\mathrm{TM}}$2 Quad Mobile processor supported by an embedded Kontron® PC. Connection to a human operator is realized by a dedicated WiFi link, it can be teleoperated 4 hours using a single battery, which can be replaced without turning the robot off when depleted.

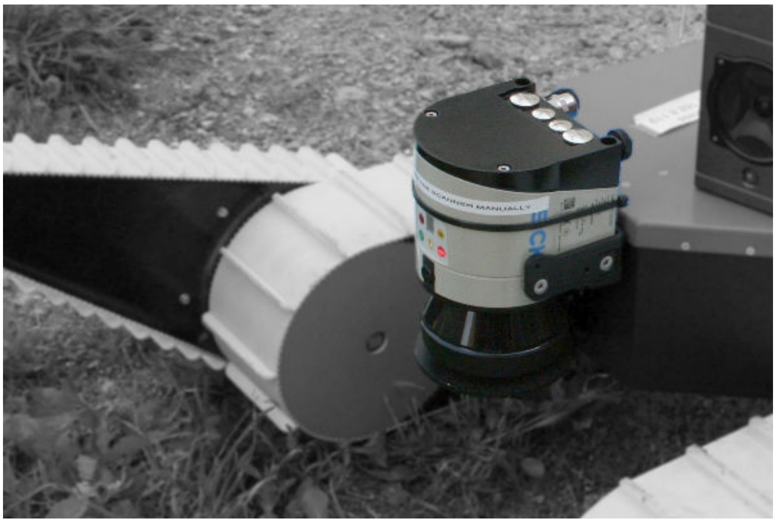


Figure 1.4. The SICK LMS-151 lidar attached to the *NIFTi* robot.

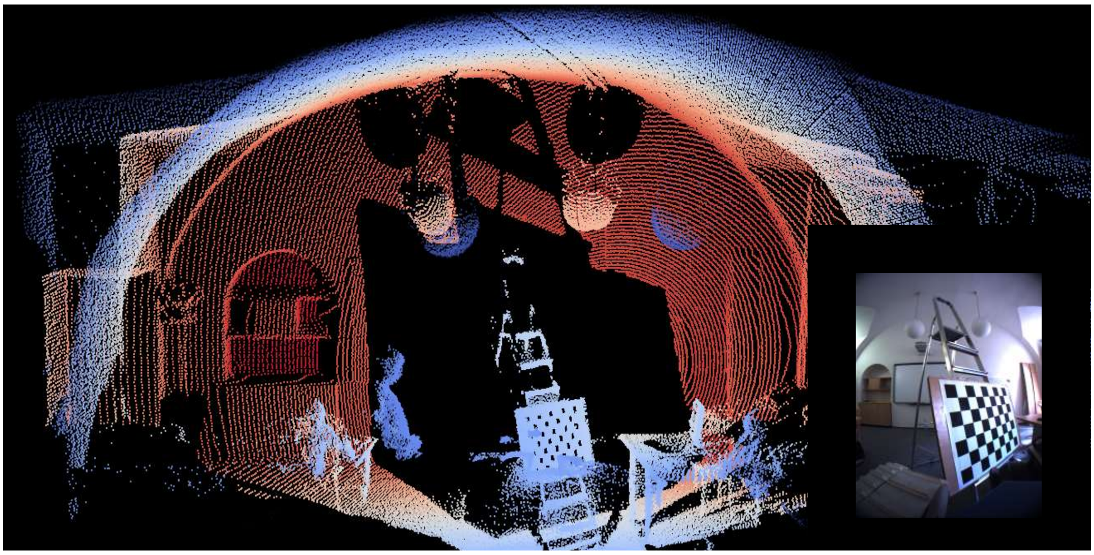


Figure 1.5. A room 3D point cloud scan acquired during the NIFTi robot SICK LMS-151 scanner calibration (not part of this work). Points are colored by the point X coordinate.

1.3. Reference Systems for Ground Truth Measurements

Two different reference systems were used during the experimental part of this work. The first one was the Vicon Bonita IR marker tracking system installed in a dedicated laboratory of the ETH Zürich. It consisted of 12 high-speed high-resolution cameras (Fig. 1.6) capturing the whole laboratory and of small ball markers reflecting IR light (1.7) attached to the body of the robot. This reference system, given that the markers are observed by a sufficient number of cameras, reach sub-millimeter accuracy. The position information was incorporated directly into the robot system and thus, each reference sample is labeled by a time stamp corresponding to all other sensor measurements acquired at that time instant. The rate of the reference is approximately 80 Hz.

The second reference system was the Leica TS15 I theodolite (Fig. 1.8) automatically tracking a prism marker (Fig. 1.9) attached to the robot. The horizontal and vertical angle measurement accuracy is 5" (5/3600 degree), the distance measuring accuracy is 3 mm (in the continuous measuring mode) with range up to 1000m (if using the Mini Prism). The theodolite was connected to a laptop via an RS-232 serial link continuously registering the prism position (at 15Hz rate).

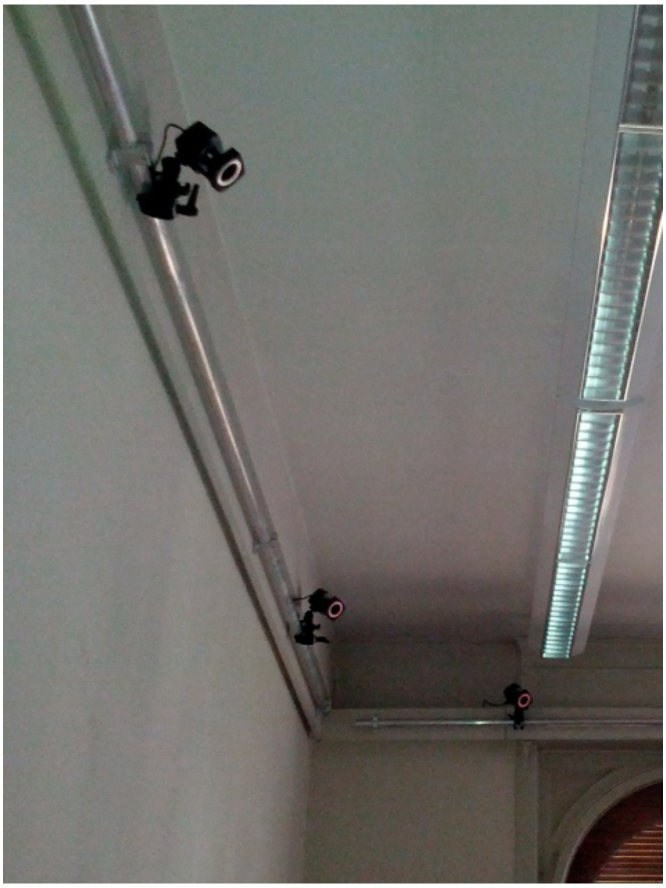


Figure 1.6. The Vicon Bonita (www.vicon.com) cameras installed in the laboratory of the ETH Zürich.

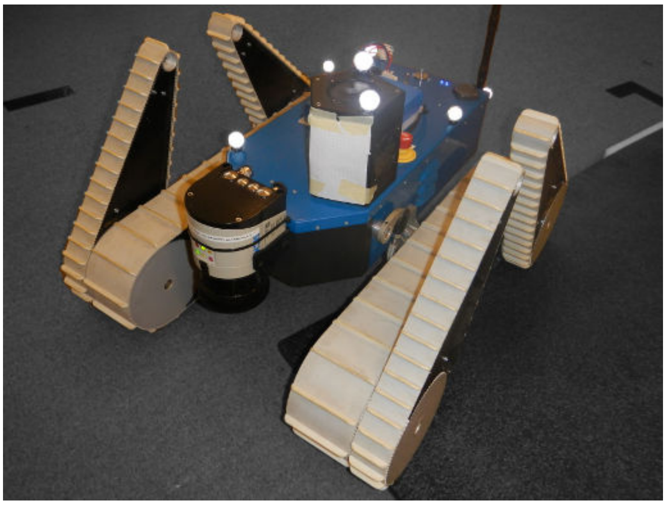


Figure 1.7. The IR light reflecting markers used with the the Vicon reference system. In this picture, the markers have been illuminated by the camera flash.

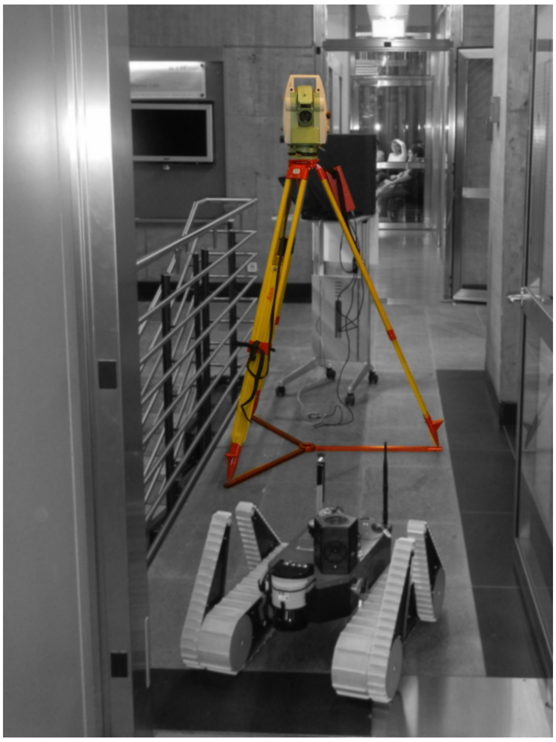


Figure 1.8. The Leica TS15 I theodolite setup during the *hallway* experiment. It is measuring the distance to the prism attached to the robot with frequency of 15Hz. The measurement is sent to the laptop via a RS-232 link. Similarly to the Vicon system measurements, these are published inside the robot system being labeled with the current robot's system time.

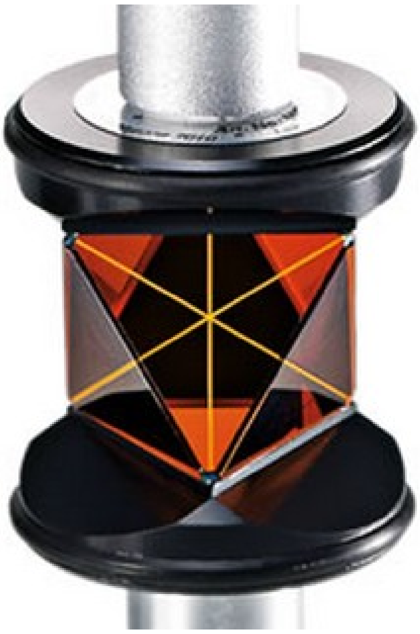


Figure 1.9. The Leica GRZ101 360 Degree Mini Prism (3cm high). It was attached to the robot to benefit from the theodolite's tracking functionality. (figure adopted from `http://www.surveyequipment.com/`

2. System Model

The object being localized - in our case, the NIFTi robot - is modeled as a rigid body with a constant rate of change of angular rates and velocities ($\dot{\omega}, \dot{a} = \text{const.}$). Presence of a constant gravitational acceleration is expected and incorporated into the system model equations. No dissipative forces are considered.

There are four coordinate frames considered:

R(obot)
frame coincides with the defined center of the robot.

I(MU)
frame represents the inertial measurement unit coordinate frame as defined by its manufacturer.

O(dometry)
frame represents the tracked gear frame of the robot.

N(avigation)
frame represents the world frame.

In all the coordinate frames, the North-West-Up axes convention is followed, with the x axis pointing forward (or to the North in the N frame), the y axis pointing to the left (or to the West) and the z axis pointing up. Rotations around each axis follow the right-hand rule (Fig. 2.1).

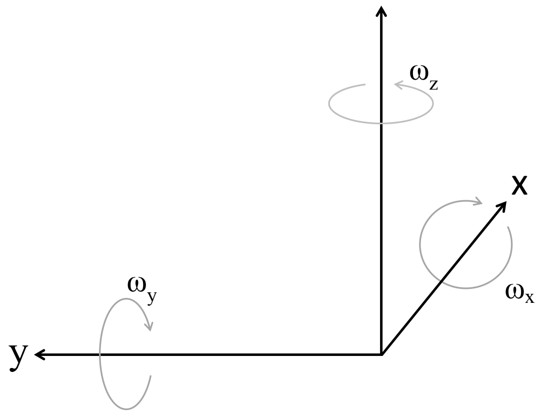


Figure 2.1. Coordinate frame axes convention. In the case of the N frame, the X axis points towards North, Y axis to the West and the Z axis points up.

2.1. Non-Linear System Model

The fundamental part of the system are differential equations describing development of its state in time. The state has been defined as

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_N \\ \mathbf{q}_N^R \\ \mathbf{v}_R \\ \omega_R \\ \mathbf{f}_R \\ \lambda \\ \mathbf{b}_{\omega,I} \\ \mathbf{b}_{f,I} \\ \beta \end{bmatrix} \tag{2.1}$$

where $\mathbf{p}_N$ is the position of the robot in the N frame, $\mathbf{q}_N^R$ is a unit quaternion representing the attitude of the robot, $\mathbf{v}_R$ is the velocity of the robot expressed in the R frame, ω_R is the angular rate, $\mathbf{f}_R$ is the specific force, λ is the scale of the visual odometry, $\mathbf{b}_{\omega,I}$ and $\mathbf{b}_{f,I}$ are the accelerometer and angular rate sensor biases expressed in the I coordinate frame and β is the rotation of the O frame around the y axis.

The state of the robot modeled as a rigid body movement propagates in time according to these equations [2, eq. 110], [12, eq. 5]:

$$\dot{\mathbf{p}}_N = C^T_{(\mathbf{q}_N^R)}\mathbf{v}_R \tag{2.2}$$

$$\dot{\mathbf{q}}_N^R = \frac{1}{2}\Omega(\omega_R)\mathbf{q}_N^R \tag{2.3}$$

$$\dot{\mathbf{v}}_R = \mathbf{f}_R - C_{(\mathbf{q}_N^R)}\mathbf{g}_N + \lfloor \mathbf{v}_R \rfloor \omega_R \tag{2.4}$$

$$\begin{aligned} \dot{\omega}_R &= 0 & \dot{\mathbf{f}}_R &= 0 & \dot{\lambda} &= 0 \\ \dot{\mathbf{b}}_{\omega,I} &= 0 & \dot{\mathbf{b}}_{f,I} &= 0 & \dot{\beta} &= 0 \end{aligned} \tag{2.5}$$

where $\mathbf{g}_N = [0, 0, g]^T$ and $\Omega(\omega_R)$ in (2.3) is a matrix representing a quaternion and vector product operation [2, eq. 108]. It is constructed as

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{2.6}$$

In (2.5), time derivations of the angular rates and the specific forces are equal to zero - usually, they are considered being an input rather than a state. However, we have included them into the state vector and let the Kalman filter update them.

2.2. Linear Error Model

To exploit benefits associated with small values of rotation, we define the error state [4, eq. 3.25]

$$\Delta \mathbf{x} = \begin{bmatrix} \Delta \mathbf{p}_N \\ \delta\theta \\ \Delta \mathbf{v}_R \\ \Delta \omega_R \\ \Delta \mathbf{f}_R \\ \Delta\lambda \\ \Delta \mathbf{b}_{\omega,I} \\ \Delta \mathbf{b}_{f,I} \\ \Delta\beta \end{bmatrix} \tag{2.7}$$

whose components express difference between the true real-world value and our best estimate, e.g.

$$\mathbf{p}_N = \hat{\mathbf{p}}_N + \Delta \mathbf{p}_N \tag{2.8}$$

where the $\hat{\mathbf{p}}_N$ is our best estimate of the true value. There is a complication with the quaternion representing attitude because to combine two rotations, quaternion product is defined instead of a simple sum. To express difference between the true value and our best estimate, the error quaternion is defined [2, eq. 158]

$$\delta \mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \tag{2.9}$$

which is for small errors in attitude approximately

$$\delta \mathbf{q} \approx [\vec{\delta q}^{T}, 1]^T \tag{2.10}$$

where $\vec{\delta q}$ is the vector part of the $\delta \mathbf{q}$. We can express this small rotation in a form of a rotation vector [2, eq. 162] $\delta\theta$:

$$\delta\theta = 2\vec{\delta q} \tag{2.11}$$

Having the error state defined, we can describe the error state propagation in time the same way we did with the system state:

$$\Delta \dot{\mathbf{p}}_N \approx C^T_{(\hat{\mathbf{q}}^R_N)} \Delta \mathbf{v}_R - C^T_{(\hat{\mathbf{q}}^R_N)} \delta\theta \tag{2.12}$$

$$\delta\dot{\theta} \approx -\lfloor \hat{\omega}_R \rfloor \delta\theta + \Delta\omega_R + \mathbf{n}_\theta \tag{2.13}$$

$$\Delta \dot{\mathbf{v}}_R \approx \Delta \mathbf{f}_R - \lfloor C_{(\hat{\mathbf{q}}^R_N)} \mathbf{g}_N \rfloor \delta\theta + \lfloor \hat{\mathbf{v}}_R \rfloor \Delta\omega_R - \lfloor \hat{\omega}_R \rfloor \Delta \mathbf{v}_R + \mathbf{n}_v \tag{2.14}$$

$$\begin{aligned} \Delta\dot{\omega}_R &= \mathbf{n}_\omega & \Delta\dot{\mathbf{f}}_R &= \mathbf{n}_f & \Delta\dot{\lambda} &= \mathbf{n}_\lambda \\ \Delta\dot{\mathbf{b}}_{\omega,I} &= \mathbf{n}_{b,\omega} & \Delta\dot{\mathbf{b}}_{f,I} &= \mathbf{n}_{b,f} & \Delta\dot{\beta} &= n_\beta \end{aligned} \tag{2.15}$$

where the approximations in (2.12-2.14) remind the fact that some terms have been neglected ($\Delta \times \Delta \approx 0, \quad C_{(\delta\mathbf{q})} \approx I$). The process of investigating the error propagation is sometimes called *perturbation analysis*, see [13, pp. 46-47], [14]. Detailed derivation of these three equations can be found in the appendix A. In contrast with the non-linear time propagation equations, the error model equations incorporate noise terms $\mathbf{n}_{(\cdot)}$. These noise terms model disturbances in the system as random variables, which drive changes in the states. They are vital for the resulting Kalman filter tuning, since they affect covariance of the states. Note that the position differential equation does not incorporate any noise term, since there is expected no direct disturbance, that could affect position directly.

A similar approach proposed by [4] treats angular rates and linear accelerations as direct system inputs. We have chosen to treat them as a part of the system state, however, we were unable to define direct inputs, that would *drive* these states. Therefore, the noise terms in the equations (2.13-2.14) express this uncertainty.

2.3. Linear Error Model in Matrix Form

The error model equations can be expressed in a compact matrix form

$$\dot{\tilde{x}} = F_c \Delta \mathbf{x} + G_c \mathbf{n} \tag{2.16}$$

where F_c is a continuous time state transition matrix, G_c is a noise coupling matrix and $\mathbf{n}$ is a noise vector composed of all the $\mathbf{n}_{(\cdot)}$ terms. Putting all the equations together, the F_c matrix looks like

$$F = \left[\begin{array}{c|c|c|c|c|c|c|c|c}
\emptyset_3 & -C^T_{(\hat{q}_N^R)} & C^T_{(\hat{q}_N^R)} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_3 & -\lfloor \hat{\omega}_R \rfloor & \emptyset_3 & I_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_3 & -\lfloor C_{(\hat{q}_N^R)} g_N \rfloor & -\lfloor \hat{\omega}_R \rfloor & \lfloor \hat{v}_R \rfloor & I_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_{1\times3} & \emptyset_{1\times3} & \emptyset_{1\times3} & \emptyset_{1\times3} & \emptyset_{1\times3} & 0 & \emptyset_{1\times3} & \emptyset_{1\times3} & 0 \\ \hline
\emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_{1\times3} & \emptyset_{1\times3} & \emptyset_{1\times3} & \emptyset_{1\times3} & \emptyset_{1\times3} & 0 & \emptyset_{1\times3} & \emptyset_{1\times3} & 0
\end{array}\right] \tag{2.17}$$

and the $G_c\mathbf{n}$ term is

$$G_c\mathbf{n} = \left[\begin{array}{c|c|c|c|c|c|c|c}
\emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_3 & \emptyset_3 & I_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_3 & \emptyset_3 & \emptyset_3 & I_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline
\emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & I_3 & \emptyset_3 & \emptyset_{3\times1} \\ \hline
\emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & I_3 & \emptyset_{3\times1} \\ \hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]
\begin{bmatrix} \mathbf{n}_\theta \\ \mathbf{n}_v \\ \mathbf{n}_\omega \\ \mathbf{n}_f \\ n_\lambda \\ \mathbf{n}_{b,\omega} \\ \mathbf{n}_{b,f} \\ n_\beta \end{bmatrix} \tag{2.18}$$

The noise coupling matrix describes how particular noise terms affect the system state. Each $n_{(\cdot)}$ term is a random variable with Gaussian probability distribution. Properties of these random variables are described by their covariances in the system noise matrix Q_c and since they are expected to be independent, the matrix Q_c is diagonal $Q_c = diag(\sigma^2_{\theta_x}, \sigma^2_{\theta_y}, \sigma^2_{\theta_z}, \sigma^2_{v_x}, \sigma^2_{v_y}, \cdots)$ where σ is standard deviation.

2.4. Linear Error Model Discretization

The previous section describes the error model in the continuous time form, which can't be used for computer processing - the expectation of dt approaching zero can't be satisfied, there will always be some finite Δt, therefore, we have to transform the continuous equations (or rather the continuous matrix equation) to the discrete time domain. To do so, we have decided to use the Van Loan discretization method [15] instead of explicitly expressing the values of the discretized matrices as [5] did.

The Van Loan approach defines matrix M

$$M = \begin{bmatrix} -F_c & GQ_cG^T \\ \emptyset & F_c^T \end{bmatrix} \Delta t \tag{2.19}$$

and claims that

$$N = \mathrm{e}^M = \begin{bmatrix} \ddots & F_d^{-1}Q_d \\ \emptyset & F_d^T \end{bmatrix} \tag{2.20}$$

From N, discretized system matrix F_d can be extracted. The discretized system noise matrix Q_d can be obtained by left multiplying the upper right part of N by F_d.

3. Measurement Model

The previous chapter describes mathematical model of the system. However, such a model is only an approximation of the real robot dynamics, moreover, it is impossible to measure the system state exactly, there is always some uncertainty. Therefore, even if a perfect model of the state was known, it would be impossible to initialize it correctly. From these reasons, it is necessary to provide another source of information, that continuously corrects the model of the system. In our case, this source are the sensors attached to the robot that provide measurements of the current state of the robot.

Generally, the measured value $\mathbf{y}$ can be described as a sum of a function $h(\mathbf{x})$ of the state $\mathbf{x}$ and of some random noise $\mathbf{m}$ caused by physical characteristics of the particular sensor:

$$\mathbf{y} = h(\mathbf{x}) + \mathbf{m} \tag{3.1}$$

Similarly, using the same function h, we can express our prediction of the measured value based on what is currently known about the system:

$$\hat{\mathbf{y}} = h(\hat{\mathbf{x}}) \tag{3.2}$$

There will surely be some difference $\Delta\mathbf{y} = \hat{\mathbf{y}} - \mathbf{y}$ caused by imperfection in the state estimate as well as by the sensor errors. Following the ideas from the previous chapter, this difference can be expressed in terms of the error state $\Delta\mathbf{x}$ (2.7):

$$\begin{aligned} \Delta\mathbf{y} = \mathbf{y} - \hat{\mathbf{y}} &= h(\mathbf{x}) - h(\hat{\mathbf{x}}) + \mathbf{m} \\ &= h(\hat{\mathbf{x}} + \Delta\mathbf{x}) - h(\hat{\mathbf{x}}) + \mathbf{m} \end{aligned} \tag{3.3}$$

In the case of h being linear, (3.3) becomes

$$\Delta\mathbf{y} = h(\Delta\mathbf{x}) + \mathbf{m} \tag{3.4}$$

however, the condition of linearity is not always fulfilled. Nevertheless, we will always be able to approximate the behavior of h in some close proximity to the current state $\hat{\mathbf{x}}$ by a similar function h', which is linear in elements of $\hat{\mathbf{x}}$ such that

$$h(\hat{\mathbf{x}} + \Delta\mathbf{x}) - h(\hat{\mathbf{x}}) \approx h'(\Delta\mathbf{x})|_{\hat{\mathbf{x}}} = H_{\hat{\mathbf{x}}}\Delta\mathbf{x} \tag{3.5}$$

where $H_{\hat{\mathbf{x}}}$ is an innovation matrix projecting observed difference in measurement on the error state elements.

3.1. Inertial Measurement Unit

The inertial measurement unit (IMU) is capable of measuring specific force [3] in all three dimensions as well as angular rates, also along all three axes.

The specific force measurement is a sum of acceleration and gravitational force, but the measurement also contains a bias, i.e. a constant or slowly changing value independent of the actual acting forces, and noise, which is expected to have normal probability distribution with a zero mean. All the values are measured in the IMU sensor coordinate frame I. Therefore,

$$\mathbf{y}_{f,I} = \mathbf{f}_I + \mathbf{b}_{f,I} + \mathbf{m}_{f,I} \tag{3.6}$$

where $\mathbf{y}_{f,I}$ is the measurement, $\mathbf{f}_I$ is the true specific force, $\mathbf{b}_{f,I}$ is the sensor bias and $\mathbf{m}_{f,I}$ is the sensor noise.

Since the interesting value $\mathbf{y}_{f,I}$ is expressed in the I frame, we define a constant rotation matrix C_R^I, which rotates vectors expressed in the R (Robot) frame to the I frame. Translation between the I and R frames does not affect the measured values directly, thus, it is not considered. Since the IMU is placed close to the R frame origin, we neglect the centrifugal force induced by a rotation of the R frame and conditioned by non-zero translation between the R and I frames. Nevertheless, using the rotation matrix, we can express the measurement using the system state:

$$\mathbf{y}_{f,I} = C_R^I \mathbf{f}_R + \mathbf{b}_{f,I} + \mathbf{m}_{f,I} \tag{3.7}$$

where both $\mathbf{f}_R$ and $\mathbf{b}_{f,I}$ are elements of the system state. If we compare the measured value and the expectation of the measurement, we can express the h function which is, in this case, equal to the h'

$$\begin{aligned}\mathbf{y}_{f,I} - \hat{\mathbf{y}}_{f,I} = \Delta\mathbf{y}_{f,I} &= C_R^I \mathbf{f}_R + \mathbf{b}_{f,I} - C_R^I \hat{\mathbf{f}}_R - \hat{\mathbf{b}}_{f,I} + \mathbf{m}_{f,I} \\ &= C_R^I \Delta\mathbf{f}_R + \Delta\mathbf{b}_{f,I} + \mathbf{m}_{f,I}\end{aligned} \tag{3.8}$$

and which can be expressed in the $H_{\hat{\mathbf{x}}}\Delta\mathbf{x}$ form as

$$\Delta\mathbf{y}_{f,I} = \begin{bmatrix}\emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & C_R^I & \emptyset_{3\times1} & \emptyset_3 & I & \emptyset_{3\times1}\end{bmatrix} \Delta\mathbf{x} + \mathbf{m}_{f,I} \tag{3.9}$$

where the error state $\Delta\mathbf{x}$ was defined in (2.7).

The angular rate measurement is treated identically; the output of the sensor is

$$\mathbf{y}_{\omega,I} = \omega_I + \mathbf{b}_{\omega,I} + \mathbf{m}_{\omega,I} \tag{3.10}$$

where ω_I is the angular rate, $\mathbf{b}_{\omega,I}$ is the sensor bias and $\mathbf{m}_{\omega,I}$ is the sensor noise. Comparing to the expected measurement, a measurement residual is obtained:

$$\mathbf{y}_{\omega,I} - \hat{\mathbf{y}}_{\omega,I} = \Delta\mathbf{y}_{\omega,I} = C_R^I \Delta\omega_R + \Delta\mathbf{b}_{\omega,I} + \mathbf{m}_{\omega,I} \tag{3.11}$$

which can be expressed in the matrix form

$$\Delta\mathbf{y}_{\omega,I} = \begin{bmatrix}\emptyset_3 & \emptyset_3 & \emptyset_3 & C_R^I & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & I & \emptyset_{3\times1}\end{bmatrix} \Delta\mathbf{x} + \mathbf{m}_{\omega,I} \tag{3.12}$$

3.2. Caterpillar Tracks Odometry

The *NIFTi* robot (1.2) is equipped with caterpillar tracks and therefore, steering is realized by setting different velocities for each of the tracks (*skid-steering*). The velocities are measured by incremental optical angle sensors with approximate sampling frequency of 15 Hz.

The problem of skid-steered odometry is described, among others, in [7, 16]. The main problem associated with the caterpillar tracks is slippage, which is inevitable when the robot turns. The slippage is affected by many parameters including the type and local properties of the surface robot traverses and it is very difficult to model. The authors of [7, 16] propose compensation for non-linear distribution of slippage between the tracks, which was estimated experimentally.

Based on experience with our robot, we neglect the slippage making this assumption:

$$v_{O,x,simple} = \frac{v_r + v_l}{2} \tag{3.13}$$

$$\dot{\theta}_{O,simple} = \frac{v_r - v_l}{D} \tag{3.14}$$

where $v_{O,x,simple}$ is the robot forward velocity; the velocities in the Y and Z axes are set to zero. The $\dot{\theta}_{O,simple}$ term is the angular rate of the robot about its Z axis, v_l and v_r are track velocities measured by incremental optical sensors and D is the distance between the tracks (Fig. 3.1). Although (3.14) provides angular rate measurement, preliminary experiments have shown that its inaccuracy makes it unusable. From that reason, we utilize only (3.13) for the fusion.

To choose between the advanced odometry proposed by [7, 16] and our simple solution, we have performed several experiments, results of one of them are shown in Fig. 3.2. We observed no significant performance improvement using the slippage-compensating odometry algorithm and from that reason, we choose to continue using our simple solution expecting that the localization errors observable in the Fig. 3.2 would be dealt with by means of the fusion system we propose.

Even simplest experiments showed a necessity of a proper alignment of R and O frames. Since the robot position is obtained by integration of velocity expressed in the R frame, we define a rotation matrix C_R^O:

$$\mathbf{v}_O = C_R^O \mathbf{v}_R \tag{3.15}$$

which expresses the $\mathbf{v}_R$ in the O frame. In the case of the NIFTi robot, we observed a minor misalignment of these two frames. It consisted of a rotation about the Y axis by one degree. Although small, this rotation caused the estimated position to rise in the Z direction constantly while the robot moved forward. We propose two approaches to handle the C_R^O. The first one handles it as a constant value and expects a calibration to be performed. The second one handles it as a part of the system state and lets the EKF to estimate it on-the-run. These two approaches are compared in the end of this section.

3.2.1. Constant Transformation between the Odometry and the Robot Frames

The *Constant Transformation* approach leads to **simpler equations** but requires **calibration** of the C_R^O transformation. The measurement equations yield:

$$\mathbf{y}_{v,O} = C_R^O \mathbf{v}_R + \mathbf{m}_{v,O} \tag{3.16}$$

where $\mathbf{y}_{v,O}$ is the linear velocity measured by the track odometry, expressed in the O frame. Because of the linearity of this relation, the measurement innovation is

$$\begin{aligned} \mathbf{y}_{v,O} - \hat{\mathbf{y}}_{v,O} = \Delta \mathbf{y}_{v,O} = & \\ &= C_R^O \mathbf{v}_R - C_R^O \hat{\mathbf{v}}_R + \mathbf{m}_{v,O} \\ &= C_R^O \Delta \mathbf{v}_R + \mathbf{m}_{v,O} \end{aligned} \tag{3.17}$$

and expressed in the matrix form

$$\Delta \mathbf{y}_{v,O} = \begin{bmatrix} \emptyset_3 & \emptyset_3 & C_R^O & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \end{bmatrix} \Delta \mathbf{x} + \mathbf{m}_{v,O} \tag{3.18}$$

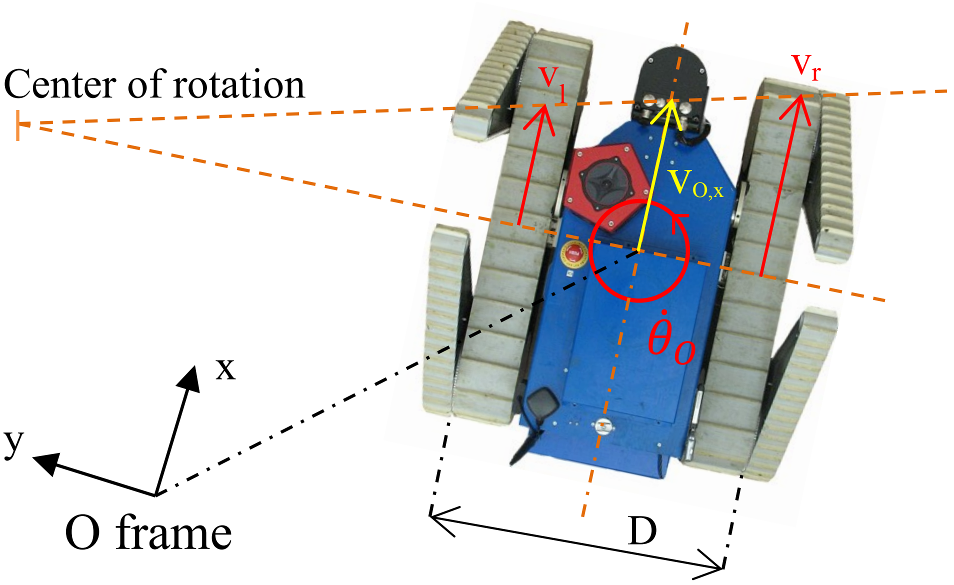


Figure 3.1. Geometry of the (simple) caterpillar track odometry. v_l and v_r stand for measured track velocities, $v_{O,x}$ and $\dot{\theta}_O$ correspond to (3.13) and (3.14). The O frame is depicted outside the robot body for better clarity, its origin coincides with the tracked gear of the robot as indicated by the black dashed line. This figure was adopted from [8]

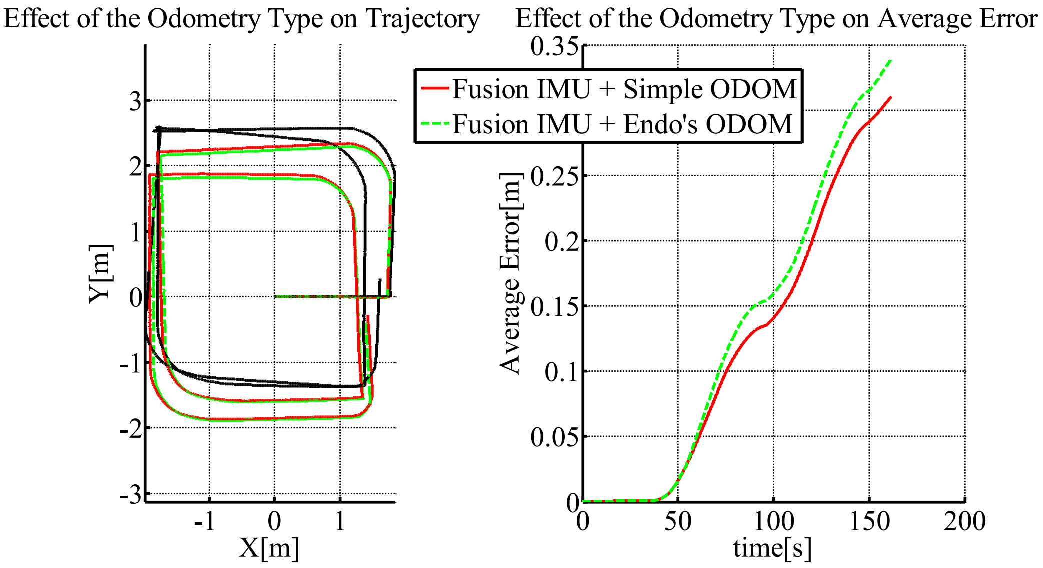


Figure 3.2. Comparing the odometry proposed by [7, 16] with the simple one proposed in this work. To generate this output, our fusion algorithm was used instead of direct velocity integration. (The *average error* is defined in the section 5)

3.2.2. Estimated Transformation between the Odometry and the Robot Frames

The *Estimated Transformation* approach handles the transformation between the two frames as a **part of the system state** and given a measurement (velocity measurement originating from a laser scan, visual odometry or a similar exterioceptive sensor) that makes this transformation observable, it is **estimated by the EKF** on-the-run with no need for calibration.

The transformation is limited to only one rotation about the Y axis and thus, only one variable representing angle is sufficient (the β angle and the $\Delta\beta$ error angle). This simplification is determined by the robot construction, where the track gear is attached to the robot body by joints with one degree of freedom - the rotation about the axis Y. This connection shows a minor backlash corresponding to the one-degree misalignment.

In this approach, the rotation matrix C_R^O yields

$$C_R^O(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{3.19}$$

where the β is the the element of the system state. Then, the corresponding innovation yields

$$\begin{aligned} \Delta\mathbf{y}_{v,O} &= \mathbf{y}_{v,O} - \hat{\mathbf{y}}_{v,O} + \mathbf{m}_{v,O} \\ &= C_R^O(\beta)\mathbf{v}_R - C_{R'}^O(\hat{\beta})\hat{\mathbf{v}}_R + \mathbf{m}_{v,O} \\ &= C_R^{R'}(\Delta\beta)C_{R'}^O(\hat{\beta})(\hat{\mathbf{v}}_R + \Delta\mathbf{v}_R) - C_{R'}^O(\hat{\beta})\hat{\mathbf{v}}_R + \mathbf{m}_{v,O} \\ &= (C_R^{R'}(\Delta\beta) - I)C_{R'}^O(\hat{\beta})\hat{\mathbf{v}}_R + C_R^{R'}(\Delta\beta)C_{R'}^O(\hat{\beta})\Delta\mathbf{v}_R + \mathbf{m}_{v,O} \end{aligned} \tag{3.20}$$

$$\begin{aligned} \Delta\mathbf{y}_{v,O} &= \\ &= \begin{bmatrix} \cos\Delta\beta - 1 & 0 & \sin\Delta\beta \\ 0 & 0 & 0 \\ -\sin\Delta\beta & 0 & \cos\Delta\beta - 1 \end{bmatrix} C_{R'}^O(\hat{\beta})\hat{\mathbf{v}}_R \\ &\quad + \begin{bmatrix} \cos\Delta\beta & 0 & \sin\Delta\beta \\ 0 & 1 & 0 \\ -\sin\Delta\beta & 0 & \cos\Delta\beta \end{bmatrix} C_{R'}^O(\hat{\beta})\Delta\mathbf{v}_R + \mathbf{m}_{v,O} \end{aligned} \tag{3.21}$$

where R' is a hypothetical coordinate frame constructed by rotating the O frame by the estimated rotation. This form, however, is not suitable for the H matrix form. We have to derive partial derivatives along the $\Delta\mathbf{x}$ state at the system state estimate $\hat{\mathbf{x}}$. The only non-zero partial derivatives are the ones along the $\Delta\beta$ and $\Delta\mathbf{v_R}$

$$\begin{aligned} \left.\frac{\partial\Delta\mathbf{y}_{v,O}}{\partial\Delta\beta}\right|_{\substack{\Delta\beta=0 \\ \Delta\mathbf{v_R}=[0,0,0]^T}} &= \begin{bmatrix} -\sin\Delta\beta & 0 & \cos\Delta\beta \\ 0 & 0 & 0 \\ -\cos\Delta\beta & 0 & -\sin\Delta\beta \end{bmatrix} C_{R'}^O(\hat{\beta})\hat{\mathbf{v}}_R \\ &\quad + \left.\begin{bmatrix} -\sin\Delta\beta & 0 & \cos\Delta\beta \\ 0 & 0 & 0 \\ -\cos\Delta\beta & 0 & -\sin\Delta\beta \end{bmatrix} C_{R'}^O(\hat{\beta})\Delta\mathbf{v}_R\right|_{\substack{\Delta\beta=0 \\ \Delta\mathbf{v_R}=[0,0,0]^T}} \\ &= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} C_{R'}^O(\hat{\beta})\hat{\mathbf{v}}_R \end{aligned} \tag{3.22}$$

similarly, the second partial derivative yields

$$\begin{aligned} \left.\frac{\partial\Delta\mathbf{y}_{v,O}}{\partial\Delta\mathbf{v}_R}\right|_{\substack{\Delta\beta=0 \\ \Delta\mathbf{v_R}=[0,0,0]^T}} &= \left.\begin{bmatrix} \cos\Delta\beta & 0 & \sin\Delta\beta \\ 0 & 1 & 0 \\ -\sin\Delta\beta & 0 & \cos\Delta\beta \end{bmatrix} C_{R'}^O(\hat{\beta})\right|_{\substack{\Delta\beta=0 \\ \Delta\mathbf{v_R}=[0,0,0]^T}} \\ &= C_{R'}^O(\hat{\beta}) \end{aligned} \tag{3.23}$$

Finally, the linearized matrix form is

$$\Delta \mathbf{y}_{v,O} = \begin{bmatrix} \emptyset_3 & \emptyset_3 & C^{O}_{R'}(\hat{\beta}) & \emptyset_3 & \emptyset_3 & \emptyset_{3\times 1} & \emptyset_3 & \emptyset_3 & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} C^{O}_{R'}(\hat{\beta})\hat{\mathbf{v}}_R \end{bmatrix} \Delta \mathbf{x} + \mathbf{m}_{v,O} \tag{3.24}$$

3.2.3. Comparison of the Constant and the Estimated C^O_R Transformation Approaches

To choose between the two approaches in determining the C^R_O transformation, an experiment was performed (Fig. 3.3). It consisted of robot passing through a straight hallway 100m long. Under these conditions, we expected the β angle estimation to work correctly resulting in the same value as the one obtained by calibration.

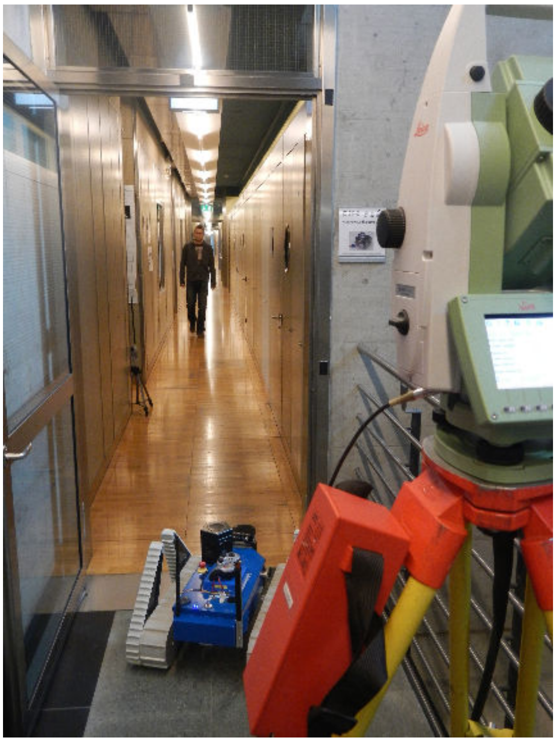


Figure 3.3. The *Hallway* experiment setup in the Autonomous System Lab, ETH Zürich. This experiment was used to verify or to negate the β angle on-the-run estimation, using the *Leica TS15 I*(1.3) theodolite as a source of reference. The conditions were ideal: 100m long hallway, perfectly leveled.

However, the value did not converge to the correct value and therefore, the position estimate kept drifting (Fig. 3.4 and 3.5). Similar results were obtained for other experiments. We conclude, that the condition of observability of the β angle was not met. Since the measurement that was supposed to make it observable was velocity provided by ICP (3.3), we analyzed its output in detail and discovered, that the direction of velocity expressed in the R frame is not accurate (the magnitude of the error is approximately one degree) and moreover, the error was composed of a noise (that was expected and the fusion algorithm would compensate it) and of a bias. This bias made the β angle unobservable since the ICP velocity measurement is the only one that provides this information (Visual Odometry position or velocity are not used for reasons explained in the section 3.4).

The conclusion is that the **constant transformation** C^R_O obtained by calibration is used, leaving the estimation approach for further work, since it (potentially) offers a more robust solution not demanding a user of the system to obtain the parameter manually.

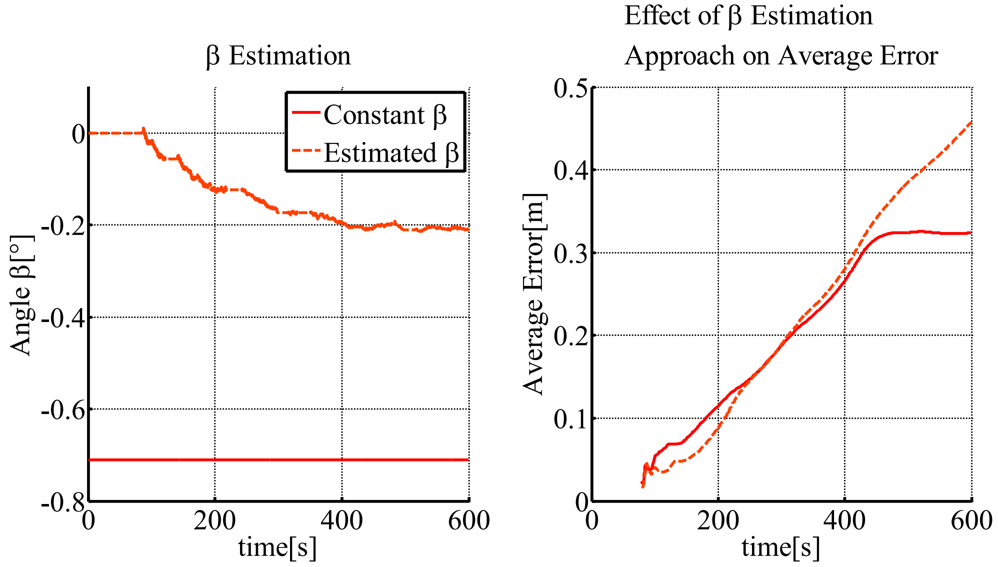


Figure 3.4. Example of the β angle estimation during an indoor experiment *hallway* (see Appendix C). It is apparent in the left part of this plot that the estimate of β converges to an inaccurate value (compare it with the constant, calibrated value). The right part of the plot shows effect of this inaccuracy on the average error(5) against the Leica reference.

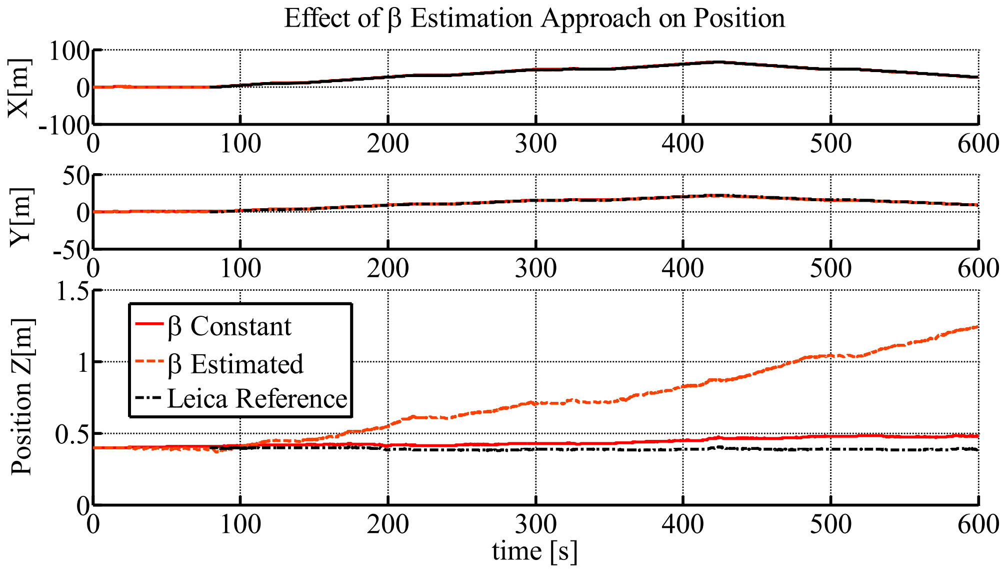


Figure 3.5. Example of the β angle estimation effect on position (mainly the Z (Up) axis). Although the β estimate error was relatively small (0.5°) , the drift in the position is clearly observable.

3.3. Iterative Closest Point Laser Scan Matching Odometry (ICP)

The ICP algorithm is used to estimate translation and rotation between every new incoming laser scan of the robot surroundings and a laser point cloud map created until that time from previously registered laser scans. This approach goes through a rapid development these days, for a state-of-the-art study, see [11]. An example of a real-world object represented by a point cloud is depicted in Fig. 3.6.

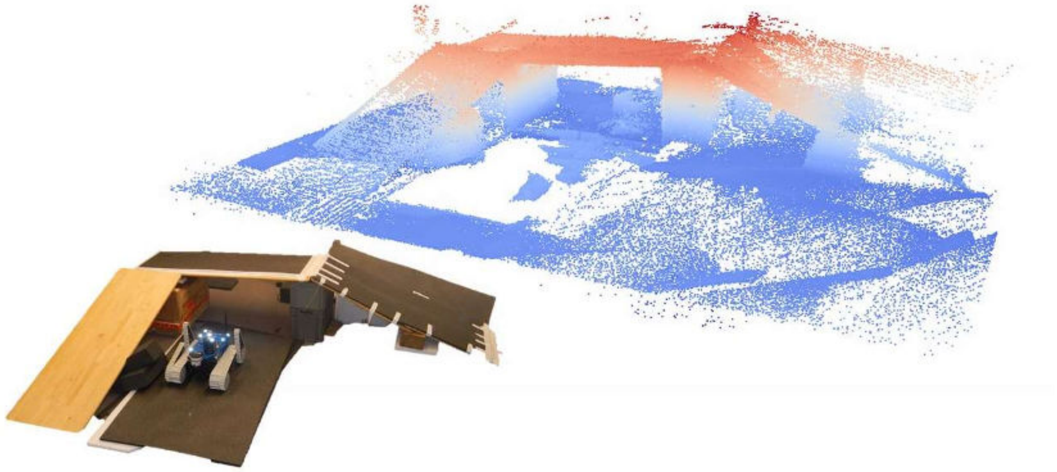


Figure 3.6. Testing environment scanned by laser, the resulting point cloud colored by elevation.

In this work, three approaches using these position measurements were proposed and tested. All of them aim to use the measurement as a velocity rather than absolute position from reasons mentioned in the section 1.2. The first proposed approach, which is called *incremental position approach*, treats an ICP localization measurement as a movement expressed in the N coordinate frame relative to the previous ICP measurement. Its purpose is to overcome the extremely low scanning frequency and the nonlinear behavior of the system during a single 3D scan. However, it is impossible to correctly discretize the system equations respecting the system non-linear dynamics and the scanning frequency at the same time and thus, corrections, that propagate to the system state from this measurement tend to be inaccurate as well. The second approach treats the ICP measurement as a velocity in the R frame (*velocity approach*) and is expected to perform better for sensor biases estimation, which are not observable directly. This approach expects the robot to move linearly between two ICP scans (allowing a simultaneous arbitrary rotation), which is a strong assumption and also a major drawback of this approach. The last approach - the *trajectory approach* tries to overcome the assumption of a linear movement laid by the previous approach by a sub-optimal usage of the estimated state to approximate a probable behavior of the system between two consecutive ICP scans. The *Trajectory Approach* was chosen as the final solution for reasons described in (3.3.4).

3.3.1. Incremental Position Approach ICP Aiding

The incremental position aiding provides an increment in position and attitude relative to the previous measurement step, while the position increment is considered in the R frame and thus behaving as a velocity measurement independent of the current N frame relationship to the ICP world frame (these two are expected to drift away since the ICP is not drift-free while registering new parts of the point cloud map).

To express an increment in ICP position between ICP discreet times $i-1$ and i expressed in the R frame at the ICP discrete time $i-1$ (Fig. 3.7), we remember the position and attitude provided by the ICP in the previous ICP measurement time $i-1$ and use them to evaluate the transformation:

$$\Delta\mathbf{p}_{R,ICP,i} = C_{(\mathbf{q}^{R}_{N,ICP,i-1})}(\mathbf{p}_{N,ICP,i} - \mathbf{p}_{N,ICP,i-1}) \tag{3.25}$$

Then, this increment is expressed in the N frame equivalent to the ICP world frame at the system discrete time $k' \equiv (i-1)$:

$$\Delta\mathbf{p}_{N,ICP,i} = C^{T}_{(\mathbf{q}^{R}_{N,k'})}\Delta\mathbf{p}_{R,ICP,i} \tag{3.26}$$

The position increment measurement evaluated this way suppresses the effect of the N and ICP world frames drift effect by ignoring it and thus behaving as a velocity measurement in the R frame. Comparing this measurement with the predicted one at the current system time $k \equiv i$:

$$\Delta\mathbf{y}_{\Delta p,ICP} = \Delta\mathbf{p}_{N,ICP,i} - \widehat{\Delta\mathbf{p}}_{N,k} + \mathbf{m}_{\Delta p,ICP} \tag{3.27}$$

$$\Delta\mathbf{y}_{\Delta p,ICP} = \Delta\mathbf{p}_{N,ICP,i} - (\hat{\mathbf{p}}_{N,k} - \hat{\mathbf{p}}_{N,k'}) + \mathbf{m}_{\Delta p,ICP} \tag{3.28}$$

provides us a measurement residual, whose projection to the error state is

$$\Delta\mathbf{y}_{\Delta p,ICP} = \begin{bmatrix} I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \end{bmatrix} \Delta\mathbf{x} + \mathbf{m}_{\Delta p,ICP} \tag{3.29}$$

Similarly, we process the attitude measurement relatively to the previous one to suppress the ICP world frame drift. To do so, we use the previous measurement to extract rotation of the body frame between these measurements:

$$\mathbf{q}^{R}_{N,ICP,i} = \mathbf{q}^{R}_{R',ICP} \otimes \mathbf{q}^{R'}_{N,ICP,i-1} \tag{3.30}$$

$$\mathbf{q}^{R}_{R',ICP} = \mathbf{q}^{R}_{N,ICP,i} \otimes \left(\mathbf{q}^{R'}_{N,ICP,i-1}\right)^{-1} \tag{3.31}$$

where $\mathbf{q}^{R}_{R',ICP}$ is the rotation that happened between the two consequent ICP measurements. To provide it as a measurement, we apply this rotation to a stored estimated attitude at the system time $k' \equiv i-1$:

$$\mathbf{y}_{q,ICP} = \mathbf{q}^{R}_{R',ICP} \otimes \mathbf{q}^{R}_{N,k'} \tag{3.32}$$

To express the measurement residual, we follow the error quaternion definition (2.9), the error quaternion yields

$$\delta\mathbf{q}_{ICP,i} = \hat{\mathbf{q}}^{R}_{N,k} \otimes (\mathbf{y}_{q,ICP})^{-1} \tag{3.33}$$

where $\hat{\mathbf{q}}^{R}_{N,k}$ is the attitude estimate at the system time $k \equiv i$. We express this residual rotation by means of a rotation vector $\delta\theta_{ICP,i}$

$$\delta\theta_{ICP,i} = 2\delta\vec{\mathbf{q}}_{ICP,i} \tag{3.34}$$

which can be projected onto the error state as

$$\Delta\mathbf{y}_{\delta\theta,ICP} = \begin{bmatrix} \emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} \end{bmatrix} \Delta\mathbf{x} + \mathbf{m}_{\delta\theta,ICP} \tag{3.35}$$

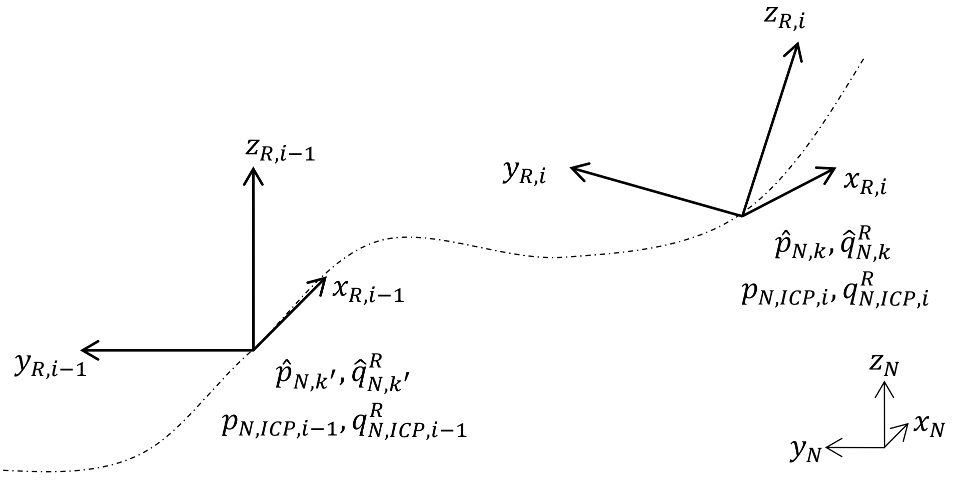

$z_{R,i}$
$y_{R,i}$
$x_{R,i}$
$\hat{p}_{N,k}, \hat{q}^R_{N,k}$
$p_{N,ICP,i}, q^R_{N,ICP,i}$
$z_{R,i-1}$
$x_{R,i-1}$
$y_{R,i-1}$
$\hat{p}_{N,k'}, \hat{q}^R_{N,k'}$
$p_{N,ICP,i-1}, q^R_{N,ICP,i-1}$
z_N
y_N
x_N

Figure 3.7. Two consecutive ICP localization measurements (at discrete times $i-1$ and i) in the N frame. For incremental approach, knowledge of the previously estimated position and attitude in time $i-1$ is also necessary.

3.3.2. Velocity Approach ICP Aiding

The second approach to applying ICP measurements linearly interpolates the robot position and attitude between these measurements. It is assumed that the robot moves directly from the previous ICP measurement position to the current one with respect to the N frame, rotating from the previous attitude to the current one by a single, constant angular rate rotation. The measurement provided to the EKF fusion algorithm is then the constant angular rate $\mathbf{y}_{\omega_R,ICP}$ and a velocity $\mathbf{y}_{v_R,ICP}$ expressed in the R frame as well. These values can be generated with an arbitrary frequency, but only after the latest ICP measurement arrives. This lays a requirement on the EKF algorithm, which has to be able to rewind back into history and recompute the all the state estimates since the previous ICP measurement. However, it is a common technique adopted by many localization and mapping algorithms.

The essential part of this algorithm is to evaluate the correct rotation between the two consecutive ICP measurements. For that, we utilize the native quaternion representation, however, before evaluating the $\mathbf{q}^R_{R',ICP}$ rotation quaternion, we need to check for the rotation to the rotation quaternion mapping issue. Every rotation (up to the 2π period) can be mapped onto two unit quaternions, $\mathbf{q}$ and $-\mathbf{q}$. To obtain a correct $\mathbf{q}^R_{R',ICP}$ rotation quaternion, we have to choose between these two that one, which is closer to the other quaternion representing the second ICP attitude measurement. To so that, we fix the first ICP quaternion $\mathbf{q}^{R'}_{N,ICP,i-1}$ and choose between the $\mathbf{q}^R_{N,ICP,i}$ and $-\mathbf{q}^R_{N,ICP,i}$ following this rule:

$$\mathbf{q}^R_{N,ICP,i} := \begin{cases} -\mathbf{q}^R_{N,ICP,i} & \text{if} \qquad ||\mathbf{q}^{R'}_{N,ICP,i-1} - \mathbf{q}^R_{N,ICP,i}|| > ||\mathbf{q}^{R'}_{N,ICP,i-1} + \mathbf{q}^R_{N,ICP,i}|| \\ \mathbf{q}^R_{N,ICP,i} & \text{otherwise} \end{cases} \tag{3.36}$$

then, we get the $\mathbf{q}^R_{R',ICP}$ using equations (3.30) and (3.31). Note that we did not apply the (3.36) rule in the previous *Incremental Position Approach* since we were not interested in the angular velocities causing the rotations. The angular velocity can be extracted from the $\mathbf{q}^R_{R',ICP}$ (see [2, eq. 3,4]):

$$\theta^R_{R',ICP} = 2\arcsin||\vec{\mathbf{q}}^R_{R',ICP}||\frac{\vec{\mathbf{q}}^R_{R',ICP}}{||\vec{\mathbf{q}}^R_{R',ICP}||} \tag{3.37}$$

$$\omega_{R,ICP} = \frac{1}{t(i)-t(i-1)}\theta^R_{R',ICP} \tag{3.38}$$

where $\theta^R_{R',ICP}$ is a rotation vector equivalent to the quaternion $\mathbf{q}^R_{R',ICP}$ and $t(i)$ and $t(i-1)$ are times corresponding to the the discrete times i and $i-1$ respectively. This angular rate can be used directly as a measurement, the corresponding measurement residual projection on the error state vector yields

$$\Delta\mathbf{y}_{\omega,ICP} = \begin{bmatrix}\emptyset_3 & \emptyset_3 & \emptyset_3 & I_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1}\end{bmatrix}\Delta\mathbf{x} + \mathbf{m}_{\omega,ICP} \tag{3.39}$$

To express an ICP velocity in the R frame, we first evaluate the velocity in the N frame

$$\mathbf{v}_{N,ICP} = \frac{\mathbf{p}_{N,ICP,i} - \mathbf{p}_{N,ICP,i-1}}{t(i)-t(i-1)} \tag{3.40}$$

Transformation from the N frame to the R frame is composed of two rotations: $\mathbf{q}^{R'}_{N,ICP,i-1}$ and $\mathbf{q}^{R''}_{R',ICP}(t)$ for arbitrary time t between the two consequent ICP measurements in the discrete times $i-1$ and i. To express the second quaternion representing the partial rotation between frames R' and R", we follow the equations [2, eq. 3,4] and use the result from (3.38)

$$\mathbf{q}^{R''}_{R',ICP}(t) = \begin{bmatrix}\sin\left(\frac{||\omega_{R,ICP}||(t-t(k'))}{2}\right)\frac{\omega_{R,ICP}}{||\omega_{R,ICP}||} \\ \cos\left(\frac{||\omega_{R,ICP}||(t-t(k'))}{2}\right)\end{bmatrix} \tag{3.41}$$

where $t(k')$ is the time of the previous ICP measurement corresponding to the ICP discrete time $i-1$. Then, the velocity expressed in the R frame is simply obtained by a transformation

$$\mathbf{v}_{R,ICP}(t) = C_{(\mathbf{q}^{R''}_{R',ICP}(t)\otimes\mathbf{q}^{R'}_{N,ICP,i-1})}\mathbf{v}_{N,ICP} \tag{3.42}$$

which can be used directly as a measurement, whose projection onto the error state vector yields

$$\Delta\mathbf{y}_{v,ICP} = \begin{bmatrix}\emptyset_3 & \emptyset_3 & I_3 & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1} & \emptyset_3 & \emptyset_3 & \emptyset_{3\times1}\end{bmatrix}\Delta\mathbf{x} + \mathbf{m}_{v,ICP} \tag{3.43}$$

3.3.3. Trajectory Approach ICP Aiding

The *Trajectory Approach* improves the previous one by adding some knowledge about the trajectory shape; the previous *Velocity Approach* assumed straight path between every two ICP measurements. Experiments have shown that such an assumption may show bad performance during more complex movements - we call it *corner cutting.* However, since the knowledge of the trajectory shape is based on the fusion algorithm estimated one, this solution leads to a suboptimal estimation. This issue is discussed in the conclusion of this work.

The aim of this approach is to replace constant velocity in the N frame by a more accurate guess. The idea is described by the Fig. 3.8. We assume, that the first estimate of the trajectory (the output of the fusion algorithm between two consequent

ICP measurements) is locally very similar to the true trajectory. Thus, it is remembered to be used as a guess around the first ICP measured pose. Then, it is duplicated and aligned with the second ICP measured pose. The resulting trajectory is a weighted average of the original and duplicated trajectory estimates, where the weights are linear functions of time equal to 1 at the time of the associated ICP measurement and equal to 0 at the time of the other ICP measurement.

The resulting trajectory is used as a source for the velocity evaluation in the N frame by a simple differentiation of the position and division by time between samples. Then, the we express the velocity in the R frame using an attitude estimated with the original trajectory estimate. The velocity expressed in the R frame can be used as a measurement, but again, the values for the time period between the two consequent ICP measurements are known *after* the second ICP measurement arrives and thus, the algorithm has to be able to rewind itself to the time of the first ICP measurement and recompute the the time period including the new velocity measurements. The average trajectory is constructed as follows: After the second ICP measurement arrives, we first align the ICP world frame with our N frame by moving them both to make the coordinates associated with the first ICP measurement (at ICP discrete time $i-1$, or system discrete time k') equal to zero. Also, the ICP world frame is rotated to equalize the attitude at the first ICP measurement and the estimated one:

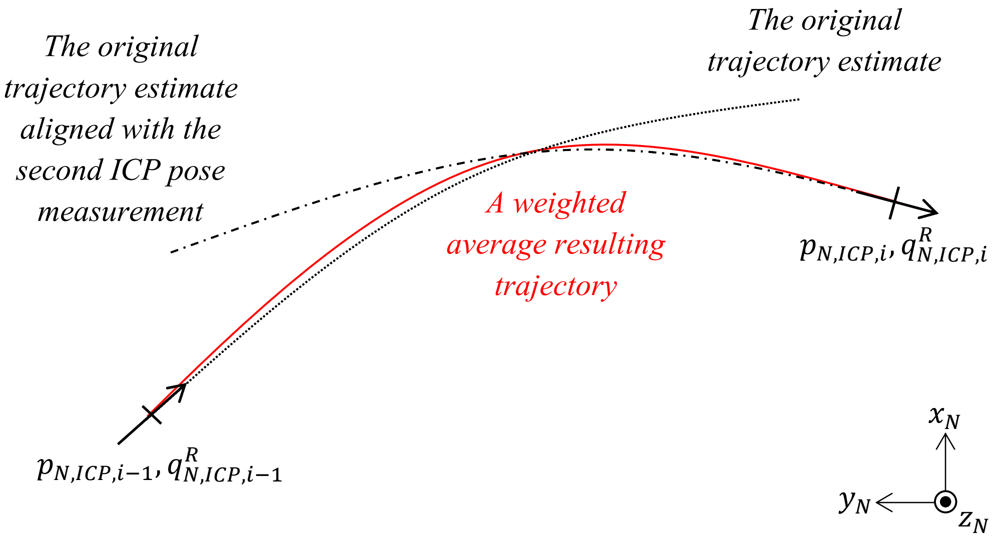

The original trajectory estimate aligned with the second ICP pose measurement
The original trajectory estimate
A weighted average resulting trajectory
$p_{N,ICP,i}, q^R_{N,ICP,i}$
$p_{N,ICP,i-1}, q^R_{N,ICP,i-1}$
x_N
y_N
z_N

Figure 3.8. A sketch demonstrating the trajectory approach idea. A trajectory estimated from other measurements before the ICP measurement arrival (dotted line) is duplicated (dashed line) and aligned with the incoming ICP measurement and a weighted average (red solid line) of these two trajectories is evaluated.

$$
\begin{aligned}
\mathbf{p}'_{N,ICP,i-1} &:= [0,0,0]^T \\
\mathbf{q}'^{R'}_{N,ICP,i-1} &:= \hat{\mathbf{q}}^R_{N,k'} \\
\mathbf{p}'_{N,ICP,i} &= C_{(\hat{\mathbf{q}}^R_{N,k'} \otimes (\mathbf{q}^{R'}_{N,ICP,i-1})^{-1})}(\mathbf{p}_{N,ICP,i} - \mathbf{p}_{N,ICP,i-1}) \\
\mathbf{q}'^{R}_{N,ICP,i} &:= \mathbf{q}^R_{N,ICP,i} \otimes \left(\mathbf{q}^{R'}_{N,ICP,i-1}\right)^{-1} \otimes \hat{\mathbf{q}}^R_{N,k'}
\end{aligned}
\tag{3.44}
$$

where the non-dashed positions and attitudes are the original ones provided by the ICP algorithm and the dashed ones are the translated and rotated ones. This initial transformation is done to suppress the inevitable drift of the distinct *world* or N frames (compare this to the *Incremental Position Approach*). To align the (*a priori*) trajectory estimate ($\hat{\mathbf{p}}_{N,l}, \quad \hat{\mathbf{q}}^R_{N,l}, \quad k' < l < k$), it is shifted until its first point lays in the origin (there is no need for rotation because of the previous step):

$$\hat{\mathbf{p}}'_{N,l} = \hat{\mathbf{p}}_{N,l} - \hat{\mathbf{p}}_{N,k'} \tag{3.45}$$

where l is an index lying between $k' \equiv i-1$ and $k \equiv i$. Then, the same *a priori* trajectory estimate is duplicated and aligned with the second ICP measured pose by shifting it so the last point of the trajectory lies in the origin, rotating it to the ICP attitude $\mathbf{q}^R_{N,ICP,i}$ and finally shifting it so the last point of the trajectory coincides with the ICP one:

$$\hat{\mathbf{p}}''_{N,l} = C_{(\mathbf{q}^R_{N,ICP,i} \otimes (\hat{\mathbf{q}}^R_{N,k})^{-1})} (\hat{\mathbf{p}}_{N,l} - \hat{\mathbf{p}}_{N,k}) + \mathbf{p}'_{N,ICP,i} \tag{3.46}$$

where $\mathbf{p}'_{N,ICP,i}$ was obtained in the equation (3.44). After these transformations, the trajectories are combined by weighted averaging. The weight vectors w'_l and w''_l have the same lengths as the trajectory vectors $\hat{\mathbf{p}}'_{N,l}$ and $\hat{\mathbf{p}}''_{N,l}$ respectively and their values are linearly spaced from 1 to 0 and from 0 to 1 respectively. Each point of the final weighted trajectory is obtained as

$$\hat{\mathbf{p}}'''_{N,l} = \hat{\mathbf{p}}'_{N,l} w'_l + \hat{\mathbf{p}}''_{N,l} w''_l \tag{3.47}$$

where the index l is $k' < l < k$. There is also the vector of attitudes estimated *a priori*, but we do not make any weighting since the main purpose of this approach is to enhance position estimates. Also, it is not completely clear, how this weighting should be done. Thus, when we need the attitudes for expressing velocity in the R frame, we use the original *a priori* ones.

From the trajectory $\hat{\mathbf{p}}'''_{N,l}$, it is simple to compute velocities in the R frame, we differentiate the positions, divide the results by appropriate time increments and get velocities $\hat{\mathbf{v}}'''_{N,l}$. Note that by differentiating, we obtain only $k - k' - 1$ velocity samples. From practical reasons, we duplicate the first one to obtain a same number of the output samples as the input samples, we assume that at frequencies close to 90Hz, this has a negligible effect on the algorithm accuracy. Similarly to (3.42), we express the velocity in the R frame

$$\hat{\mathbf{v}}'''_{R,l} = C_{(\hat{\mathbf{q}}^R_{N,l})} \mathbf{v}'''_{N,l} \tag{3.48}$$

and it may be used directly as a measurement

$$\mathbf{y}_{v,ICP,l} := \hat{\mathbf{v}}'''_{R,l} \tag{3.49}$$

Then, the measurement residual projection onto the error state is identical to (3.43).

Concerning the attitude aiding, the same technique is used as the one used in the *Incremental Position Approach*. It is for further discussion whether there might be some similar way for enhancing the attitude estimation. However, the results indicate that the current solution is accurate enough.

Practical experiments have shown a need for one more improvement in the *Trajectory Approach ICP Aiding* measurements preprocessing. Since the output of the *Trajectory Approach* is velocity rather than position, we can apply some nonholomic constraints making assumptions about the robot movement. The reason to do that is the fact that although the ICP is very accurate in measuring translation between consequent measurements, the attitude measurement is not so accurate or more precisely, noise in the pitch angle causes wrong estimates of the velocity expressed in the R frame, resulting in a problem we call *a climbing robot* (the very same problem addressed in 3.2.3). Although we would expect a random walk behavior, the system tended to slowly climb in the Z or Up axis. We have observed that while the robot moved forward on a leveled plane, the pitch angle indicated by our algorithm was slightly above zero (cca. 0.6 degree), but the direction of the movement measured by the ICP tended to be zero in average (that is the bias mentioned in 3.2.3). This small inaccuracy caused the problem.

To correct it, we apply a simple constraint on the velocity expressed in the R frame: it can happen only in a plane defined by X and Y axes of the O frame. Velocity vectors, that do not comply this constraint (i.e. practically all of them) are rotated to the XY O plane preserving their magnitude. Applying this constraint **implies** usage of the *Constant* C_O^R *Transformation* (3.2.1), since the other approach (3.2.2) observes results of this velocity measurement. Using that approach would probably cause unstable behavior of the whole fusion algorithm.

Applying the constraint is done as follows: The normal of the ODOM plane expressed in the R frame yields

$$\mathbf{n}_{XY,R} = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.50}$$

where the angle β corresponds to the simplified rotation between the R and ODOM frames discussed in the *Caterpillar Odometry Aiding* section. The rotation of the velocity is done first by projection of the velocity vector into the XY plane

$$\hat{\mathbf{v}}'_{R,l,constr} = \lfloor \mathbf{n}_{XY,R} \rfloor \left(\lfloor \hat{\mathbf{v}}'''_{R,l} \rfloor \, \mathbf{n}_{XY,R} \right) \tag{3.51}$$

and then rescaling it so the magnitude is preserved

$$\hat{\mathbf{v}}_{R,l,constr} = \frac{\hat{\mathbf{v}}'_{R,l,constr}}{||\hat{\mathbf{v}}'_{R,l,constr}||} ||\hat{\mathbf{v}}'''_{R,l}|| \tag{3.52}$$

while for $\hat{\mathbf{v}}'''_{R,l} = [0,0,0]^T$, the result is a zero vector as well, avoiding the singularity.

Finally, since we know the whole trajectory element in the time of these computations, we choose to eliminate noise in the velocity originating from differentiation by a simple zero-phase filter realized by a two-way moving average filter. Its order was experimentally set to 3.

3.3.4. Comparison of ICP Aiding Approaches

The figures 3.9 and 3.10 demonstrate typical performance of the three proposed approaches. The *Incremental Position* approach is the simplest one from both the implementation and computation points of view; it does not require the algorithm to recompute part of the measurements history. However, as the Fig. 3.9 shows, it creates steps in the state estimation and it is unable to compensate for the drift in the Z axis.

To avoid the steps in the state estimation, the *Velocity Approach* was proposed. It increases number of measurements by expressing the change in position as a constant velocity. Yet, the idea of linear interpolation of the trajectory in order to be able to express the velocity caused the *corner cutting* effect clearly visible in the Fig. 3.10.

For the undesired effects of the two previous approaches, the **Trajectory Approach** was proposed. It compensates the *corner cutting* effect, does not show the estimate step behavior and moreover, it corrects the drift in the Z axis. For these reasons, **we choose this approach for the final fusion algorithm**. However, it brings two complications: a need to recompute the history of the state estimation since the previous ICP measurement (typically 300 samples) and the fact that a state estimate is used to modify a measurement. That contradicts the expectation of uncorrelated measurements and leads to a sub-optimal solution. This issue stays open for further discussion.

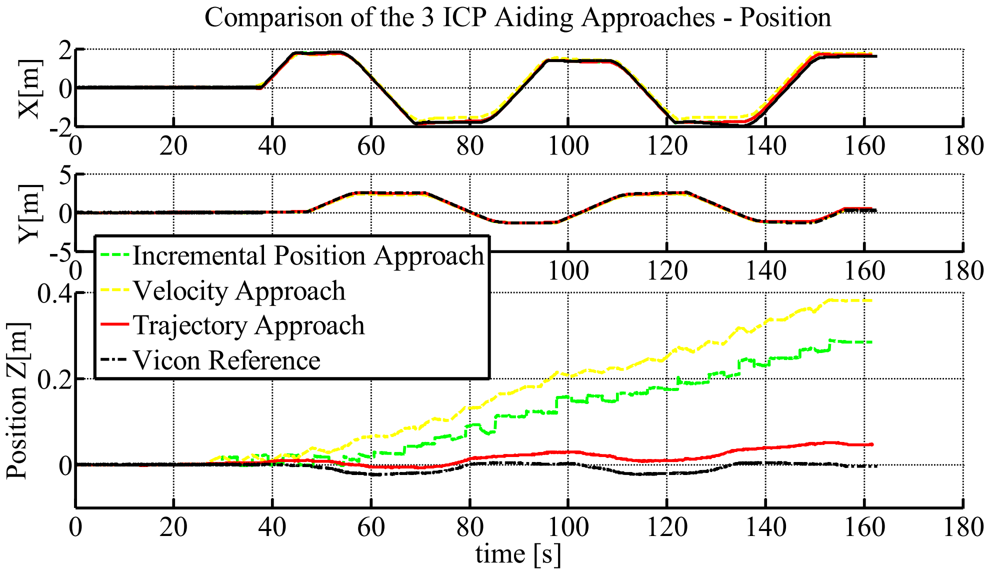


Figure 3.9. Comparison of the three ICP aiding approaches effects on position estimation. Note the apparent steps in the Z coordinate for the *Incremental Position* aiding approach.

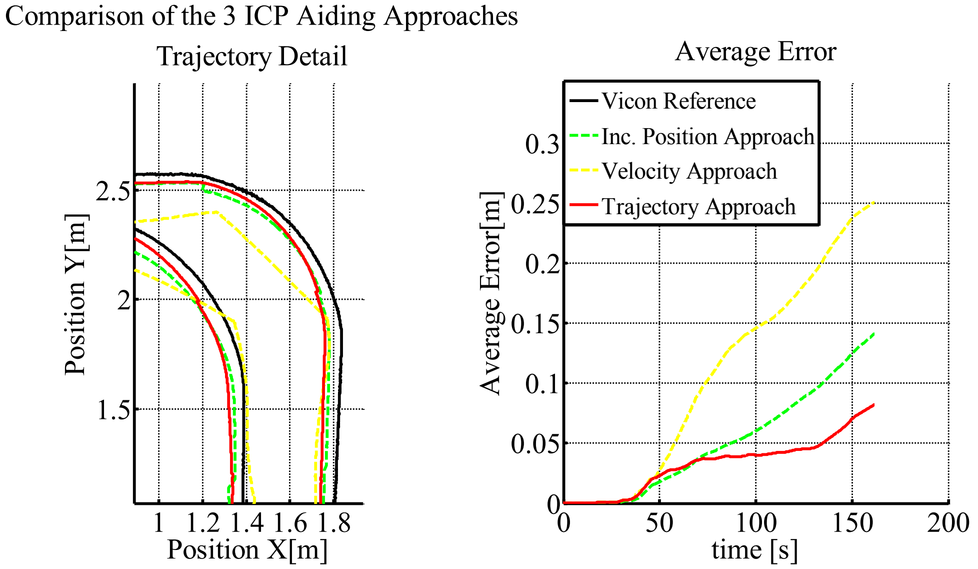


Figure 3.10. Comparison of the three ICP aiding approaches effects on the trajectory estimate and on the average error(5). Note the *corner cutting* effect of the *Velocity Approach.*

3.4. Monocular Visual Odometry

Visual odometry (VODOM) is an algorithm estimating translation and rotation of a camera body based on images recorded by the camera. The basic principle is demonstrated in the Fig. 3.12. A single scene is observed by a camera from distinct poses and the goal of the visual odometry is to estimate these poses based on the images recorded by the camera. For more detailed introduction into the problem, see [9, 10].

The NIFTi robot is equipped with an omni-directional PointGrey Ladybug®3 camera (Fig. 3.11), which composed 6 images captured by its 6 lenses to a panoramic picture. According to the robot visual odometry implementation author Jiří Diviš and his supervisor Tomáš Svoboda, using an omni-directional camera instead of a normal single-lens one leads to a more robust motion estimation. A reason for that is the fact that a significant number of the image features leave the field-of-view when a normal single-lens camera moves or rotates. This is not an issue in the case of the omni-directional camera which observes a whole scene around it. An example of the feature detection in an omni-directional picture is displayed in the Fig. 3.13.

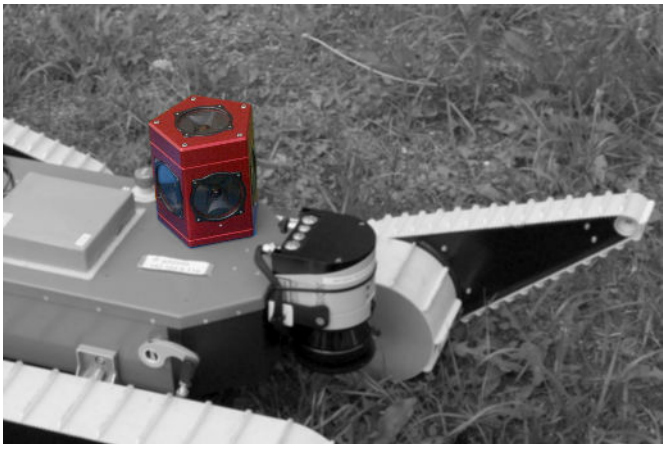


Figure 3.11. A PointGrey Ladybug®3 omni-directional camera mounted on the NIFTi Search&Rescue robot. It captures images from its 6 cameras and composes them into one panoramic picture.

However, modeling the set of the 6 physical cameras as a single omni-directional one causes a scale of the scene around the robot to be unobservable i.e. translations estimated by the visual odometry are relative to the first translation observed, whose magnitude is always set to be 1. From this reason, the current implementation of the fusion algorithm **utilizes only the rotation part** of the estimated motion, which is not affected by the scale. Exploiting the translational part is left for further work with the scale λ already being incorporated in the system state (2.1). Some conclusions concerning the scale estimation can be found in [17].

The incorporation of the VODOM measurements into the fusion system **is equivalent** to the ICP *Trajectory Approach* (3.3.3, the attitude aiding part), which incrementally processes the incoming attitude quaternions following the equations (3.30 - 3.35). The implementation of the VODOM attitude aiding is identical.

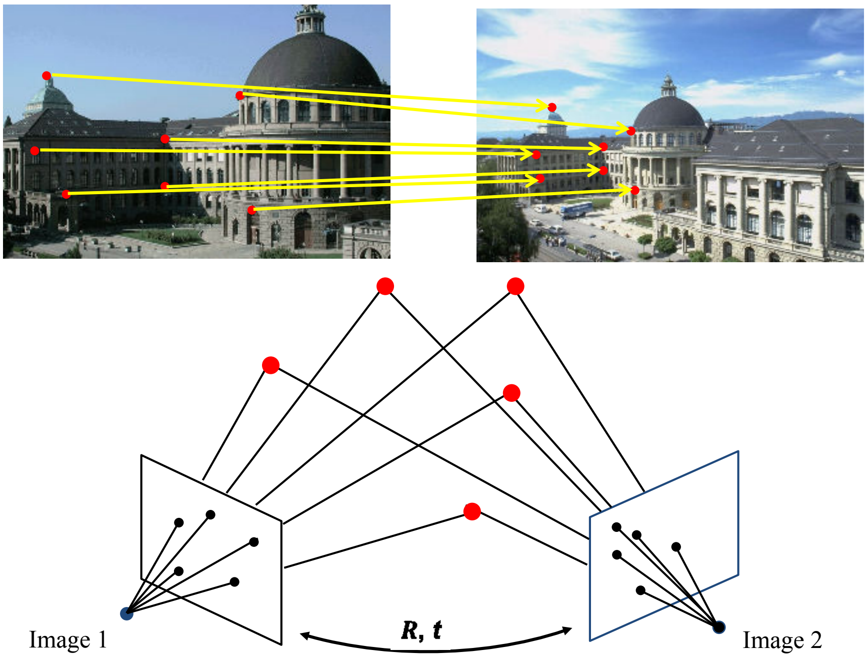


Figure 3.12. Visual odometry basic principle demonstration. A single scene is observed by a camera from different (and unknown) poses, selected features are searched for in the images and if found, they are paired between these images. Finally, camera poses are estimated based on geometrical constraints laid by these image features. (Picture adopted from Davide Scaramuzza's Visual Odometry Tutorial, http://robotics.ethz.ch/~scaramuzza/Davide_Scaramuzza_files/publications/ppt/Visual_Odometry_Tutorial.pptm)

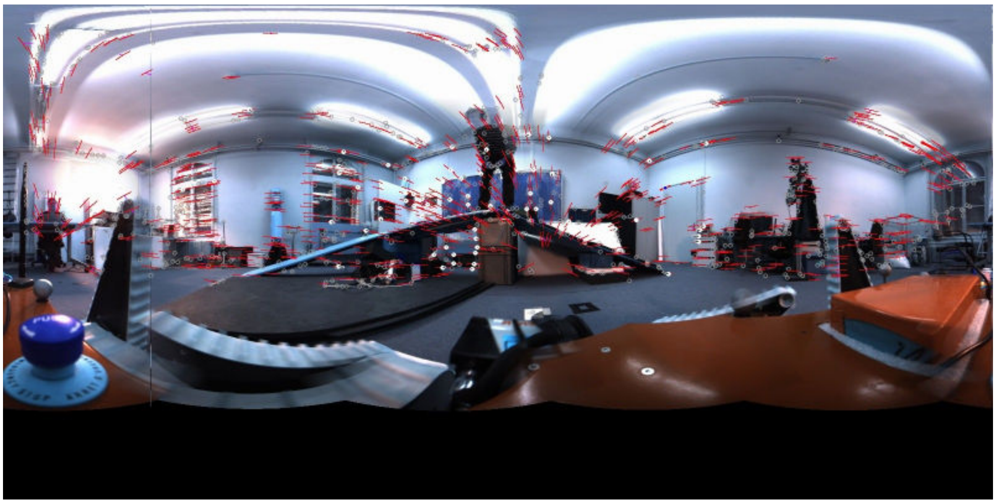


Figure 3.13. Panoramic picture composed of 6 images captured by the Ladybug$^{®}$3 omnicamera in a laboratory of ETH Zürich with correspondences between two consequent panoramic pictures as detected by the visual odometry algorithm implemented by Jiří Diviš for the NIFTi robot. Each red line connects a feature detected in this picture and its new position in the next frame. Directions of the lines indicate that the robot is moving forward. (Color representation is not correct due to the debugging nature of this image)

4. Extended Kalman Filter

The extended Kalman filter [18, 19], (Fig. 4.1) is a modification of the Kalman filter [20] , i.e. optimal observer minimizing variances of observed states. This modification is intended for systems which are non-linear yet the errors caused by this non-linearity can be neglected compared to the errors induced by measurement and the system noise [21]. It is left without proof that local observability [22, p.60] of the observed states ensures convergence of the estimation. For applications with no computational performance limitations, higher order extended Kalman filter or unscented Kalman filter [23] can be used - these modifications show faster convergence.

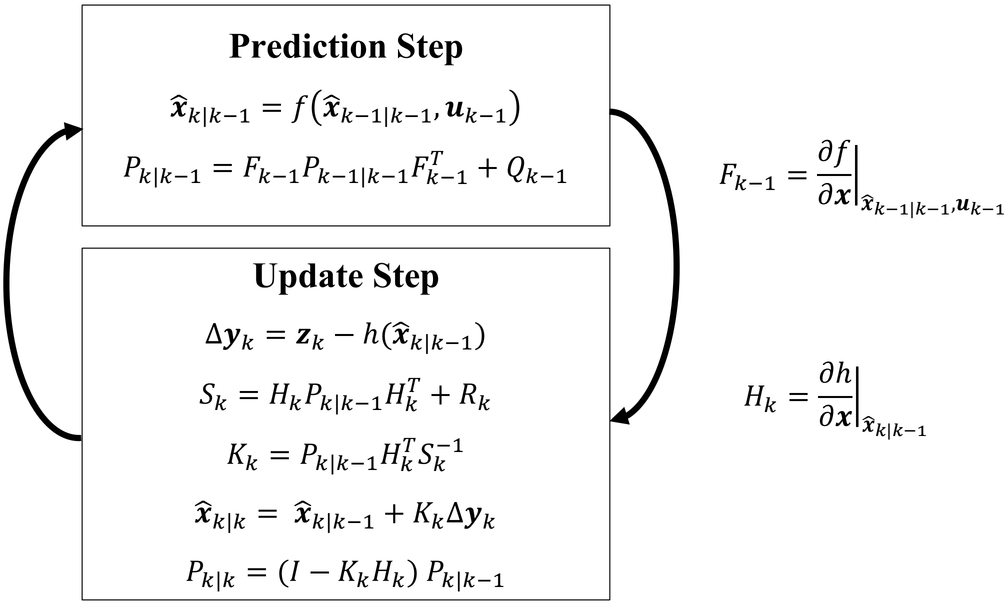

Prediction Step
$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1})$
$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1}$
Update Step
$\Delta \boldsymbol{y}_k = \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k|k-1})$
$S_k = H_k P_{k|k-1} H_k^T + R_k$
$K_k = P_{k|k-1} H_k^T S_k^{-1}$
$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + K_k \Delta \boldsymbol{y}_k$
$P_{k|k} = (I - K_k H_k)\, P_{k|k-1}$
$F_{k-1} = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}}$
$H_k = \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k|k-1}}$

Figure 4.1. Standard EKF computation flowchart, in the prediction step, the a priori state and the covariances are estimated, using the non-linear system equation $f()$ and its linearized matrix form F. In the update step, measurement residual $\Delta\mathbf{y}$ is obtained by comparing the incoming measurement $\mathbf{z}$ with its predicted counterpart. The residual covariance S and the Kalman gain K are evaluated and used to update the state and covariance to obtain the a posteriori estimates.

As indicated in the System Model chapter (2), the error state EKF is used in our approach - the state of the system is expressed as a sum of the current best estimate ($\hat{x}$) and some small error (Δx). The only difference compared to the standard EKF is that the linearized system matrices F and Q describe only the error state and the error state covariance propagation in time rather than the whole state and state covariance propagation in time. As explained in the chapter (2) and demonstrated in the Appendix A, this makes the matrices F and Q much easier to express. A flowchart demonstration the error state EKF computation is shown in Fig. 4.2.

The flowchart in the Fig. 4.2 can be decomposed into ten steps which also describe

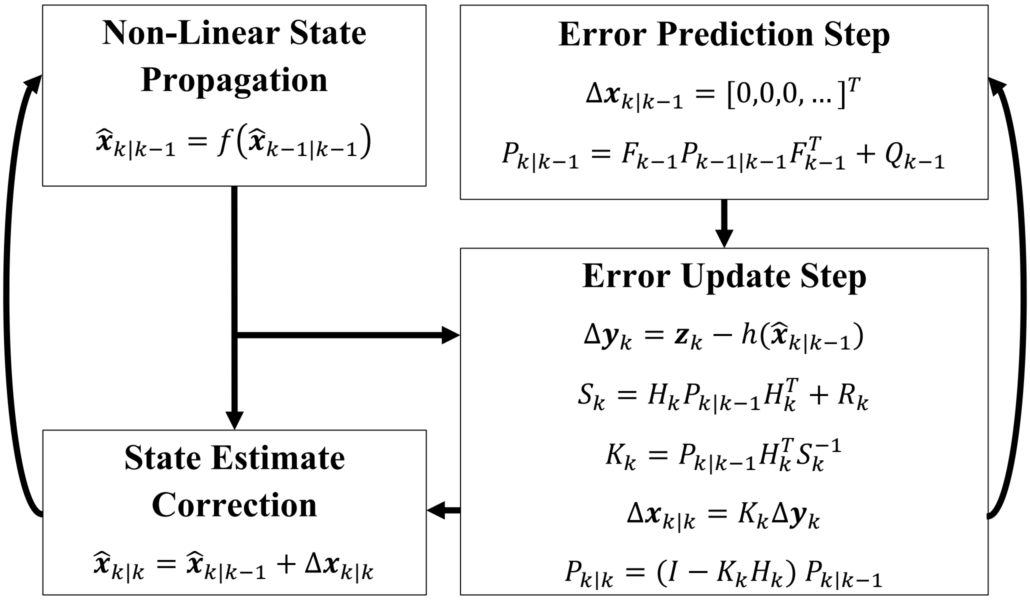

Non-Linear State Propagation
$\widehat{\boldsymbol{x}}_{k|k-1} = f(\widehat{\boldsymbol{x}}_{k-1|k-1})$
Error Prediction Step
$\Delta\boldsymbol{x}_{k|k-1} = [0,0,0,\ldots]^T$
$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1}$
Error Update Step
$\Delta\boldsymbol{y}_k = \boldsymbol{z}_k - h(\widehat{\boldsymbol{x}}_{k|k-1})$
$S_k = H_kP_{k|k-1}H_k^T + R_k$
$K_k = P_{k|k-1}H_k^TS_k^{-1}$
$\Delta\boldsymbol{x}_{k|k} = K_k\Delta\boldsymbol{y}_k$
$P_{k|k} = (I - K_kH_k)\,P_{k|k-1}$
State Estimate Correction
$\widehat{\boldsymbol{x}}_{k|k} = \widehat{\boldsymbol{x}}_{k|k-1} + \Delta\boldsymbol{x}_{k|k}$

Figure 4.2. This flowchart illustrates the error state EKF algorithm implemented in our fusion algorithm. Note that respecting the System Model chapter (2) the nonlinear system propagation equation f is a function of the state $\hat{x}$ only. Apart from the standard EKF, the F, Q and H matrices were not obtained by a direct linearization of the system nonlinear equations but as described in chapters 2 and 3.

the implementation of the error state EKF in the fusion sytem:

1. A new measurement (or measurements) arrive, there is a state estimate ($\hat{x}$) and its error covariance matrix (P) available from the previous step (or from the initialization if this is the first iteration)
2. The state estimate $\hat{x}$ is propagated in time using the nonlinear system equations
3. Continuous form of F and G matrices are evaluated based on the $\hat{x}$
4. The Van Loan discretization is used to evaluate the discrete form of F and Q
5. The error state covariance matrix P is propagated in time (based solely on the previous knowledge of the system)
6. Expected measurements are compared with the real ones, the difference is expressed in a form of a measurement residual Δy
7. An innovation matrix H expressing the measurement residual as a linear combination of the error state components is evaluated
8. Using the a priori estimate of P, H and variance of the sensors signals R, the Kalman gain matrix K is computed
9. The error state Δx is evaluated using the Kalman gain and the measurement residual, the a posteriori estimate of the error state covariance matrix P is evaluated as well
10. The a priori state estimate $\hat{x}$ is corrected by the estimate of the error Δx

4.1. Innovation Matrix H Construction

The cycle of the error state EKF could be repeated each time a single measurement arrives, yet, for performance reasons, we choose to group the measurements by the incoming IMU measurements, replacing the measurement times by the IMU ones. Each time a measurement arrives that is not an IMU one, it is delayed until the next IMU measurement arrives. The maximal time error caused by this grouping is $1/(2 \cdot 90)$s and thus it can be neglected compared to the measurement periods - the second fastest measurement is the track odometry measurement, which is generated with frequency of 15Hz. To group the measurements, they are stacked into one $\mathbf{z}'$ vector in a defined order. In the same order, the innovation matrix H' is constructed by stacking up the H matrices defined in the chapter 3. Finally, the measurement noise covariance matrix R' is constructed as a diagonal one from the $\mathbf{m}_{(.)}$ terms defined in the chapter 3.

The overall system including the error state EKF loop and the measurement sources is depicted in (Fig. 4.3).

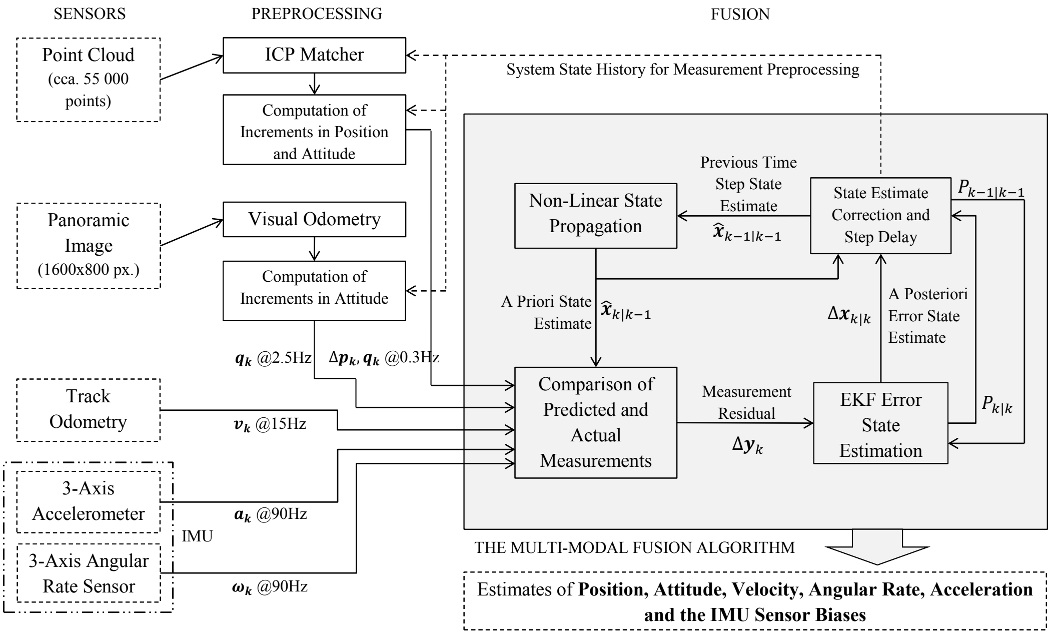

SENSORS
PREPROCESSING
FUSION
Point Cloud (cca. 55 000 points)
ICP Matcher
Computation of Increments in Position and Attitude
System State History for Measurement Preprocessing
Panoramic Image (1600x800 px.)
Visual Odometry
Computation of Increments in Attitude
Previous Time Step State Estimate
Non-Linear State Propagation
$\hat{x}_{k-1|k-1}$
State Estimate Correction and Step Delay
$P_{k-1|k-1}$
A Priori State Estimate
$\hat{x}_{k|k-1}$
$\Delta x_{k|k}$
A Posteriori Error State Estimate
q_k @2.5Hz
$\Delta p_k, q_k$ @0.3Hz
Track Odometry
v_k @15Hz
Comparison of Predicted and Actual Measurements
Measurement Residual
Δy_k
EKF Error State Estimation
$P_{k|k}$
3-Axis Accelerometer
IMU
a_k @90Hz
3-Axis Angular Rate Sensor
ω_k @90Hz
THE MULTI-MODAL FUSION ALGORITHM
Estimates of Position, Attitude, Velocity, Angular Rate, Acceleration and the IMU Sensor Biases

Figure 4.3. The overall system block schematics.

4.2. Application of the ICP Trajectory Aiding Approach

The ICP Trajectory Aiding Approach (3.3.3) minor drawback is the fact that it generates measurements for the time period that precedes the last incoming ICP measurement. To be able to use these measurements, the initial system state at the time of the previous ICP measurement is remembered as well as all the non-ICP measurements that arrived since that point. When the new ICP measurement arrives, the velocity measurements are generated and the whole time period between the ICP measurements is recomputed applying them. The cost of this approach is doubling the computation performance requirements.

5. Evaluation

This chapter analyses performance of the proposed fusion system. There are two metrics used, the first one, *Final Position Error* expressed in % of the total experiment length is often used in literature to give a basic idea about the localization approach accuracy. It is defined as

$$e_{rel} = \frac{||\mathbf{p}_l - \mathbf{p}_{ref,l}||}{\text{distance travelled}} \tag{5.1}$$

where l is the index of the last position sample $\mathbf{p}_l$ with the corresponding reference position $\mathbf{p}_{ref,l}$. The second metric, which provides more integral characteristics is *Average Position Error* defined as

$$e_{avg}(l) = \frac{\sum_{i=1}^{l} ||\mathbf{p}_i - \mathbf{p}_{ref,i}||}{l} \tag{5.2}$$

where $1 < l <$ (total number of samples). We prefer this metric, since it accounts errors along the whole trajectory. If we provide an output of this metric as a single number we assume $l =$ total number of samples.

Three sections follow, each one analyses the algorithm performance from a different point of view, the first one analyses in detail two selected experiments which illustrate capabilities of the fusion algorithm in three-dimensional environment with slippages and for the outdoor experiment, moving people and relatively long distance travelled (260m). The second section takes in account standard conditions experiments and makes statistics over these experiments for different combinations of modalities. The third section analyses situations where the experiment environment causes one or several fusion modalities to fail and impacts of these failures on the fusion algorithm performance.

5.1. Evaluation of Selected Experiments

5.1.1. 3D Structure Crossing with Caterpillar Tracks Slippage

This experiment simulated an unstructured environment with a slippery surface (Fig. 5.1) the robot had to traverse. Since the purpose of the NIFTi robot are Search and Rescue missions, such environments can be expected (e.g. collapsed buildings full of debris and dust that impairs traction on smooth surfaces such as exposed concrete walls or floors, mass traffic accidents with oil spills etc.).

While traversing the slippery surface, the basic track odometry inevitably failed with the tracks moving with significantly diminished traction. From that reason, trajectory (Fig. 5.2, yellow lines) and the state estimates resulting from the *IMU* and *ODOM* fusion showed unacceptable error growth (Fig. 5.3). Nevertheless, adding other modalities significantly improved the localization accuracy. The resulting state estimates for combination of all modalities are shown in Fig. 5.4, 5.5 and 5.6. The algorithm was capable to compensate for the track odometry failure and provided reliable trajectory estimate.

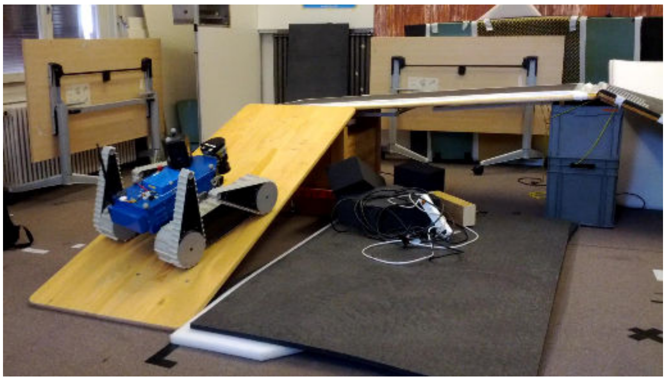


Figure 5.1. A part of the 3d structure used during this experiment. The robot is climbing up the wooden surface, which deteriorates the traction to the point the robot slides back down with each trial to steer. The reference system Vicon captures the whole scene.

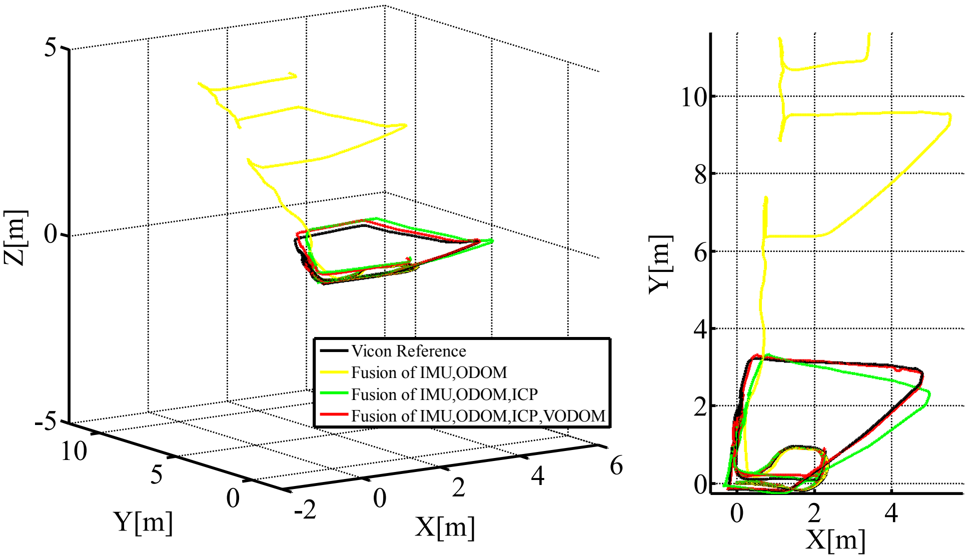


Figure 5.2. Trajectories resulting from the three fusion modalities combinations. Note that the the green trajectory corresponding to the *IMU+ODOM+ICP* combination can be improved by adding the visual odometry attitude measurements (resulting in the red one).

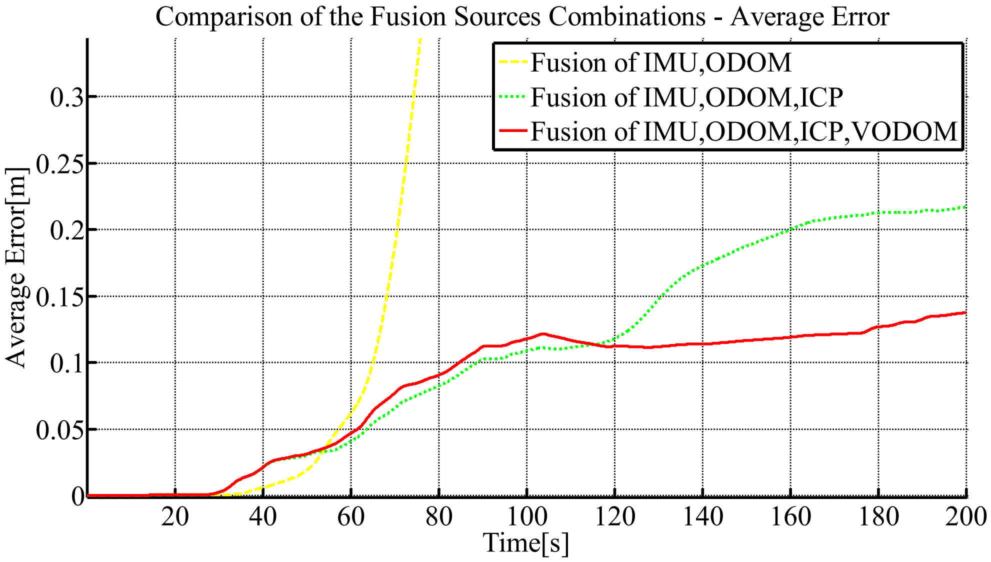


Figure 5.3. Average errors for the fusion modalities combinations. It is clear the basic dead-reckoning *IMU+ODOM* combination fails the moment the robot enters the slippery surface.

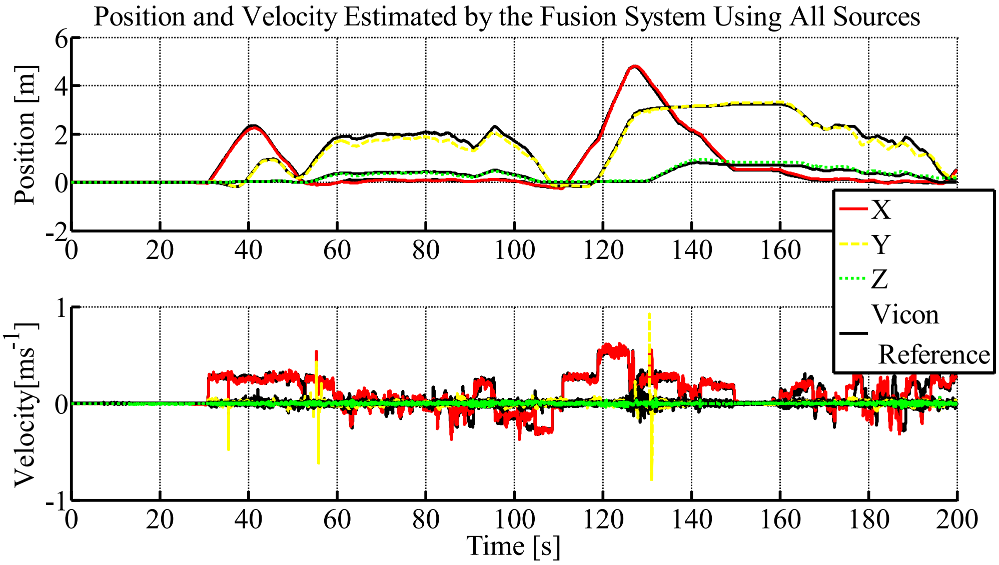


Figure 5.4. The position and velocity estimates for the best combination *IMU+ODOM+ICP+VODOM*. The result closely follows the Vicon reference (black) even though the complete track odometry failure.

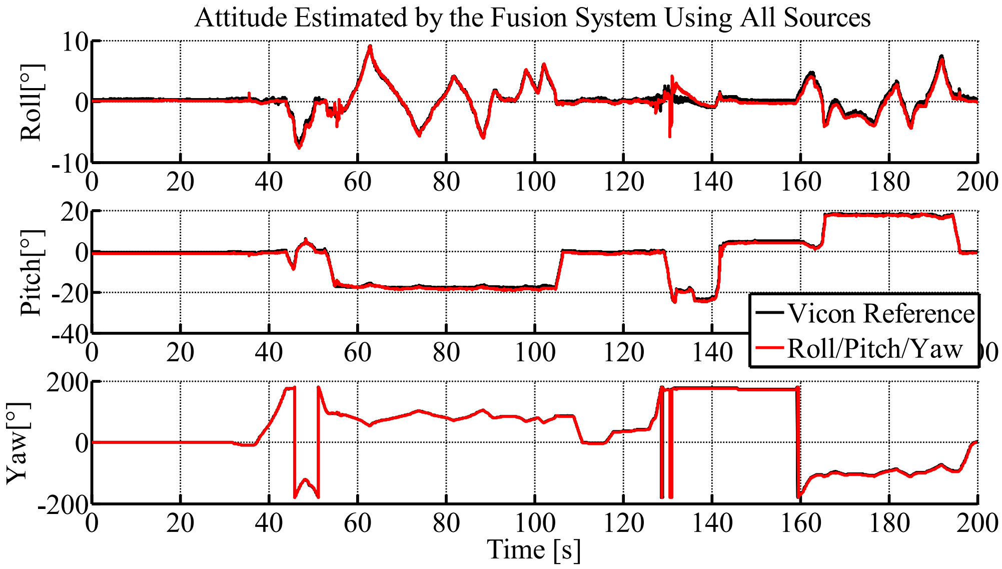


Figure 5.5. The attitude estimates for the best combination *IMU+ODOM+ICP+VODOM* with the Vicon reference.

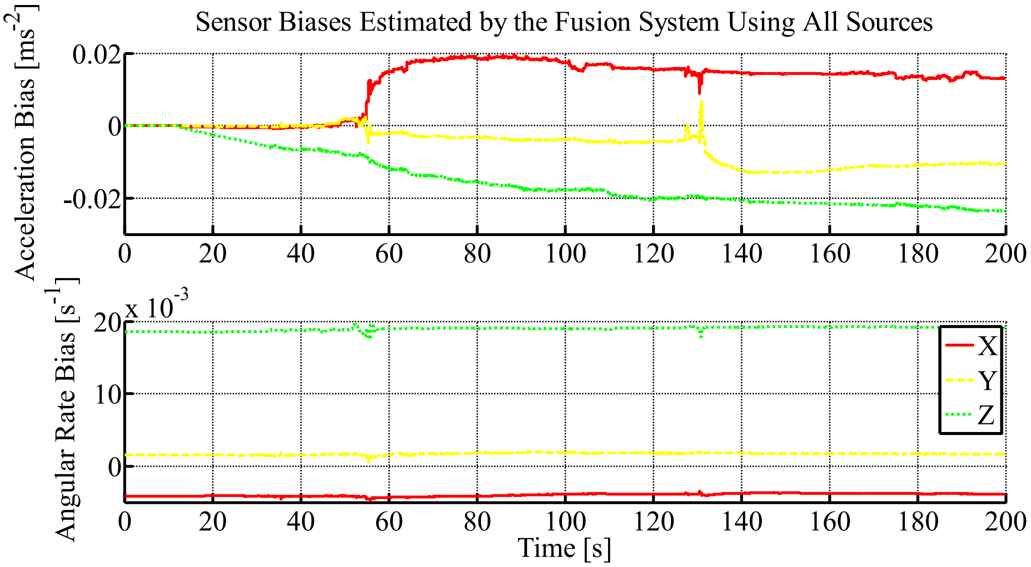


Figure 5.6. The sensor biases estimated for the best combination *IMU+ODOM+ICP+VODOM*. The angular rate sensor biases are initialized during a 10-second initial calibration (the robot is left steady and the mean values of the angular rate signals are obtained).

5.1.2. Outdoor Experiment

This experiment took place on the Clausiusstrasse street (Fig. 5.7) in Zurich next to the main ETH building. The purpose of this experiment was to test the ICP and VODOM performance in an open area. While the VODOM performs better outside because of the camera features behaving closely to the camera geometrical model, the ICP - compared to a closed room - missed a significant amount of spatial information (the laser range is limited approximately to 50 meters, there is no ceiling etc.). The Leica theodolite provided reference during this experiment.

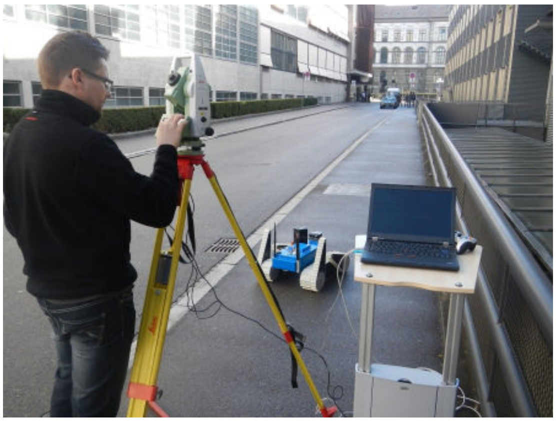


Figure 5.7. The experiment setup in the street, the Leica reference theodolite is just being set up.

The outcome (and the shape of the trajectory) is displayed below (Fig. 5.8 – 5.12) demonstrating the improvement of performance by adding the modalities. The basic dead-reckoning combination showed drift in the yaw angle caused by the angular sensor noise integration. That one could be corrected by adding the VODOM attitude measurements correcting this drift (visual odometry is not drift-free, yet performs better than the angular rate integration). Still, that enhanced combination suffered from the inaccurate track odometry velocity measurements (the green line in the Fig. 5.8). That problem was corrected by adding the ICP modality into the fusion scheme resulting into the best localization estimates.

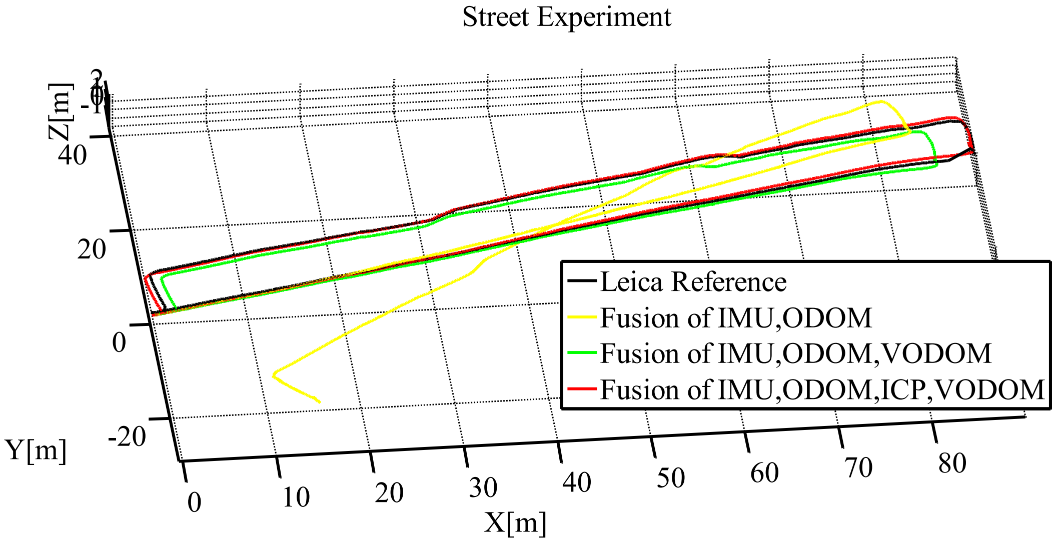


Figure 5.8. Trajectories resulting from different modalities combinations.

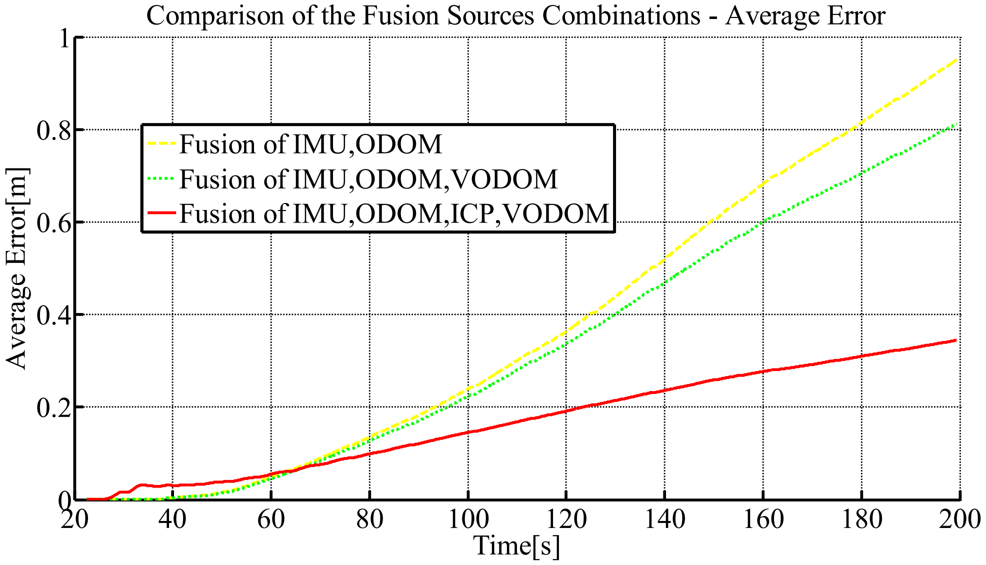


Figure 5.9. Comparison of the three combinations in terms of the average error. Apparently, the solution shows increasing error for all of the modalities combinations, yet the last one performs best.

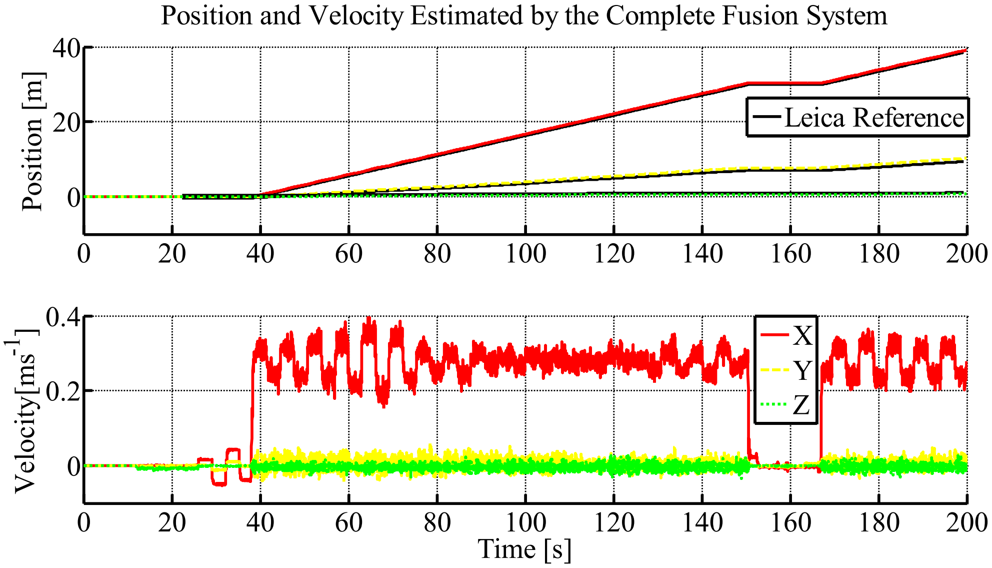


Figure 5.10. Position and velocity estimates for the best combination of the modalities (*IMU+ODOM+ICP+VODOM*). The velocity has no reference since the Leica reference system provides no attitude information and thus, we cannot express the velocity in the R frame without using our attitude estimates which is not desirable.

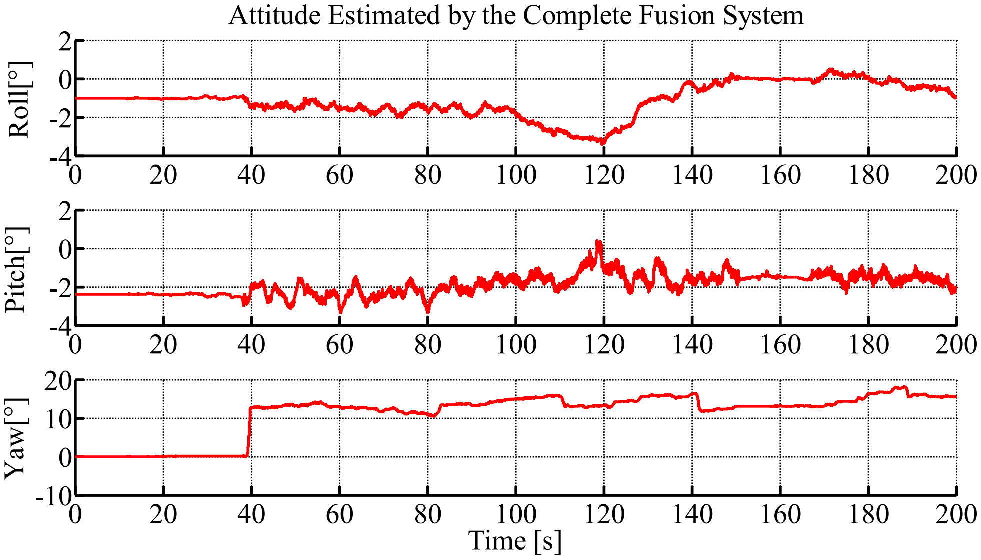


Figure 5.11. Attitude estimates for the best fusion modalities combination.

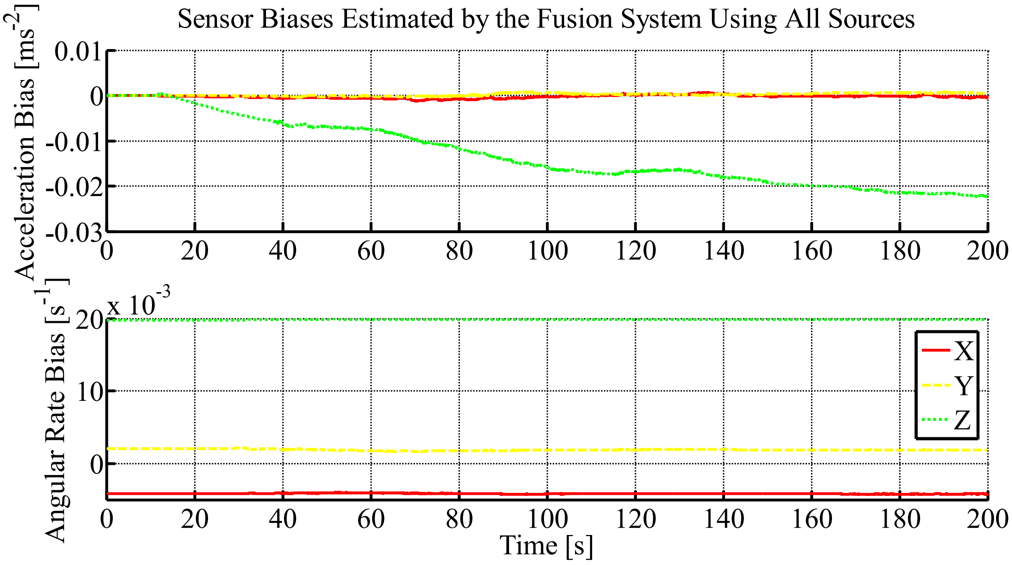


Figure 5.12. Attitude estimates for the best fusion modalities combination (*IMU+ODOM+ICP+VODOM*).

5.2. Fusion Algorithm Performance under Standard Conditions

To verify the fusion algorithm performance, experiments with standard conditions were selected (i.e. experiments in the Vicon laboratory with the standard robot system setup traversing the 3D structure or driving on a flat ground; all outdoor and indoor experiments with the Leica reference), four fusion modalities combinations were chosen: *IMU + ODOM*, *IMU + ODOM + VODOM*, *IMU + ODOM + ICP* and *IMU + ODOM + VODOM + ICP* and using the metrics (5.1, 5.2), statistical performance was obtained. The results are shown in the Tables (5.1 – 5.4), the group of all selected experiments was divided into two groups by the reference type.

Summing up the results, the Vicon laboratory experiments (Table 5.1 and 5.2) came out well while using the ICP localization in one of the last two combinations. Results of these were comparable, the standard deviation ranges intersected. The results for the *IMU+ODOM+VODOM* and *IMU+ODOM* were slightly worse and again, comparable taking standard deviation ranges in account. From this Vicon laboratory set of experiments, we concluded that the main improvement brought by the fusion system was the ability to correct slippage of the caterpillar tracks by the ICP position measurement modality.

Analyzing the Leica reference experiments (Tables 5.3 and 5.4), we observed strongly diminished performance while using the VODOM modality. It needs to be said that the environments these experiments were performed in were far from the ideal ones for the visual odometry algorithm (except for the *street* experiments, where the visual odometry worked perfectly). The indoor environments cause geometrical problems because for the visual odometry implemented for the omni-directional camera, it is desirable that all image features lay as far from the camera as possible. The outdoor experiments were performed in a park (Fig. 5.13) and such an environment complicates feature pairing between consequent camera images because of all the repetitive textures as leaves, branches etc.

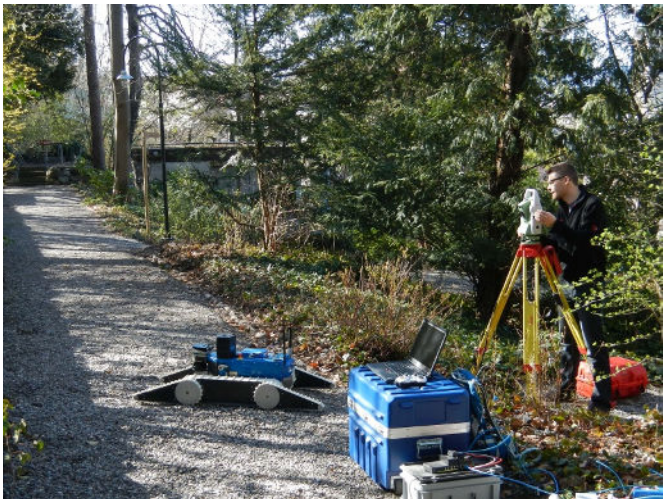


Figure 5.13. The outdoor experiment setup in the park below the main ETH building in Zürich.

However, the visual odometry implementation for the NIFTi robot is still being improved. Nevertheless, the fusion algorithm, given the right fusion modalities used, shown average position error comparable to the size of the robot which was a threshold for considering the localization successful - especially for the outdoor experiments in the park with the gravel paths and steep stairs.

			Final Position Error in % of the Distance Travelled			
Exp.	Distance Traveled [m]	Exp. Duration [s]	Odometry, IMU	Odometry, IMU, VODOM	Odometry, IMU, ICP	All
0.1	29.69	162	1.79	1.29	0.78	1.09
0.2	47.42	254	2.17	4.09	1.71	0.97
0.3	36.52	186	1.99	2.52	0.36	0.48
0.4	48.74	244	3.15	2.74	0.50	0.13
0.5	29.40	237	2.22	2.14	0.42	0.51
1.1.a	82.10	585	2.51	6.19	0.90	4.71
1.1.b	74.64	452	2.05	3.55	0.98	1.10
1.2.a	74.65	387	1.70	1.53	2.28	0.90
1.3.a	30.57	194	1.98	2.74	1.59	1.75
1.3.b	26.58	287	2.67	2.05	1.90	1.07
1.3.c	26.57	236	1.53	1.98	0.77	0.86
1.3.d	26.96	208	1.25	2.47	0.95	1.79
1.3.e	29.13	211	1.27	1.35	0.88	1.11
1.3.f	26.35	180	1.37	1.49	0.94	2.58
1.4.g	40.23	240	6.58	7.01	0.88	1.04
1.8.a	21.01	167	5.26	5.05	0.61	0.68
1.8.b	19.04	209	5.94	5.91	0.55	0.78
2.1.a	10.95	405	3.44	2.78	2.15	2.04
2.1.b	8.65	238	2.87	2.76	1.36	1.44
2.2.a	9.36	284	4.14	3.46	1.83	1.78
2.2.b	9.02	282	2.90	3.18	2.73	2.61
2.3.a	10.82	308	3.79	3.24	1.43	1.39
2.3.b	9.45	237	5.36	5.40	2.66	2.60
2.5.b	12.75	204	2.65	3.04	2.66	3.43
2.5.c	7.81	179	1.58	1.70	2.82	3.12
2.5.d	10.85	165	3.85	4.45	3.25	1.66
3.7.a	10.83	163	2.36	1.85	0.62	0.63
3.7.b	12.79	237	15.42	14.86	2.48	2.52
3.7.c	12.07	239	28.42	26.32	2.89	3.36
Averages and Standard Deviations			4.21 ± 5.39	4.38 ± 4.98	1.51 ± 0.89	1.66 ± 1.07

Table 5.1. Table summing up selected fusion modalities combinations performance when used in the fusion system. The metric is the Final Position Error expressed in percents of the overall distances travelled during each experiment. The experiments are labelled by their IDs, description of each one can be found in the appendix C

			Average Position Error [m]			
Exp.	Distance Traveled [m]	Exp. Duration [s]	Odometry, IMU	Odometry, IMU, VODOM	Odometry, IMU, ICP	All
0.1	29.69	162	0.31	0.36	0.08	0.14
0.2	47.42	254	0.57	0.94	0.36	0.21
0.3	36.52	186	0.38	0.51	0.11	0.11
0.4	48.74	244	0.66	0.66	0.13	0.09
0.5	29.40	237	0.16	0.16	0.05	0.06
1.1.a	82.10	585	0.78	3.26	0.31	2.43
1.1.b	74.64	452	0.87	1.13	0.29	0.36
1.2.a	74.65	387	0.69	0.67	0.51	0.39
1.3.a	30.57	194	0.31	0.34	0.17	0.19
1.3.b	26.58	287	0.38	0.49	0.22	0.28
1.3.c	26.57	236	0.18	0.22	0.11	0.14
1.3.d	26.96	208	0.17	0.30	0.14	0.20
1.3.e	29.13	211	0.16	0.31	0.12	0.15
1.3.f	26.35	180	0.22	0.32	0.23	0.21
1.4.g	40.23	240	0.88	0.90	0.18	0.37
1.8.a	21.01	167	0.33	0.30	0.08	0.08
1.8.b	19.04	209	0.38	0.38	0.06	0.06
2.1.a	10.95	405	0.15	0.14	0.14	0.15
2.1.b	8.65	238	0.10	0.10	0.10	0.09
2.2.a	9.36	284	0.12	0.13	0.09	0.09
2.2.b	9.02	282	0.13	0.13	0.12	0.12
2.3.a	10.82	308	0.15	0.15	0.09	0.09
2.3.b	9.45	237	0.27	0.27	0.12	0.12
2.5.b	12.75	204	0.15	0.17	0.16	0.25
2.5.c	7.81	179	0.08	0.08	0.07	0.08
2.5.d	10.85	165	0.36	0.34	0.15	0.16
3.7.a	10.83	163	0.22	0.23	0.06	0.07
3.7.b	12.79	237	1.06	1.06	0.14	0.15
3.7.c	12.07	239	1.93	1.94	0.17	0.18
Averages and Standard Deviations			0.42 ± 0.40	0.55 ± 0.66	0.16 ± 0.10	0.24 ± 0.43

Table 5.2. The same experiments as in the table 5.1 but using the Average Position Error metric. Note that the results are similar comparing the first two and the last two modalities combinations. In this case, the units of the metric are meters, we can therefore compare the results to the size of the NIFTi robot, which is approximately 1 meter long - the average error of the localization is well below the size of the robot.

			Final Position Error in % of the Distance Travelled			
Exp.	Distance Traveled [m]	Exp. Duration [s]	Odometry, IMU	Odometry, IMU, VODOM	Odometry, IMU, ICP	All
basement 1	120.62	825	2.08	36.73	1.83	38.71
basement 2	175.67	853	1.37	11.44	2.42	5.96
hallway straight	159.42	738	1.10	14.10	0.43	7.69
street 1	135.18	584	2.78	0.81	0.24	0.80
street 2	259.86	992	9.74	0.50	0.26	0.28
park big loop	145.31	918	2.65	48.98	1.03	33.57
park small loop	88.20	601	1.94	67.95	1.25	43.45
park straight	99.29	560	1.20	62.61	0.62	50.92
2 floors	238.28	1010	9.10	2.18	0.58	2.56
2 floors opposite	203.23	1107	3.23	56.67	0.51	52.05
Averages and Standard Deviations			3.52 ± 3.19	30.2 ± 27.3	0.92 ± 0.72	23.6 ± 22.0

Table 5.3. Relative error for the Leica-referenced experiments. The unflattering values in the VODOM columns are probably caused by the environments unsuited for the VODOM localization algorithm. The errors originating from low number of paired features probably lead to behavior described in the 5.3.2 fail case.

			Average Position Error [m]			
Exp.	Distance Traveled [m]	Exp. Duration [s]	Odometry, IMU	Odometry, IMU, VODOM	Odometry, IMU, ICP	All
basement 1	120.62	825	3.31	20.69	0.65	17.09
basement 2	175.67	853	6.19	22.47	2.68	10.74
hallway straight	159.42	738	2.77	11.76	0.37	7.23
street 1	135.18	584	2.14	0.99	0.41	0.57
street 2	259.86	992	8.92	2.21	0.50	0.83
park big loop	145.31	918	1.79	29.51	0.72	21.38
park small loop	88.20	601	1.55	28.20	0.59	19.48
park straight	99.29	560	0.57	46.70	0.83	47.86
2 floors	238.28	1010	7.83	4.76	0.51	3.98
2 floors opposite	203.23	1107	3.87	45.01	1.33	42.63
Averages and Standard Deviations			3.90 ± 2.82	21.2 ± 16.6	0.86 ± 0.70	17.2 ± 16.6

Table 5.4. Average position error statistics regarding the experiments referenced by the Leica system. The metric confirms that the visual odometry performed significantly worse than other modalities.

5.3. Fail-Case Analysis

In this last section of the *Evaluation* chapter, we intentionally made some of the fusion modalities fail and observed the overall fusion system response. The aim was to discover the true limits of the system under realistic failure conditions. In the last section (5.3.5), we also analysed several ICP localization algorithm features and their impact on the fusion

5.3.1. Unintended Robot Frame Motion

In this experiment, the robot was pushed and pulled on a trolley through the laboratory (Fig. 5.14) to cause the track odometry to fail completely as shown in Fig. 5.15 and 5.16.

In the case of a simple dead-reckoning combination *IMU+ODOM*, the robot estimated no changes in position. However, adding other modalities compensated this and the resulting average error changed from 1.8m to 8cm. Nevertheless, this result was conditioned by a proper function of the ICP algorithm, which partially relies on the position estimates provided by the odometry system of the robot - in this case, we moved the robot very slowly to allow the ICP to stitch laser scans correctly even though the odometry system provided incorrect position estimates.

In the case of higher velocities i.e. bigger distances between consecutive laser scans, the ICP would fail to converge.

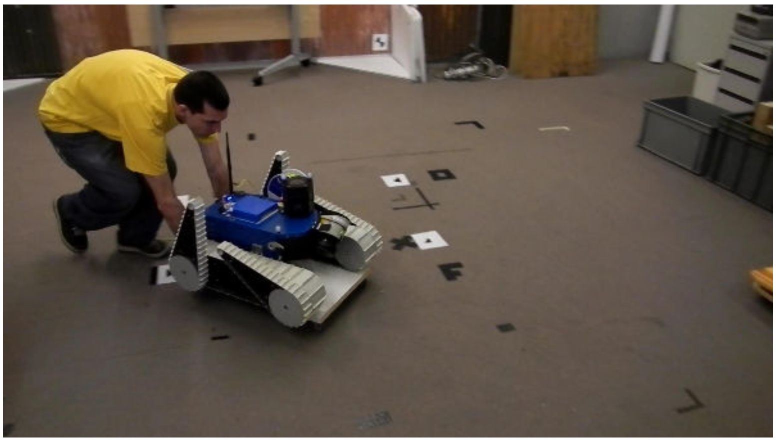


Figure 5.14. There is a trolley beneath the robot, which is pushed forward through the laboratory.

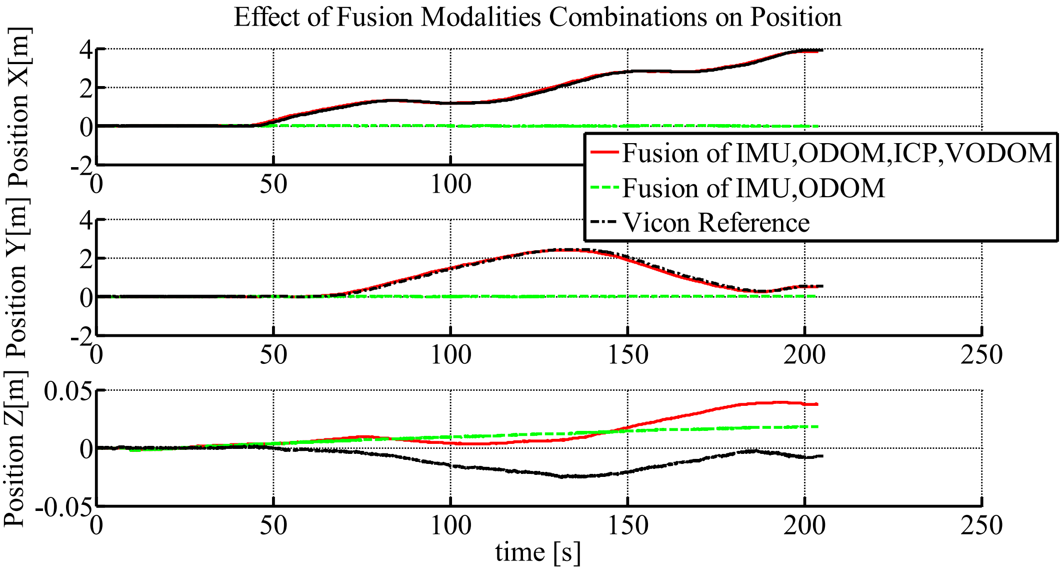


Figure 5.15. The position estimated by the simple dead-reckoning combination *IMU+ODOM* compared to the fusion of all localization sources.

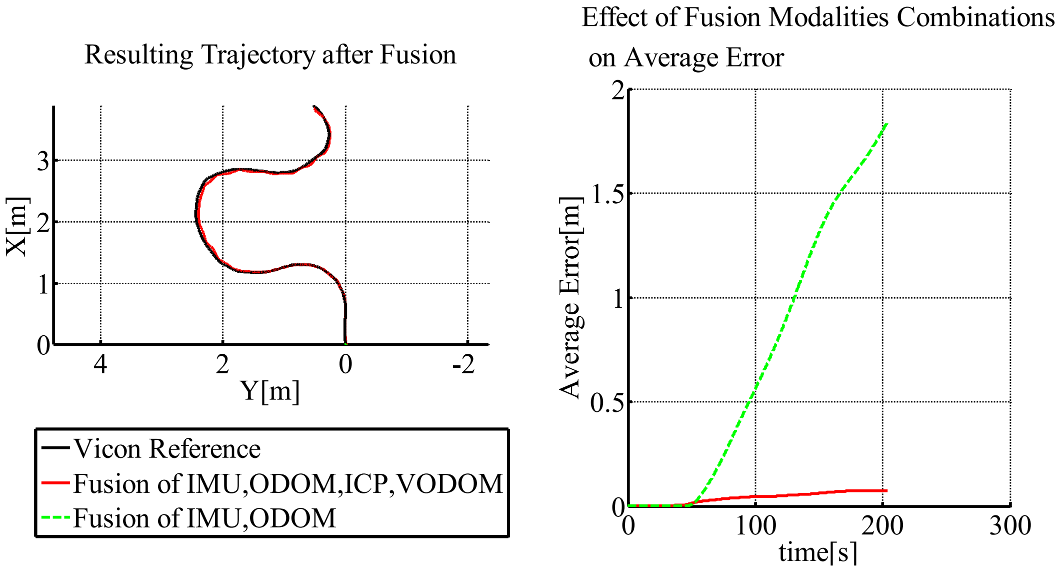


Figure 5.16. Trajectory estimated using the complete fusion system and average errors for both combinations of the modalities.

5.3.2. Blocked Camera

This experiment simulated partial loss of the field of view of the omni-camera; the most probable cause would be dirt or dust, which must be taken in account in the case of a search&rescue robot. One of the omni-camera lenses was blindfolded (Fig. 5.17) and the robot was navigated over the 3D structure.

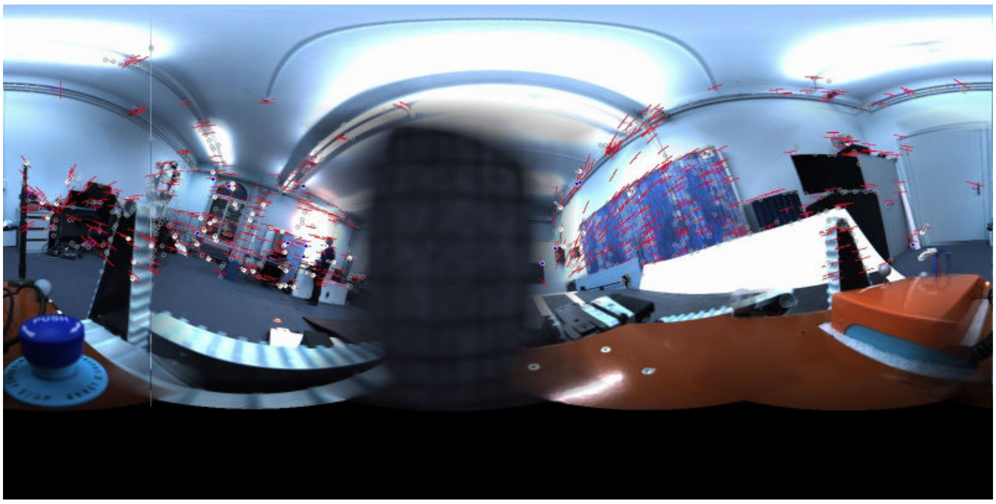


Figure 5.17. One lens of the omni-camera has been blindfolded to simulate worsen visual conditions or a view blocked by dust or dirt.

In this case, the insufficient number of image features caused the visual odometry to estimate incorrect attitude. This attitude propagated into the state estimate and made the fusion algorithm fail (Fig. 5.18 – 5.20). Although the fusion algorithm had proven to be robust against the track odometry failures so far, it was found sensitive to attitude measurement errors, originating both in the VODOM and ICP modalities.

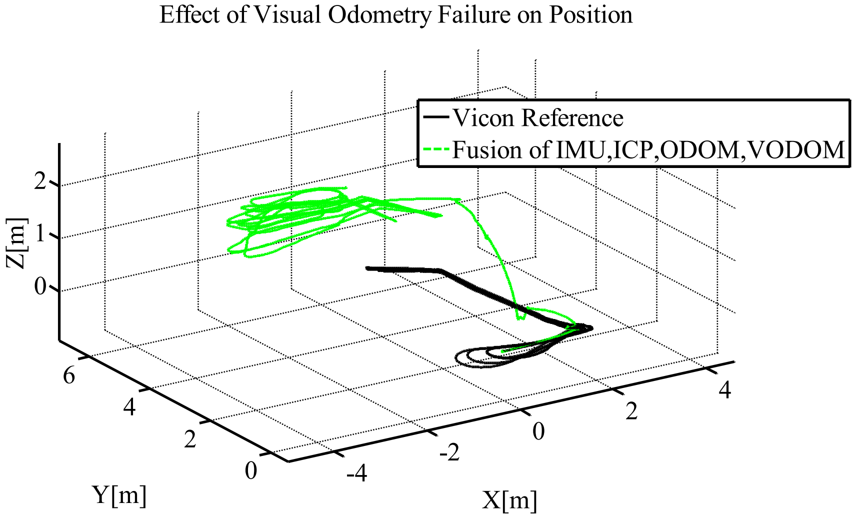


Figure 5.18. The trajectory deteriorated by a faulty VODOM measurement.

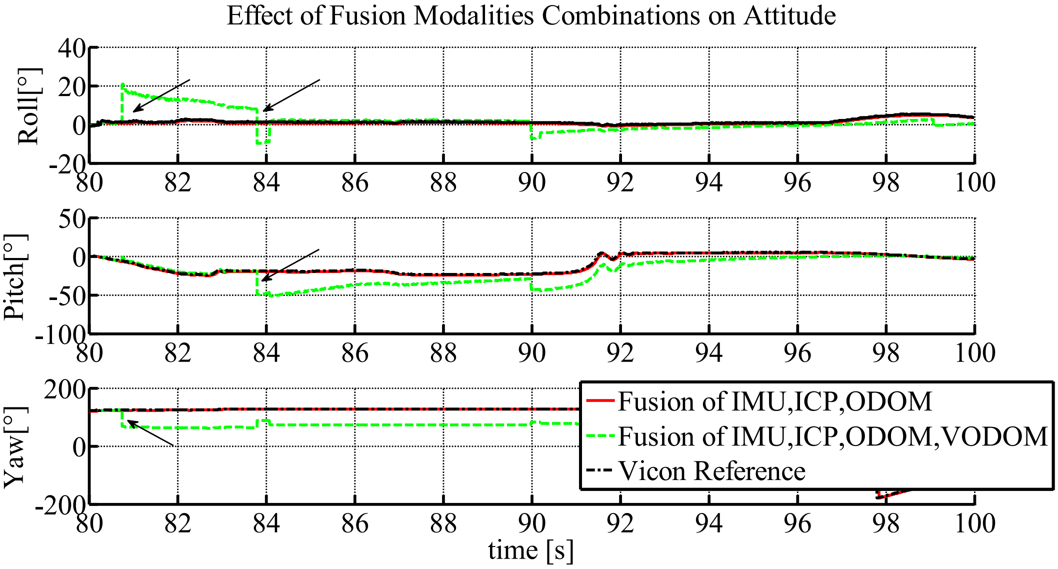


Figure 5.19. Comparison of attitude estimated using the whole fusion scheme including the visual odometry and one excluding it from the fusion. The black arrows mark points where the faulty measurements arrived. Since the *roll* and *pitch* angles are observable from the *IMU+ODOM* measurements [6], they converge back to the correct value. However, the yaw angle rests diverged.

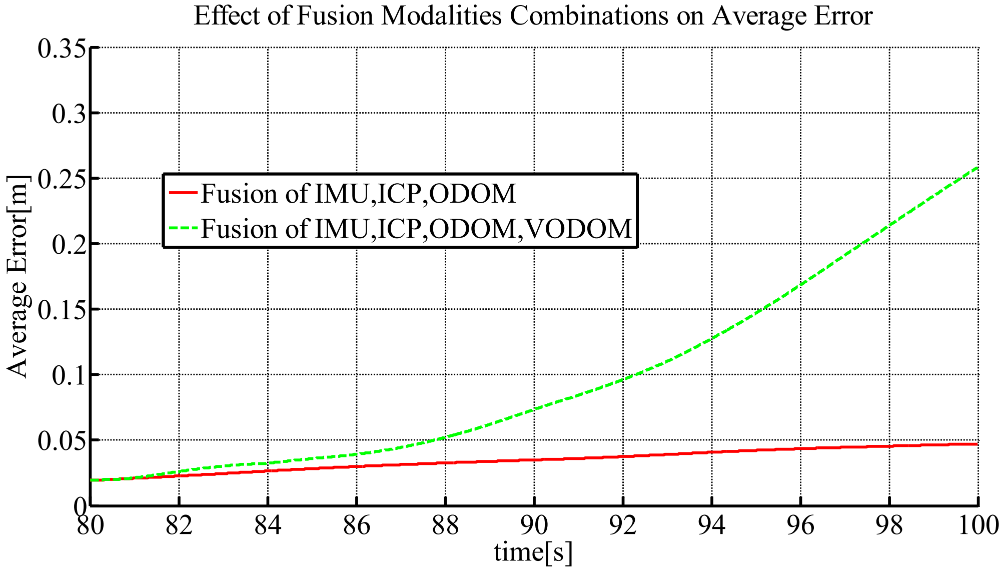


Figure 5.20. The effect of the faulty VODOM measurement on the average error, compared to the fusion scheme excluding the VODOM.

5.3.3. Laser Scanner Outages

Although the robot is otherwise mechanically robust, the rotational mechanism of the laser rangefinder assembly is actually its bottleneck prone to malfunction in harsh conditions. This fail case simulates outages in generating 3D pointclouds; the ICP algorithm stops generating measurements in that case and waits for a new laser scan to come. The fusion algorithm was designed and tested for ICP measurement period 3 seconds; this experiment tested its performance for ICP measurement outages longer than 40 seconds (vertices of the blue trajectory in the left part of the Fig. 5.21 are the ICP position measurements).

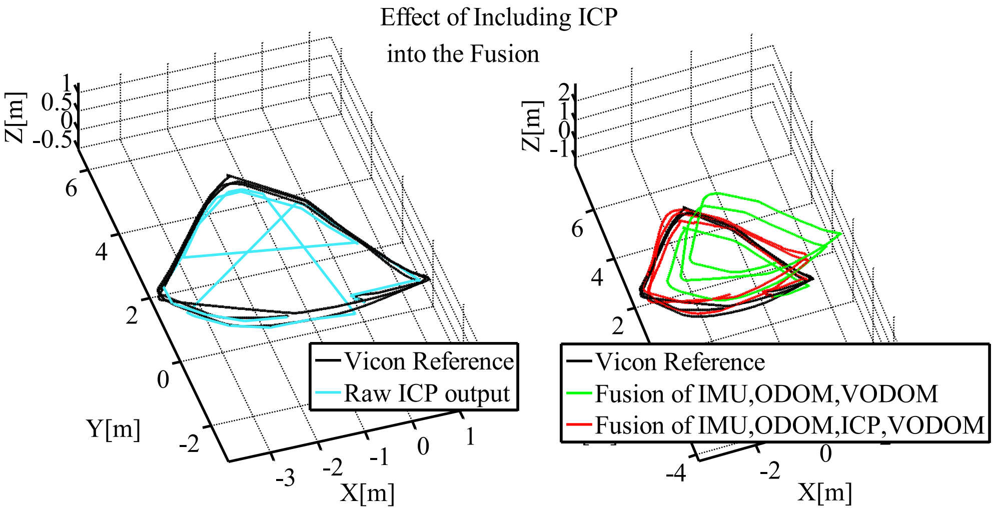


Figure 5.21. 3D plots of the sparse ICP position measurements (left) and the impact of the ICP modality added into the fusion scheme (right).

The fusion algorithm dealt with the laser outages satisfactorily. Although sparse, the ICP modality diminished average error of the position estimate compensating for the track odometry slippage (Fig. 5.22).

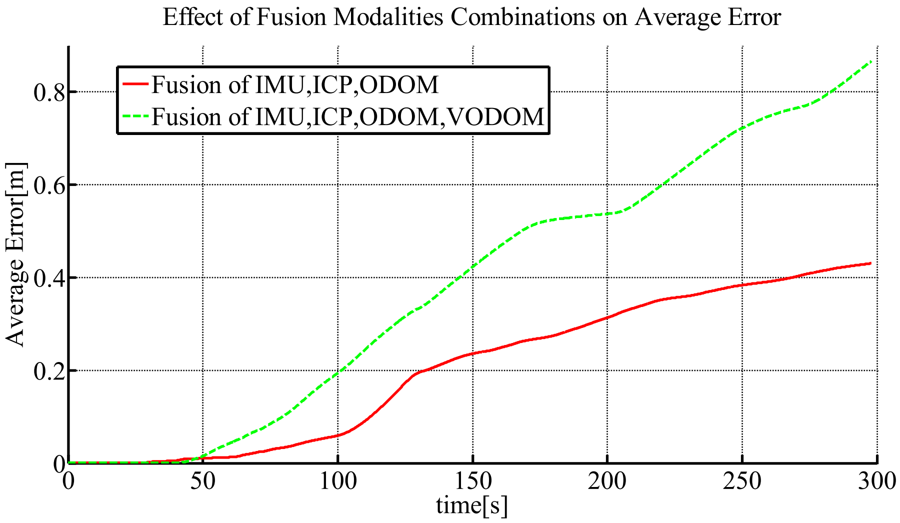


Figure 5.22. The two modalities combinations effect on average error.

5.3.4. Moving Object and Limited Laser Range

This experiment was performed to analyze the impact of faulty ICP position measurements on the fusion system performance. However, to obtain reference data, we were forced to perform it in the Vicon laboratory, which is almost a perfect environment for the ICP localization. To make it generate faulty measurements, we artificially reduced the laser maximal range to two meters preventing it from reaching ceiling and walls of the room. Also, a big board (Fig. 5.23) was moved back away from the robot after it got into the robot's laser scanner range (artificially set to the two meters) confusing the ICP algorithm - this was supposed to simulate dynamically changing environment. The step in the ICP position measurement is visible in the 3D plot in the Fig. 5.24.

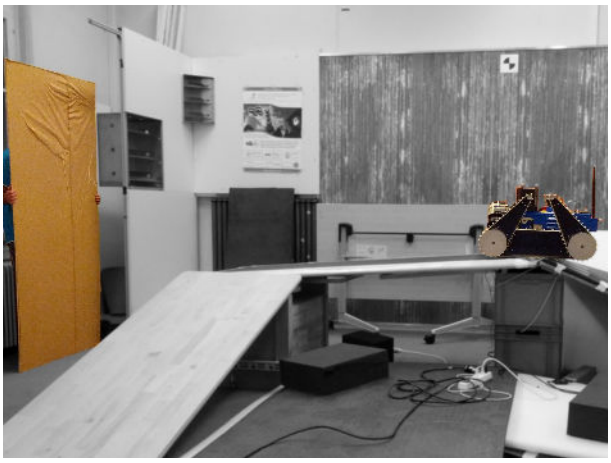


Figure 5.23. The big paper board was held on one spot until the robot reached it and then moved back away from the robot as depicted here. This artificial moving wall confused the ICP localization.

Although we caused this ICP localization failure artificially, a failure like this might occur in areas filled with smoke or dust, both impenetrable by the laser beam. The moving wall may represent moving rescue workers or firemen at the place of the deployment of the robot.

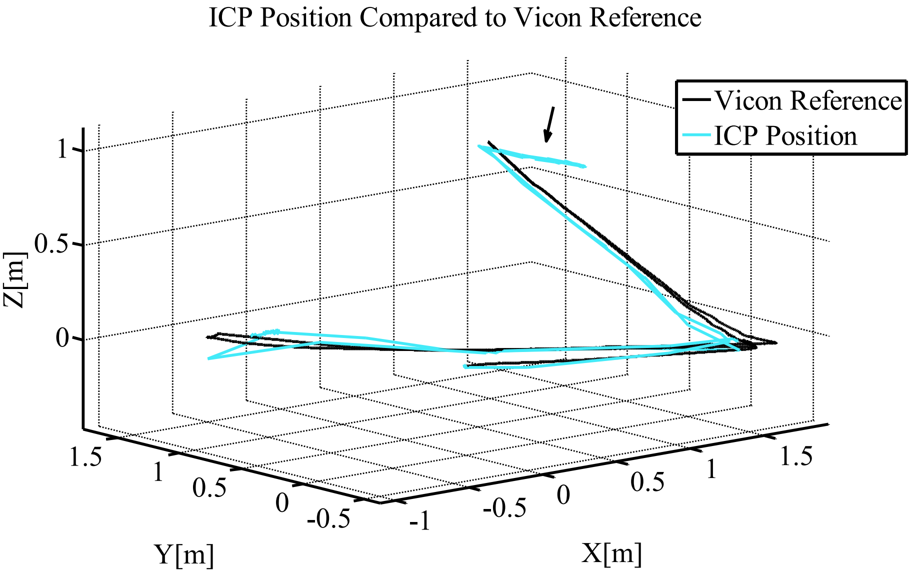


Figure 5.24. 3D plot with an apparent erroneous step in the ICP position measurement.

The results of the experiment showed similar results as in the VODOM failure scenario. The fusion algorithm - at its current implementation - could not handle these crucial ICP failures, it accepted them and propagated into the system state. The result were undesired errors in localization (Fig. 5.25 and 5.26).

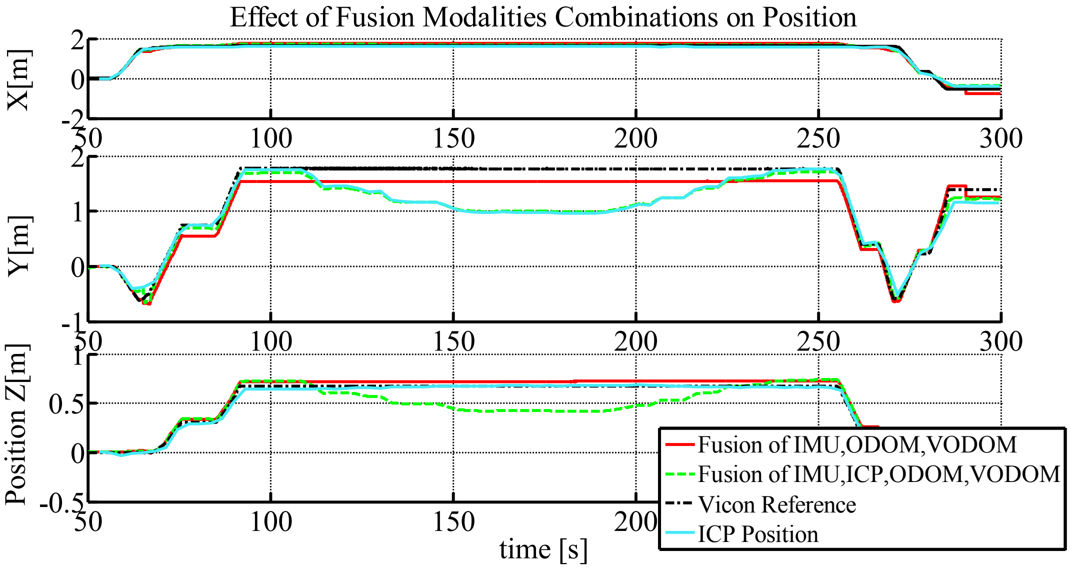


Figure 5.25. Effect of the ICP faulty position measurement propagating into the fusion algorithm. Note the ICP position measurement affecting the position estimate for the complete fusion scheme. The error in the Z coordinate is caused by applying the nonholomic constraint: the robot *moved* in parallel with its O frame, which was not leveled at that moment.

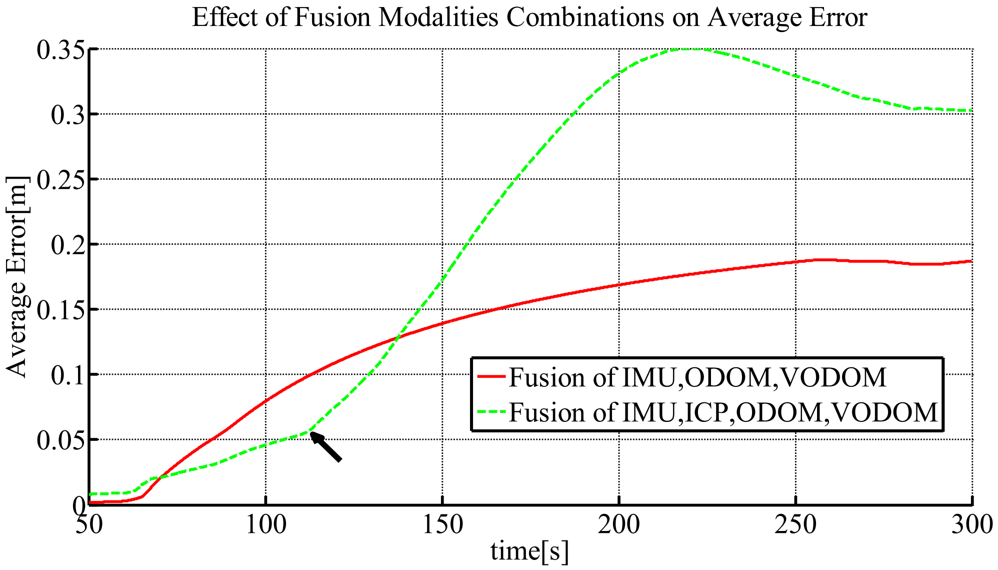


Figure 5.26. The ICP faulty position measurement affecting the average position error development. The black arrow marks the time we moved the big paper board.

5.3.5. Deformations of the ICP Map

The final fail case describes a standard ICP localization algorithm issue in some environments, when it gradually drifts (bends) while building its inner point cloud map. It is not entirely clear what features of the environment cause (or amplify) these deformations; an example is shown in the Fig. 5.27. The experiment was performed in the ASL building on two floors, the robot was navigated through two hallways making a circle over the two floors. In this case, the reference was the Leica theodolite, registering the robot's position on the upper floor (similar experiment setup to the one depicted in Fig. 3.3).

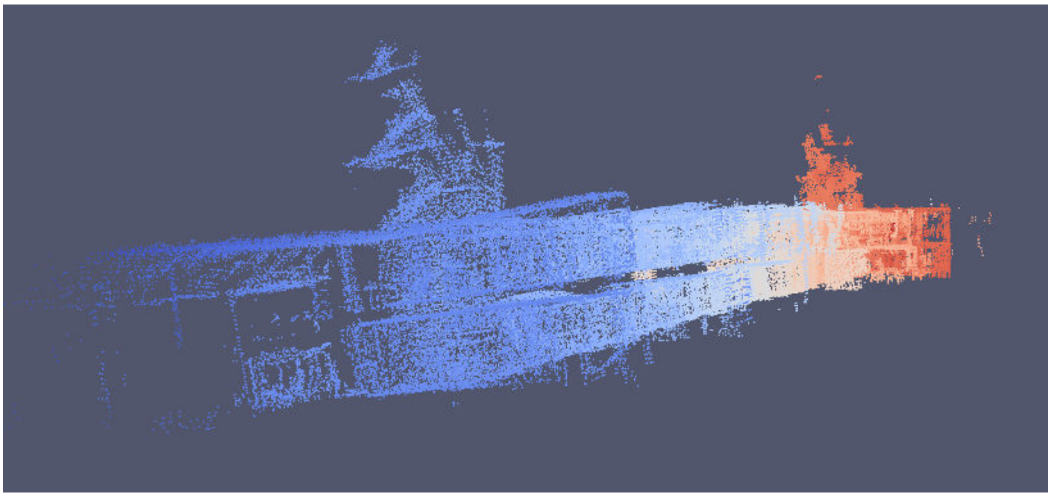


Figure 5.27. The ICP localization algorithm inner point cloud map. The twisting is apparent after coloring the points by their distance from the map origin. The structures at the ends of the corridors are two staircases. They should be parallel, but we observe a counter-clockwise twist of approximately 10 degrees.

Although the ICP position and attitude measurements error relative to the original coordinate frame grew with distance traveled from the original position, our approach to the fusion of the modalities treating them as velocities compensated for this error as it is apparent in Fig. 5.28 and 5.29. We observed this ability to straighten up the trajectory on the outdoor experiments as well. We consider this property one of the main contributions of the fusion system.

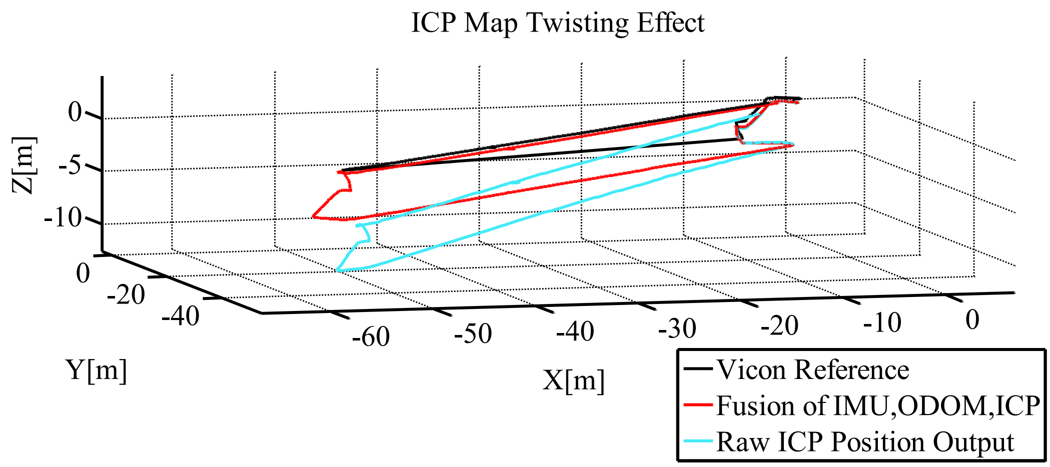


Figure 5.28. 3D plots of the trajectory of the experiment in the ASL building (ETH Zürich). It compares the bent trajectory of the original ICP localization (blue), the output of the fusion system (ref) and the Leica reference position (black, available only on the top floor and in the further staircase)

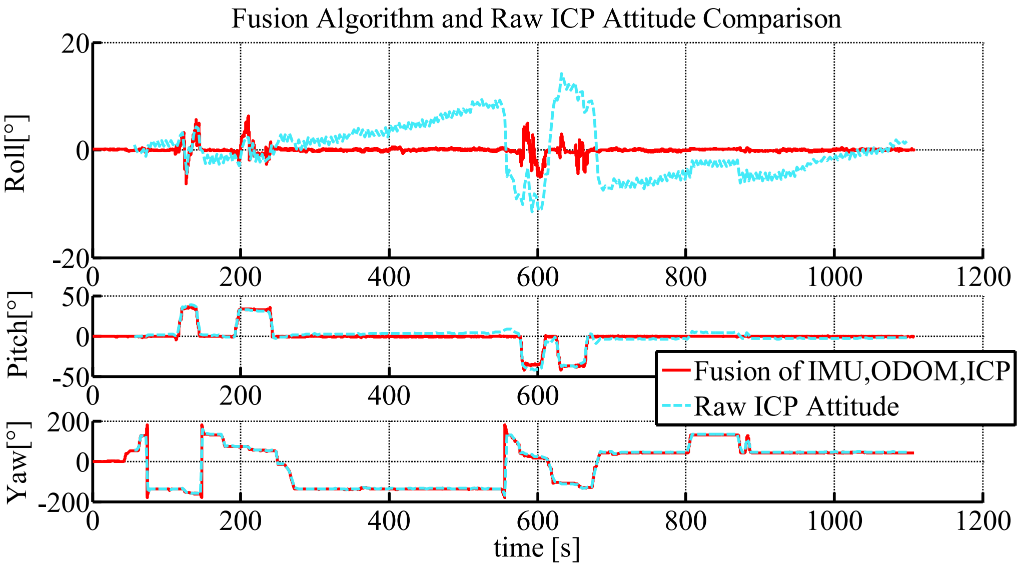


Figure 5.29. The attitude comparison, the ICP attitude shows the drift in the roll angle - an interesting fact is, that it twists back (almost) to the original value, although the robot returned through a different corridor.

6. Conclusion

In this work, we have designed a localization algorithm for a mobile Urban Search&Rescue robot NIFTi. The localization algorithm fuses several sensor modalities exploiting their strong points (high rate of the track odometry, independence of the visual odometry, local metric accuracy of the ICP odometry). This fusion is done by means of the error state EKF, treating all sensor measurements as velocities rather than absolute positions or attitudes. Such an approach deals with the drift of the world frames of the sensor modalities. While designing this algorithm, we have benefited from the previous experience in this area (see Appendix E).

Performance of the designed algorithm has been extensively tested both by indoor and outdoor experiments (over 4 kilometers traveled), comparing the algorithm output with the high-precision reference provided by the Vicon and Leica reference systems. The experiments have proven ability of the fusion algorithm to correct the track odometry errors caused by slippages, to overcome outages of the laser scanner and to correct the ICP pointcloud map deformations. On the other hand, sensitivity to erroneous attitude measurements originating both from the visual odometry and from the ICP odometry was detected. Still, given an adequate sensor modality combination, the average position error was comparable to the size of the robot; that was our threshold to consider the solution successful.

There are several issues left for the future work. The main issue to be solved is the automatic sensor modality failure detection and resolution which was well outside the scope of this thesis and according to the state of the art, it provides unresolved challenge in the whole robotic community. Apart from the failure detection and resolution, the proposed estimation of the R and O frames transformation would relieve the robot user from manually calibrating it, yet, another (unbiased) source of the velocity measurement must be provided for it to work correctly. Such a source might be the visual odometry, whose scale has been already incorporated into the system state. However, a stable scale estimation must ensured, which is another issue left for the future work.

The algorithm was implemented and tested in the MATLAB environment, nevertheless, the code was written so it would be simple to reimplement it *C++* for the Robot Operating System.

All the points of the submission of this thesis were successfully fulfilled except for a minor change; the final *C++* implementation will be realized by our colleagues at the ETH.

Bibliography

[1] W. G. Breckenridge, “Quaternions - proposed standard conventions,” tech. rep., JPL, 1999. ix, 58, 62

[2] N. Trawny and S. I. Roumeliotis, “Indirect kalman filter for 3d attitude estimation - a tutorial for quaternion algebra,” tech. rep., University of Minnesota, 2005. ix, 11, 12, 25, 59, 60

[3] P. G. Savage, “What do accelerometers measure?,” May 8, 2005 2005. x, 4, 15

[4] S. M. Weiss, “Vision based navigation for micro helicopters,” 2012. Dissertation, ETH Zurich. 3, 6, 12, 13

[5] L. Oswald, “Robust localization for search and rescue robots,” 2012. 3, 14

[6] V. Kubelka and M. Reinstein, “Complementary filtering approach to orientation estimation using inertial sensors only,” in *ICRA2012: Proceedings of 2012 IEEE International Conference on Robotics and Automation* (L. Parker, ed.), (Madison, USA), pp. 599–605, Omnipress; IEEE Robotics and Automation Society, May 2012. 4, 49

[7] D. Endo, Y. Okada, K. Nagatani, and K. Yoshida, “Path following control for tracked vehicles based on slip-compensating odometry,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 2871–2876, 2007. 4, 16, 17, 18

[8] M. Reinstein, V. Kubelka, and K. Zimmermann, “Terrain adaptive odometry for mobile skid-steer robots,” in *ICRA2013: Proceedings of 2013 IEEE International Conference on Robotics and Automation*, (Karlsruhe, Germany), pp. 4691–4696, IEEE, May 2013. 4, 18

[9] D. Scaramuzza and F. Fraundorfer, “Visual odometry: Part i - the first 30 years and fundamentals,” *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011. 4, 30

[10] F. Fraundorfer and D. Scaramuzza, “Visual odometry : Part ii: Matching, robustness, optimization, and applications,” *Robotics & Automation Magazine, IEEE*, vol. 19, no. 2, pp. 78–90, 2012. 4, 30

[11] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, “Comparing icp variants on real-world data sets,” *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013. 5, 22

[12] A. Nemra and N. Aouf, “Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering,” *Sensors Journal, IEEE*, vol. 10, no. 4, pp. 789–798, 2010. 11

[13] E.-H. Shin, “Estimation techniques for low-cost inertial navigation.” Dissertation, May 2005. 12

[14] K. R. Britting, *Inertial navigation systems analysis.* New York: Wiley-Interscience, 1971. 12

[15] C. V. Loan, "Computing integrals involving the matrix exponential," *Automatic Control, IEEE Transactions on*, vol. 23, no. 3, pp. 395–404, 1978. 14

[16] K. Nagatani, N. Tokunaga, Y. Okada, and K. Yoshida, "Continuous acquisition of three-dimensional environment information for tracked vehicles on uneven terrain," in *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*, pp. 25–30, 2008. 16, 17, 18

[17] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of imu and vision for absolute scale estimation in monocular slam," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 287–299, 2011. 30

[18] B. A. McElhoe, "An assessment of the navigation and course corrections for a manned flyby of mars or venus," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-2, no. 4, pp. 613–623, 1966. 32

[19] G. L. Smith, S. F. Schmidt, and L. A. McGee, "Optimal filtering and linear prediction applied to a midcource navigation system for the circumlunar mission," tech. rep., U.S. Government Printing Office, 1962. 32

[20] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960. 32

[21] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB.* New Jersey: Wiley, 3rd ed., 2008. 32

[22] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, pp. 56–79, January 01 2011. 32

[23] S. J. Julier and J. K. Uhlmann, "A new extension of the kalman filter to nonlinear systems," in *Proc. AeroSense: 11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, pp. 182–193, 1997. 32

[24] S. I. Roumeliotis, "A kalman filter for processing 3-d relative pose measurements," tech. rep., California Institute of Technology, 2002. 58

A. Linear Error Model Equations Derivation

This appendix goes through the error model derivation in detail. Note that we leave out coordinate frame indices in derivations for better legibility.

A.1. Position Error Differential Equation

Starting with the first nonlinear equation (2.2)

$$\dot{\mathbf{p}}_N = C^T_{(\mathbf{q}^R_N)} \mathbf{v}_R$$

we remind that the error quaternion has been defined as (2.9)

$$\delta\mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1}$$

therefore, the true value of the attitude quaternion can be expressed as a multiplication of the best estimate and the error quaternion

$$\mathbf{q} = \delta\mathbf{q} \otimes \hat{\mathbf{q}} \tag{A.1}$$

Following the the idea of an error state $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$, the position is composed of the best estimate and the error position

$$\mathbf{p} = \hat{\mathbf{p}} + \Delta\mathbf{p} \tag{A.2}$$

The quaternion $\mathbf{q}$ expresses rotation from the N frame to the R frame, the position $\mathbf{p}$ is expressed in the N frame and the velocity $\mathbf{v}$ is expressed in the R frame. If we express the time derivative of (A.2), we can use the nonlinear expression (2.2)

$$\dot{\mathbf{p}} = \dot{\hat{\mathbf{p}}} + \Delta\dot{\mathbf{p}} \tag{A.3}$$

$$\Delta\dot{\mathbf{p}} = \dot{\mathbf{p}} - \dot{\hat{\mathbf{p}}} \tag{A.4}$$

$$\Delta\dot{\mathbf{p}} = C^T_{(\mathbf{q})}\mathbf{v} - C^T_{(\hat{\mathbf{q}})}\hat{\mathbf{v}} \tag{A.5}$$

After substitution from (A.1) and similarly to (A.2) but with velocities, we get

$$\Delta\dot{\mathbf{p}} = C^T_{(\delta\mathbf{q}\otimes\hat{\mathbf{q}})}(\hat{\mathbf{v}} + \Delta\mathbf{v}) - C^T_{(\hat{\mathbf{q}})}\hat{\mathbf{v}} \tag{A.6}$$

Now, since we follow the JPL quaternion multiplication rule proposal [1], the rotation can be split respecting the order of the quaternion multiplication

$$\Delta\dot{\mathbf{p}} = (C_{(\delta\mathbf{q})}C_{(\hat{\mathbf{q}})})^T(\hat{\mathbf{v}} + \Delta\mathbf{v}) - C^T_{(\hat{\mathbf{q}})}\hat{\mathbf{v}} \tag{A.7}$$

$$\Delta\dot{\mathbf{p}} = C^T_{(\hat{\mathbf{q}})}C^T_{(\delta\mathbf{q})}(\hat{\mathbf{v}} + \Delta\mathbf{v}) - C^T_{(\hat{\mathbf{q}})}\hat{\mathbf{v}} \tag{A.8}$$

After expanding the equation, we get

$$\Delta\dot{\mathbf{p}} = C^T_{(\hat{\mathbf{q}})}(C^T_{(\delta\mathbf{q})} - I)\hat{\mathbf{v}} + C^T_{(\hat{\mathbf{q}})}C^T_{(\delta\mathbf{q})}\Delta\mathbf{v} \tag{A.9}$$

where I is a 3×3 identity matrix. We simplify this equation following approach similar to [24], namely equations (6-12). The idea is approximating the $(C^T_{(\delta\mathbf{q})} - I)$ term by vector multiplication $\lfloor \delta\theta \rfloor$, where the $\delta\theta$ is the rotation vector defined in (2.11). The quaternion $\delta\mathbf{q} = [x, y, z, w]^T$ represents a small error rotation, thus, x, y and z are close to zero and w is almost 1 [2, eq. 161]. Therefore,

$$C^T_{(\delta\mathbf{q})} = \begin{bmatrix} 1-2y^2-2z^2 & 2(xy+zw) & 2(xz-yw) \\ 2(xy-zw) & 1-2x^2-2z^2 & 2(yz+xw) \\ 2(xz+yw) & 2(yz-xw) & 1-2x^2-2y^2 \end{bmatrix}^T \tag{A.10}$$

can by approximated as

$$C^T_{(\delta\mathbf{q})} \approx \begin{bmatrix} 1 & 2z & -2y \\ -2z & 1 & 2x \\ 2y & -2x & 1 \end{bmatrix}^T \tag{A.11}$$

neglecting all xy, yz, zx, x^2, y^2 and z^2 terms and substituting $w = 1$. Then,

$$C^T_{(\delta\mathbf{q})} - I \approx \begin{bmatrix} 0 & -2z & 2y \\ 2z & 0 & -2x \\ -2y & 2x & 0 \end{bmatrix} \tag{A.12}$$

is actually a vector multiplication expressed in the matrix multiplication form

$$\begin{bmatrix} 0 & -2z & 2y \\ 2z & 0 & -2x \\ -2y & 2x & 0 \end{bmatrix} = 2 \left\lfloor \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right\rfloor \tag{A.13}$$

Since the $[x, y, z]^T$ term is accordingly to (2.10) and (2.11) the vector part of the quaternion - which is equal to $\frac{1}{2}\delta\theta$ - we conclude that

$$C^T_{(\delta\mathbf{q})} - I \approx 2\lfloor \vec{\delta q} \rfloor = \lfloor \delta\theta \rfloor \tag{A.14}$$

Using this approximation, we can rewrite (A.9) to

$$\Delta\dot{\mathbf{p}} \approx C^T_{(\hat{\mathbf{q}})} \lfloor \delta\theta \rfloor \hat{\mathbf{v}} + C^T_{(\hat{\mathbf{q}})} C^T_{(\delta\mathbf{q})} \Delta\mathbf{v} \tag{A.15}$$

which is equivalent to

$$\Delta\dot{\mathbf{p}} \approx -C^T_{(\hat{\mathbf{q}})} \lfloor \hat{\mathbf{v}} \rfloor \delta\theta + C^T_{(\hat{\mathbf{q}})} C^T_{(\delta\mathbf{q})} \Delta\mathbf{v} \tag{A.16}$$

only reversing the vector multiplication order ($x \times y = -y \times x$). If we consider $C^T_{(\delta\mathbf{q})} \approx I$ since the very small error rotation, we obtain the final form of the position error differential equation

$$\Delta\dot{\mathbf{p}} \approx -C^T_{(\hat{\mathbf{q}})} \lfloor \hat{\mathbf{v}} \rfloor \delta\theta + C^T_{(\hat{\mathbf{q}})} \Delta\mathbf{v} \tag{A.17}$$

which is suitable for the EKF since partial derivatives are easy to express from this equation.

A.2. Attitude Error Differential Equation

Expressing attitude error differential equation from the non-linear form:

$$\dot{\mathbf{q}} = \frac{1}{2}\Omega(\omega)\mathbf{q} \tag{A.18}$$

closely follows approach in [2, eq. 165-177]. The derivation starts with the error quaternion definition:

$$\mathbf{q} = \delta\mathbf{q} \otimes \hat{\mathbf{q}} \qquad \Bigg| \frac{\mathrm{d}}{\mathrm{d}t} \tag{A.19}$$

$$\dot{\mathbf{q}} = \dot{\delta\mathbf{q}} \otimes \hat{\mathbf{q}} + \delta\mathbf{q} \otimes \dot{\hat{\mathbf{q}}} \tag{A.20}$$

and since we are interested in $\dot{\delta\mathbf{q}}$, we substitute for $\dot{\mathbf{q}}$ and $\dot{\hat{\mathbf{q}}}$ from (A.18):

$$\frac{1}{2}\begin{bmatrix}\omega\\0\end{bmatrix} \otimes \mathbf{q} = \dot{\delta\mathbf{q}} \otimes \hat{\mathbf{q}} + \delta\mathbf{q} \otimes \left(\frac{1}{2}\begin{bmatrix}\hat{\omega}\\0\end{bmatrix} \otimes \hat{\mathbf{q}}\right) \qquad \Bigg| -\frac{1}{2}\left(\delta\mathbf{q} \otimes \begin{bmatrix}\hat{\omega}\\0\end{bmatrix} \otimes \hat{\mathbf{q}}\right) \tag{A.21}$$

$$\dot{\delta\mathbf{q}} \otimes \hat{\mathbf{q}} = \frac{1}{2}\left(\begin{bmatrix}\omega\\0\end{bmatrix} \otimes \mathbf{q} - \delta\mathbf{q} \otimes \begin{bmatrix}\hat{\omega}\\0\end{bmatrix} \otimes \hat{\mathbf{q}}\right) \qquad \Big| \otimes \hat{\mathbf{q}}^{-1} \tag{A.22}$$

$$\dot{\delta\mathbf{q}} = \frac{1}{2}\left(\begin{bmatrix}\omega\\0\end{bmatrix} \otimes \delta\mathbf{q} - \delta\mathbf{q} \otimes \begin{bmatrix}\hat{\omega}\\0\end{bmatrix}\right) \tag{A.23}$$

where $\hat{\omega} = \omega - \Delta\omega$ according to the error state definition. If we substitute for ω, we get

$$\begin{aligned}
\dot{\delta\mathbf{q}} &= \frac{1}{2}\left(\begin{bmatrix}\hat{\omega}+\Delta\omega\\0\end{bmatrix} \otimes \delta\mathbf{q} - \delta\mathbf{q} \otimes \begin{bmatrix}\hat{\omega}\\0\end{bmatrix}\right) && \text{(A.24)}\\
&= \frac{1}{2}\left(\begin{bmatrix}\hat{\omega}\\0\end{bmatrix} \otimes \delta\mathbf{q} - \delta\mathbf{q} \otimes \begin{bmatrix}\hat{\omega}\\0\end{bmatrix}\right) + \frac{1}{2}\begin{bmatrix}\Delta\omega\\0\end{bmatrix} \otimes \delta\mathbf{q} && \text{(A.25)}\\
&= \frac{1}{2}\left(\begin{bmatrix}-\lfloor\hat{\omega}\rfloor & \hat{\omega}\\ -\hat{\omega}^T & 0\end{bmatrix} \cdot \delta\mathbf{q} - \begin{bmatrix}+\lfloor\hat{\omega}\rfloor & \hat{\omega}\\ -\hat{\omega}^T & 0\end{bmatrix} \cdot \delta\mathbf{q}\right) + \frac{1}{2}\begin{bmatrix}\Delta\omega\\0\end{bmatrix} \otimes \delta\mathbf{q} && \text{(A.26)}\\
&= \frac{1}{2}\begin{bmatrix}-2\lfloor\hat{\omega}\rfloor & \emptyset_{3\times1}\\ \emptyset_{1\times3} & 0\end{bmatrix} \cdot \delta\mathbf{q} + \frac{1}{2}\begin{bmatrix}\Delta\omega\\0\end{bmatrix} \otimes \delta\mathbf{q} && \text{(A.27)}\\
&\approx \frac{1}{2}\begin{bmatrix}-2\lfloor\hat{\omega}\rfloor & \emptyset_{3\times1}\\ \emptyset_{1\times3} & 0\end{bmatrix} \cdot \delta\mathbf{q} + \frac{1}{2}\begin{bmatrix}-\lfloor\Delta\omega\rfloor & \Delta\omega\\ -\Delta\omega^T & 0\end{bmatrix} \cdot \begin{bmatrix}\vec{\delta q}\\1\end{bmatrix} && \text{(A.28)}\\
&= \frac{1}{2}\begin{bmatrix}-2\lfloor\hat{\omega}\rfloor & \emptyset_{3\times1}\\ \emptyset_{1\times3} & 0\end{bmatrix} \cdot \delta\mathbf{q} + \frac{1}{2}\begin{bmatrix}\Delta\omega\\0\end{bmatrix} + O(|\Delta\omega||\delta\mathbf{q}|) && \text{(A.29)}
\end{aligned}$$

Neglecting the second order terms, we can write

$$\dot{\delta\mathbf{q}} = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\vec{\delta q}\\q_4\end{bmatrix} \approx \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\frac{1}{2}\delta\theta\\1\end{bmatrix} \approx \begin{bmatrix}-\lfloor\hat{\omega}\rfloor\vec{\delta q} + \frac{1}{2}\Delta\omega\\0\end{bmatrix} \approx \begin{bmatrix}-\frac{1}{2}\lfloor\hat{\omega}\rfloor\delta\theta + \frac{1}{2}\Delta\omega\\0\end{bmatrix} \tag{A.30}$$

and by expressing the time derivative of $\delta\theta$, we obtain

$$\dot{\delta\theta} \approx -\lfloor\hat{\omega}\rfloor\delta\theta + \Delta\omega \tag{A.31}$$

which corresponds to (2.13).

A.3. Velocity Error Differential Equation

Velocity is described by differential equation (2.4):

$$\dot{\mathbf{v}}_R = \mathbf{f}_R - C_{(\mathbf{q}_N^R)}\mathbf{g}_N + \lfloor\mathbf{v}_R\rfloor\omega_R$$

which will be used to express the velocity error differential equation. The same way we did in (A.4), we express the time derivative of $\Delta\mathbf{v}$ as

$$\Delta\dot{\mathbf{v}} = \dot{\mathbf{v}} - \dot{\hat{\mathbf{v}}} \tag{A.32}$$

and substitute from (2.4) knowing that $\mathbf{v} = \hat{\mathbf{v}} + \Delta\mathbf{v}$ and $\mathbf{f} = \hat{\mathbf{f}} + \Delta\mathbf{f}$. We obtain

$$\begin{aligned} \Delta\dot{\mathbf{v}} =& (\hat{\mathbf{f}} + \Delta\mathbf{f}) - C_{(\delta\mathbf{q}\otimes\hat{\mathbf{q}})}\mathbf{g} + \lfloor\hat{\mathbf{v}} + \Delta\mathbf{v}\rfloor(\hat{\omega} + \Delta\omega) \\ &- \hat{\mathbf{f}} + C_{(\hat{\mathbf{q}})}\mathbf{g} - \lfloor\hat{\mathbf{v}}\rfloor\hat{\omega} \end{aligned} \tag{A.33}$$

We split the right part of the equation into corresponding term pairs

$$\begin{aligned} &\Delta\dot{\mathbf{v}} = \\ &\quad ((\hat{\mathbf{f}} + \Delta\mathbf{f}) - \hat{\mathbf{f}}) && \}\star \\ &\quad + (-C_{(\delta\mathbf{q}\otimes\hat{\mathbf{q}})}\mathbf{g} + C_{(\hat{\mathbf{q}})}\mathbf{g}) && \}\dagger \\ &\quad + (\lfloor\hat{\mathbf{v}} + \Delta\mathbf{v}\rfloor(\hat{\omega} + \Delta\omega) - \lfloor\hat{\mathbf{v}}\rfloor\hat{\omega}) && \}\diamond \end{aligned} \tag{A.34}$$

and simplify them separately to keep clarity. The first two terms ($\star$) can be simplified into

$$(\hat{\mathbf{f}} + \Delta\mathbf{f}) - \hat{\mathbf{f}} = \Delta\mathbf{f} \tag{A.35}$$

The second part ($\dagger$) is simplified similarly to the approach in sub-section A.1

$$\begin{aligned} -C_{(\delta\mathbf{q}\otimes\hat{\mathbf{q}})}\mathbf{g} + C_{(\hat{\mathbf{q}})}\mathbf{g} &= (-C_{(\delta\mathbf{q})} + I)C_{(\hat{\mathbf{q}})}\mathbf{g} \\ &\approx \lfloor\delta\theta\rfloor C_{(\hat{\mathbf{q}})}\mathbf{g} \\ &= -\lfloor C_{(\hat{\mathbf{q}})}\mathbf{g}\rfloor\delta\theta \end{aligned} \tag{A.36}$$

and finally the third part ($\diamond$)

$$\begin{aligned} \lfloor\hat{\mathbf{v}} + \Delta\mathbf{v}\rfloor(\hat{\omega} + \Delta\omega) - \lfloor\hat{\mathbf{v}}\rfloor\hat{\omega} &= (\lfloor\hat{\mathbf{v}}\rfloor + \lfloor\Delta\mathbf{v}\rfloor)(\hat{\omega} + \Delta\omega) - \lfloor\hat{\mathbf{v}}\rfloor\hat{\omega} \\ &= \lfloor\hat{\mathbf{v}}\rfloor\hat{\omega} + \lfloor\hat{\mathbf{v}}\rfloor\Delta\omega + \lfloor\Delta\mathbf{v}\rfloor\hat{\omega} + \lfloor\Delta\mathbf{v}\rfloor\Delta\omega - \lfloor\hat{\mathbf{v}}\rfloor\hat{\omega} \\ &= \lfloor\hat{\mathbf{v}}\rfloor\Delta\omega + \lfloor\Delta\mathbf{v}\rfloor\hat{\omega} + \lfloor\Delta\mathbf{v}\rfloor\Delta\omega \\ &= \lfloor\hat{\mathbf{v}}\rfloor\Delta\omega - \lfloor\hat{\omega}\rfloor\Delta\mathbf{v} + \lfloor\Delta\mathbf{v}\rfloor\Delta\omega \\ &\approx \lfloor\hat{\mathbf{v}}\rfloor\Delta\omega - \lfloor\hat{\omega}\rfloor\Delta\mathbf{v} \end{aligned} \tag{A.37}$$

where in the last step the product of two error terms is neglected. Finally, by summing the three simplified parts, we obtain the velocity differential equation

$$\Delta\dot{\mathbf{v}} \approx \Delta\mathbf{f} - \lfloor C_{(\hat{\mathbf{q}})}\mathbf{g}\rfloor\delta\theta + \lfloor\hat{\mathbf{v}}\rfloor\Delta\omega - \lfloor\hat{\omega}\rfloor\Delta\mathbf{v} \tag{A.38}$$

B. Quaternion Multiplication and Inversion Definition

The quaternion product is defined according to [1] as

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_4p_1 + q_3p_2 - q_2p_3 + p_1p_4 \\ -q_3p_1 + q_4p_2 + q_1p_3 + q_2p_4 \\ q_2p_1 - q_1p_2 + q_4p_3 + q_3p_4 \\ -q_1p_1 - q_2p_2 - q_3p_3 + q_4p_4 \end{bmatrix} \tag{B.1}$$

The quaternion inversion is defined according to [1] as

$$\mathbf{q}^{-1} = \begin{bmatrix} -q_1 \\ -q_2 \\ -q_3 \\ q_4 \end{bmatrix} \tag{B.2}$$

while it is left without proof that

$$\mathbf{q}^{-1} \otimes \mathbf{q} = \mathbf{q} \otimes \mathbf{q}^{-1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{q}_{\text{identity}} \tag{B.3}$$

C. List of Experiments

The Experiments: X.Z.(a,b,c)

- X:
  - **1**: The **Structure A**
  - **2**: The **Pile**
  - **3**: No structure, just the **Vicon room**
  - **3**: The **Leica in the Hallway**
  - **4**: The **Building** without reference
  - **5**: **Outside** without reference
- Z: **Experiment serial number**
- (a,b,c...): **Optional repetition**

Re-Running the 20130307

Experiment ID	0.1
Description	*Big squares*
Conditions tested	#General performance
Expected time	8 min
Start / End [hh:mm]	16:17/16:20 not repeated
Notes	1st round 16:18, 2nd 16:19

Experiment ID	0.2
Description	*Big squares opposite direction*
Conditions tested	#General performance
Expected time	8 min
Start / End [hh:mm]	16:31/16:33 try again: 20130325_17:05/17:09
Notes	first time no data stored

Experiment ID	0.3
Description	*Random path I*
Conditions tested	#General performance
Expected time	8 min
Start / End [hh:mm]	16:34/16:37 try again: 20130325_17:10/17:13
Notes	including backward path from 16:35 and spot turn 16:36

Experiment ID	0.4
Description	*Random path I - opposite direction*
Conditions tested	#General performance
Expected time	8 min
Start / End [hh:mm]	20130325_17:14/17:18
Notes	

Experiment ID	0.5
Description	*Backlash/Tilt of Platform*
Conditions tested	#General performance
Expected time	2 min
Start / End [hh:mm]	20130325_17:23/17:27
Notes	

The Structure A Experiments

Experiment ID	1.1.a
Description	Up the non-slippery slope, cross the structure, turn around, down to the carpet, repeat 5 times
Conditions tested	#General performance
Expected time	8 min
Start / End [hh:mm]	20130326_12:12/12:21
Notes	Condition #11

Experiment ID	1.1.b
Description	Up the non-slippery slope, cross the structure, turn around, down to the carpet, repeat 5 times
Conditions tested	#General performance
Expected time	8 min
Start / End [hh:mm]	20130326_12:22/12:30
Notes	Condition #11

Experiment ID	1.2.a
Description	Up the non-slippery slope, cross the structure, turn around, down to the carpet, repeat 5 times, **driving backwards**
Conditions tested	#General performance
Expected time	8 min
Start / End [hh:mm]	20130326_12:33/12:40
Notes	Condition #11

Experiment ID	1.3.a
Description	Up the non-slippery slope, cross the structure, turn around, down to the carpet, repeat 3 times, **halogen light on - stationary**
Conditions tested	#8,9
Expected time	10 min
Start / End [hh:mm]	20130326_14:07/14:10
Notes	Condition #11

Experiment ID	1.3.b
Description	Up the non-slippery slope, cross the structure, turn around, down to the carpet, repeat 3 times, **halogen light on - stationary**
Conditions tested	#8,9
Expected time	10 min
Start / End [hh:mm]	20130326_14:15/14:18
Notes	Condition #11

Experiment ID	1.3.c
Description	Up the non-slippery slope, cross the structure, turn around, down to the carpet, repeat 3 times, **halogen light on - moving**
Conditions tested	#8,9
Expected time	10 min
Start / End [hh:mm]	20130326_14:24/14:28
Notes	Condition #11

Experiment ID	1.3.d
Description	Up the non-slippery slope, cross the structure, turn around, down to the carpet, repeat 3 times, **halogen light on - moving**
Conditions tested	#8,9
Expected time	10 min
Start / End [hh:mm]	20130326_14:30/14:33
Notes	Condition #11

Experiment ID	1.3.e
Description	Up the non-slippery slope, cross the structure, turn around, down to the carpet, repeat 3 times, **halogen light off - control experiment**
Conditions tested	#8,9
Expected time	10 min
Start / End [hh:mm]	20130326_14:47/14:50
Notes	Condition #11

Experiment ID	1.3.f
Description	Up the non-slippery slope, cross the structure, turn around, down to the carpet, repeat 3 times, **halogen light on (stationary) - main lights off**
Conditions tested	#8,9
Expected time	10 min
Start / End [hh:mm]	20130326_14:53/14:56
Notes	Condition #11

Experiment ID	1.4.a
Description	Up the structure, cross it, down to the carpet, around, up again and so on, twice, **reduced laser radius & moving obstacle**
Conditions tested	#4,10
Expected time	10 min
Start / End [hh:mm]	20130326_16:32/16:36
Notes	Condition #11, **Moving artificial wall within the laser range,** laser range reduced to 3.5m

Experiment ID	1.4.b
Description	Up the structure, cross it, down to the carpet, around, up again and so on, once, **reduced laser radius & moving obstacle**
Conditions tested	#4,10
Expected time	10 min
Start / End [hh:mm]	20130326_16:38/16:41
Notes	Condition #11, **Moving artificial wall within the laser range,** laser range reduced to 3.5m

Experiment ID	1.4.c
Description	Up the structure, cross it, down to the carpet, around, up again and so on, 5 times around, **reduced laser radius & moving obstacle**
Conditions tested	#4,10
Expected time	10 min
Start / End [hh:mm]	20130326_16:44/16:50
Notes	Condition #11, **Moving artificial wall within the laser range,** laser range reduced to 3.5m

Experiment ID	1.4.d bad data
Description	Up the structure, cross it, down to the carpet, around, up again and so on, 5 times around, **reduced laser radius**
Conditions tested	#4,10
Expected time	10 min
Start / End [hh:mm]	20130326_17:00/17:06 csv_files: 17:32:15
Notes	Condition #11, laser range reduced to 3m

Experiment ID	1.4.e bad data
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **reduced laser radius**
Conditions tested	#4,10, 1
Expected time	10 min
Start / End [hh:mm]	20130326_17:07/17:12 csv_files: 17:42:42
Notes	Condition #11, loop closed 3x, laser range reduced to 3m

Experiment ID	1.4.f bad data
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **reduced laser radius - opposite direction** (up the slippery slope first)
Conditions tested	#4,10, 1
Expected time	10 min
Start / End [hh:mm]	20130326_17:13/17:18 csv_files: 17:47:08
Notes	Condition #11, loop closed 3x, laser range reduced to 3m

Experiment ID	1.4.d second try
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **reduced laser radius**
Conditions tested	#4,10, 1
Expected time	10 min
Start / End [hh:mm]	20130327_11:06/11:10
Notes	Condition #11, laser range reduced to 3m

Experiment ID	1.4.e second try
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **reduced laser radius**
Conditions tested	#4,10, 1
Expected time	10 min
Start / End [hh:mm]	20130327_11:11/11:17
Notes	Condition #11, loop closed 3x, laser range reduced to 3m

Experiment ID	1.4.f second try
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **reduced laser radius - opposite direction** (up the slippery slope first)
Conditions tested	#4,10, 1
Expected time	10 min
Start / End [hh:mm]	20130327_11:17/11:22
Notes	Condition #11, loop closed 3x, laser range reduced to 3m

Experiment ID	1.4.g
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **reduced laser radius - control experiment**
Conditions tested	#4,10, 1
Expected time	10 min
Start / End [hh:mm]	20130327_11:25/11:29
Notes	Condition #11, loop closed 3x, laser range is no longer reduced

Experiment ID	1.6.a
Description	Up the structure, cross it, down to the carpet, around, up again and so on, 5 times around, **blindfold some camera lenses**
Conditions tested	#11
Expected time	10 min
Start / End [hh:mm]	20130326_16:02/16:09
Notes	visual odom died at the beginning, 16:09 laser died

Experiment ID	1.6.b
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **blindfold some camera lenses**
Conditions tested	#11, 1
Expected time	10 min
Start / End [hh:mm]	20130327_11:30/11:33
Notes	11:33 laser died

Experiment ID	1.6.c
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **blindfold some camera lenses**
Conditions tested	#11, 1
Expected time	10 min
Start / End [hh:mm]	20130327_11:37/11:41
Notes	visual odom died

Experiment ID	1.7.a
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **pause the laser scanning**
Conditions tested	#3,14,1
Expected time	10 min
Start / End [hh:mm]	20130327_11:45/11:50
Notes	

Experiment ID	1.7.b
Description	Up the structure, cross it, down the slippery slope, close the loop, up again and so on, 3 times around, **pause the laser scanning - opposite direction** (up the slippery slope first)
Conditions tested	#3,14, 1
Expected time	10 min
Start / End [hh:mm]	20130327_11:52/11:56
Notes	

Experiment ID	1.8.a
Description	Go forward and hit the obstacle by one track, simulating the commo autonomous navigation scenario, go back & hit again, repeat 5x
Conditions tested	#13, 1
Expected time	5 min
Start / End [hh:mm]	20130327_11:58/11:12:01
Notes	

Experiment ID	1.8.b
Description	Go forward and hit the obstacle by one track, simulating the commo autonomous navigation scenario, go back & hit again, repeat 5x
Conditions tested	#13, 1
Expected time	5 min
Start / End [hh:mm]	20130327_12:02/12:05
Notes	

Experiment ID	1.9.a
Description	Around the the whole structure, including the slippery surface, 15mi around, changing the speed occasionally
Conditions tested	#1,2
Expected time	8 min
Start / End [hh:mm]	20130327_12:09/12:24
Notes	visual odom died, not complete bag file due to limited hard disk space

Experiment ID	1.9.b
Description	Around the the whole structure, including the slippery surface, 15mi around, changing the speed occasionally **- opposite direction**
Conditions tested	#1,2
Expected time	8 min
Start / End [hh:mm]	20130327_12:43/12:59
Notes	laser died at the end (12:59), visual odom died

Experiment ID	1.10.a
Description	Go up the slippery slope, make the robot slip intentionally, when it's back down, go around to make the odometry work, then try to climb up again and make it slip again.
Conditions tested	#1,2
Expected time	5 min
Start / End [hh:mm]	
Notes	20130327_14:18/14:21

Experiment ID	1.10.b
Description	Go up the slippery slope, make the robot slip intentionally, when it's back down, go around to make the odometry work, then try to climb up again and make it slip again.
Conditions tested	#1,2
Expected time	5 min
Start / End [hh:mm]	20130327_14:21/14:25
Notes	

Experiment ID	1.10.c
Description	Go up the slippery slope, make the robot slip intentionally, when it’s back down, go around to make the odometry work, then try to climb up again and make it slip again.
Conditions tested	#1,2
Expected time	5 min
Start / End [hh:mm]	20130327_14:26/14:29
Notes	

Experiment ID	2.1.a
Description	Go up the pile, down the pile, once
Conditions tested	# 1,2,...?
Expected time	7 min
Start / End [hh:mm]	03/28: 11:17 / 11
Notes	Using flippers

Experiment ID	2.1.b
Description	Go up the pile, down the pile, once
Conditions tested	# 1,2
Expected time	7 min
Start / End [hh:mm]	11:25
Notes	Flippers, video recorded

Experiment ID	2.2.a
Description	Go up the pile, turn around, down the pile, once
Conditions tested	# 1,2
Expected time	7 min
Start / End [hh:mm]	11:30/
Notes	Video recorded

Experiment ID	2.2.b
Description	Go up the pile, turn around, down the pile, once
Conditions tested	# 1,2
Expected time	7 min
Start / End [hh:mm]	11:36 / 11:40
Notes	

Experiment ID	2.3.a
Description	Go up the pile, turn around, down the pile, once, **net all over the pile**
Conditions tested	# 1,2
Expected time	8 min
Start / End [hh:mm]	11:47 11:52
Notes	Video recorded

Experiment ID	2.3.b
Description	Go up the pile, turn around, down the pile, once, **net all over the pile**
Conditions tested	# 1,2
Expected time	8 min
Start / End [hh:mm]	11:53 11:57
Notes	

Experiment ID	2.4.a
Description	Go up the pile, turn around, down the pile, once, **net all over the pile, reduced laser range**
Conditions tested	# 1,2,4,10
Expected time	8 min
Start / End [hh:mm]	12:03 / 12:08
Notes	Partially recorded on the video

Experiment ID	2.4.b
Description	Go up the pile, turn around, down the pile, once, **net all over the pile, reduced laser range**
Conditions tested	# 1,2,4,10
Expected time	8 min
Start / End [hh:mm]	12:08 / 12:12
Notes	

Experiment ID	2.5.a
Description	Perform the non-zero roll movement experiment somehow, **reduced laser range**
Conditions tested	#1,2,4,10 General performance
Expected time	5 min
Start / End [hh:mm]	12:13/12:19
Notes	

Experiment ID	2.5.b
Description	Perform the non-zero roll movement experiment somehow
Conditions tested	#1,2, General performance
Expected time	5 min
Start / End [hh:mm]	12:20/12:25
Notes	

Experiment ID	2.5.c
Description	Perform the non-zero roll movement experiment somehow
Conditions tested	#1,2, General performance
Expected time	5 min
Start / End [hh:mm]	12:31/12:34
Notes	Just wood cubes, one track on them, so the robot is rolled, recorded on video, the structure had gaps between the cubes, so they moved

Experiment ID	2.5.d
Description	Perform the non-zero roll movement experiment somehow
Conditions tested	#1,2, General performance
Expected time	5 min
Start / End [hh:mm]	12:37/12:39
Notes	Just wood cubes, one track on them, so the robot is rolled, recorded on video, the structure held this time

No Structure Needed

Experiment ID	3.1.a
Description	Put the robot on a trolley and pull it slowly
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	12:46/12:50
Notes	no turns, INSO constraint on

Experiment ID	3.1.b
Description	Put the robot on a trolley and pull it slowly
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	12:53/13:00
Notes	no turns, INSO constraint on

Experiment ID	3.2.a
Description	Put the robot on a trolley and pull it as fast as the robot would move by itself
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	14:43/14:44
Notes	no turns, INSO constraint on

Experiment ID	3.2.b
Description	Put the robot on a trolley and pull it as fast as the robot would move by itself
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	14:47
Notes	no turns, INSO constraint on

Experiment ID	3.3.a
Description	Put the robot on a trolley and pull it slowly
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	14:51/14:54
Notes	with turns, INSO constraint on

Experiment ID	3.3.b
Description	Put the robot on a trolley and pull it slowly
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:00/ 15:03
Notes	with turns, INSO constraint off

Experiment ID	3.4.a
Description	Put the robot on a trolley and pull it as fast as the robot would move by itself
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:05/15:06
Notes	with turns, INSO constraint off

Experiment ID	3.4.b
Description	Put the robot on a trolley and pull it as fast as the robot would move by itself
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:08/15:15:09
Notes	with turns, INSO constraint on

Experiment ID	3.4.c
Description	Put the robot on a trolley and pull it as fast as the robot would move by itself
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:11/15:13
Notes	with turns, INSO constraint on, medium velocity (look at the vicon data)

Experiment ID	3.5.a
Description	Put the robot on a trolley, go a circle
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:16/15:18
Notes	with turns, INSO constraint on, medium velocity (look at the vicon) , 2 rounds

Experiment ID	3.5.b
Description	Put the robot on a trolley, go a circle
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:19/ 15:20
Notes	with turns, INSO constraint on, fast, 2 rounds

Experiment ID	3.5.c
Description	Put the robot on a trolley, go a circle
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:25/
Notes	with turns, INSO constraint **off**, medium speed, 2 rounds

Experiment ID	3.5.d
Description	Put the robot on a trolley, go a circle
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:29/ 15:31
Notes	with turns, INSO constraint **off**, fast, 2 rounds

Experiment ID	3.6.a
Description	Put the robot on a trolley sideways, laser pointing inside, go a circle
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:32/15:35
Notes	with turns, INSO constraint **off**, medium, 2 rounds

Experiment ID	3.6.b
Description	Put the robot on a trolley sideways, laser pointing inside, go a circle
Conditions tested	# 5,6
Expected time	5 min
Start / End [hh:mm]	15:36/15:38
Notes	with turns, INSO constraint **on**, medium, 2 rounds

Experiment ID	3.7.a
Description	Robot crossing a pallet
Conditions tested	# ?
Expected time	5 min
Start / End [hh:mm]	15:43/15:45
Notes	INSO constraint **on**, on video, on a trolley - unstable

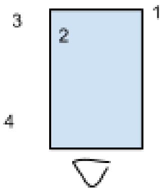
Experiment ID	3.7.b
Description	Robot crossing a pallet
Conditions tested	# ?
Expected time	5 min
Start / End [hh:mm]	15:47/15:51
Notes	INSO constraint **on**, on video, pallet on a wood brick, brick:25cm, brick+pallet = 29 cms.

Experiment ID	3.7.c
Description	Robot crossing a pallet
Conditions tested	# ?
Expected time	5 min
Start / End [hh:mm]	15:54/15:59
Notes	INSO constraint **on**, on video, 2 pallets on a wood brick, brick:25cm, brick+pallet = 29 cms.

Experiment ID	3.8.a
Description	Return home
Conditions tested	# ?
Expected time	5 min
Start / End [hh:mm]	16:01
Notes	stairs, elevators and so on...

The Leica Experiments:

The Reference Testing and Validation - Exps

Experiment ID	combined_run_starts_leveled_replayICP
Description	Basic basement hallway run
Conditions tested	Reference system function
Expected time	10-15 min
Start / End [hh:mm]	04/09 17:16
Notes 	Screws position by Leica (in the **leica coord frame**): 1: Ea.= 0.2389 , No.= 1.1628 ,Ev.=-1.2973 2: Ea.= 0.1615 , No.= 1.0202 ,Ev.= -1.2976 3: Ea.= 0.1593 , No.= 0.9842 ,Ev.= -1.3186 4: Ea.= 0.4654 , No.= 0.8136 ,Ev.= -1.3099 Perform the initialization and go straight forward (just to see that the frames are aligned. prizm: Ea.= 0.1249 , No.= 1.0051 ,Ev.= -1.1074

Experiment ID	straight_run_starts_tilted_replayICP
Description	Start tilted, repeat the previous trajectory
Conditions tested	# Reference System, Initialization, ICP function
Expected time	10 min
Start / End [hh:mm]	04/09 17:43
Notes	Same as for the 0.1: 1: Ea.= 0.6866 , No.= 2.4372 ,Ev.=-1.3111 2: Ea.= 0.6293 , No.= 2.2849 ,Ev.= -1.2910 3: Ea.= 0.6332 , No.= 2.2426 ,Ev.= -1.2992 4: Ea.= 0.9600 , No.= 2.1248 ,Ev.= -1.2543 but choose different attitude, preferably with non-zero roll and pitch.

Experiment ID	hallway
Description	Going straight through the ASL floor corridor
Conditions tested	# ICP map twisting
Expected time	10 min
Start / End [hh:mm]	04/12 15:20
Notes	Screws saved in the leica export protocol

Experiment ID	street1
Description	The street next to the CLA building, a rectangular trajectory
Conditions tested	# ICP map twisting, VODOM performance
Expected time	10 min
Start / End [hh:mm]	04/12 15:57
Notes	Screws saved in the leica export protocol

Experiment ID	street2
Description	The street next to the CLA building, a rectangular trajectory, repetition
Conditions tested	# ICP map twisting, VODOM performance
Expected time	10 min
Start / End [hh:mm]	04/12 16:13
Notes	Screws saved in the leica export protocol

Experiment ID	big_circle
Description	The park under the polyterrasse, longer loop including two stairs
Conditions tested	# ICP map, VODOM performance
Expected time	15 min
Start / End [hh:mm]	04/15 10:27
Notes	Screws saved in the leica export protocol

Experiment ID	circle
Description	The park under the polyterrasse, shorter loop
Conditions tested	# ICP map, VODOM performance
Expected time	15 min
Start / End [hh:mm]	04/15 10:09
Notes	Screws saved in the leica export protocol

Experiment ID	straight
Description	The park under the polyterrasse, forth and back on a leveled path
Conditions tested	# ICP map, VODOM performance, the rising robot issue
Expected time	5 min
Start / End [hh:mm]	04/15 09:54
Notes	Screws saved in the leica export protocol

Experiment ID	return home
Description	Return from the park to the lab, long outdoor stairs along the polymensa, crossing street, walking people...
Conditions tested	# Everything
Expected time	20 min
Start / End [hh:mm]	04/15 10:44
Notes	No reference, laser died

Experiment ID	two_floors:first run
Description	Going through two floors of the CLA building in a loop, reference on the upper floor and the closer staircase (partially)
Conditions tested	# ICP map bending
Expected time	20 min
Start / End [hh:mm]	04/18 08:42
Notes	Screws in the log file

Experiment ID	two_floors:second run
Description	Going through two floors of the CLA building in a loop, reference on the upper floor and the closer staircase (partially), opposite direction
Conditions tested	# ICP map bending
Expected time	20 min
Start / End [hh:mm]	04/18 09:02
Notes	Screws in the log file

D. Contents of the Compact Disc

The CD attached to this thesis contains:

- A pdf copy of this thesis
- The experimental dataset with photo-documentation

The final C++ implementation will be published when the NIFTi project finishes.

E. List of Previous Publications

This is a list of our previous publications related to this work:

Conference articles:

Kubelka, V.; Reinstein, M., "Complementary filtering approach to orientation estimation using inertial sensors only," Robotics and Automation (ICRA), 2012 IEEE International Conference on, May 2012

Reinstein, M.; Kubelka, V.; Zimmermann, K., "Terrain Adaptive Odometry for Mobile Skid-Steer Robots", "Robotics and Automation (ICRA), 2013 IEEE International Conference on, May 2013

Journal paper, Major revision:

Simanek, J.; Reinstein, M.; Kubelka, V., "Evaluation of the EKF-based Estimation Architectures for Data Fusion in Mobile Robots", Transactions on Mechatronics, TMECH-11-2012-2648.R1