Master thesis

Robust Multi-Hypothesis Tracker Fusing Diverse Sensor Information

Master’s degree in Automatic Control and Robotics

Author: Ely Repiso Polo

Director: Alberto Sanfeliu Cortés

September 2015
Abstract

This Master’s Thesis presents a developed tracking algorithm which fuses different object detections. These object detections are obtained from different detectors. These detectors are: a laser leg detector and a tag vision detector. The fusion has the objective to obtain more robust object detections; moreover, it has the capability to distinguish specific people. In the future, it is intend to add new detectors, such as: a skeleton person detector and a radar detector. Adding new detectors is direct and easy, since the tracker has a general and flexible implementation which allows the change of the type and number of detectors.

The Robust multi-hypothesis tracker fusing diverse sensor information uses as the backbone of the theory the multi-hypothesis tracker approach proposed by Reid [28]. Then, the algorithm has been modified to add some new improvements.

For work with social robots in the algorithm was added two functions in the probabilities of the hypothesis which allows us to control the confirmation and the deletion of the tracks. Also, the algorithm was improved by adding another function that lets we use the people’s velocities orientations to improve the association between tracks and detections in crossing situations. Furthermore, the algorithm was extended with the fusion of different detections.

For work with autonomous vehicles in the European Project Cargo ANTs\(^1\) the algorithm was adapted for tracking locally (without map) and global (with a map); was improved to use the track velocity and a specific detector to distinguish between dynamic and static objects and was adapted to group different objects of the same type using distance. The objects can have dynamic or static type.

Finally, the tests of the tracker algorithm have been carried out in the fields of social robots and autonomous vehicles (Cargo ANTs project). The experiments have been carried out with Tibi and Dabo robots, illustrate that our tracking approach can robustly and efficiently track multiple people without changes on their id’s and without losing them. Also, the experiments show that the tracker can distinguish a specific person by using the fusion; even in situations which has several people or objects that cross with or occlude the specific person. The experiments have been carried out in the Cargo ANTs project show that the tracker is able to follow only the single real dynamic object, and that the tracker is able to group different detections obtained inside the same object.

Keywords: Kalman filter, multi-hypothesis tracking, sensor fusion, autonomous vehicles, mobile and social robotics.

\(^1\)The European Project Cargo ANTs, aims to take a step further in the safe freight transportation by creating free-travelling smart vehicles - such as Automated Guided Vehicles (AGVs) and Automated Trucks (ATs) - that can co-operate in highly dynamic shared workspaces as cargo port terminals.
Acknowledgements

I would like to thank, Alberto Sanfeliu, for giving me the opportunity of work with Tibi and Dabo in the IRI, developing this big tracking project. Also, I would like to express my specials thanks to Gonzalo Ferrer, that taught me a lot of things during my beginnings at the IRI.

Regarding the writing of the Master thesis, I want to express my gratitude to all the people that help me in some manner in the English and memory structure corrections, these people are: Jose Polo, Mireya Muñoz and Andreu Corominas.

Finally, I would like to express my big gratitude towards my parents and family for their support during all these years through get here. Also, thanks to all of my different friends that supported me and brought me to distract from time to time.
List of acronyms and abbreviations

* **Tracker**: Is the tracker implemented, The Robust multi-hypothesis tracker fusing diverse sensor information.

* **Tracks**: Are the tracker objects that represent the real people followed by the tracker, each track has a specific identifier (id) to differentiate the different people in the scene.

* **Id**: Is the identifier of one person, there are numbers that the tracker uses for identify each specific person. These numbers are part of the tracks.

* **False alarm or False positive**: In the context of the project, the false alarms are detections that seems to be people, but these are not really people.

* **False negatives**: In the context of the project, false negatives happens when people are present in the scene, but fail to be detected by any of the sensors.

* **Cluster**: In the tracker this refers to sub groups of detections and tracks associated by distance.

* **Robot Operating System (ROS)**: Software used to program robots.

* **Node**: An executable program; basic single unit in ROS.

* **Publisher**: ROS node that publishes information of some type. For example: laser coordinate points.

* **Subscriber**: ROS node that gets information published by another node.

* **Rviz**: 3D Graphic visualization environment of ROS.

* **Rosbag**: File where data is stored. This data can be the outputs of the sensors of a robotic system, while we are performing an experiment. These files are used to reproduce later the same experiment in a computer simulation without using the robot to test the implemented programs.

* **Markers**: Geometric shapes represented in the Rviz such as spheres, arrows, cubes, etc. These serve to show the output variables of a node.

* **IRI**: Instituto de Robótica e Informática industrial.
* Papers: Relatively short article intended for publication in scientific journals of scientific research.

* Fusion: In our case this refers to the association of different detections to the same track. The tracker uses the different characteristics of each type of detection for knowing different aspects associated to the person/object that we are following. And also, the tracker uses the fusion to differentiate one person from another.

* Tag: Black and white geometric figure, which serves in vision to detect some different points or positions. In our case, the tag serves to differentiate a specific person by the utilization of a vision detector. The vision detector sends a detection in the place that detects the tag.

* FME: Faculty of Mathematics and Statistics (Facultat de Matemàtiques i Estadística).

* MHT: Multi-hypothesis tracker.
Contents

1 Introduction 4
  1.1 Motivation ................................................. 5
  1.2 Project objectives ........................................ 6
  1.3 Outline of the Thesis ..................................... 6

2 Hardware and Software 7
  2.1 Hardware of Tibi and Dabo .................................... 7
    2.1.1 Tibi and Dabo ........................................ 7
    2.1.2 The sensors ........................................... 8
      2.1.2.1 Horizontal laser (Front: UTM-30LX, Rear: UTM-30LX-EW) ........................................ 8
      2.1.2.2 Eye cameras (Bumblebee2) ........................ 10
      2.1.2.3 LADYBUG2 ......................................... 11
      2.1.2.4 Segway (RMP 200) .................................. 11
  2.2 Software .................................................. 12
    2.2.1 Robot Operating System (ROS) .................................. 12
    2.2.2 ROS nodes that the tracker uses .......................... 15
      2.2.2.1 Laser people detector ................................ 16
      2.2.2.2 Vision tag detector ................................. 17
      2.2.2.3 Cargo ANTs laser object detector ................. 17
      2.2.2.4 People prediction .................................. 18
      2.2.2.5 Odometry node ..................................... 19

3 State of the art 21
  3.1 Introduction ............................................... 21
  3.2 Basic theory of the implemented tracker ...................... 22
    3.2.1 Motivation ............................................ 22
    3.2.2 Multi-hypothesis tracker ............................... 23
      3.2.2.1 Detections, tracks and the relationship between them ........................................ 23
      3.2.2.2 Prediction and correction of tracks (Kalman filter) ..................................... 24
      3.2.2.3 Association between tracks and detections, the association hypothesis and its probabilities: ........................................ 26
4 Robust multi-hypothesis tracker fusing diverse sensor information

4.1 Filtering of false positives and false negatives

4.1.1 Filtering false positives (confirmation function)

4.1.2 Filtering false negatives (deletion function)

4.1.3 Final probability formula

4.2 Dealing with tracking people in crossing situations

4.2.1 Using distance and the orientation of the velocity in crossing situations

4.2.2 Final probability formula

4.3 Fusion of different person/object detections

4.3.1 Changes in the hypothesis

4.3.2 Changes in the probabilities

4.3.2.1 Changes in the confirmation of tracks

4.3.2.2 Changes in the elimination of the tracks

4.3.2.3 Changes to confirm fusion tracks with probability

4.3.2.4 Final formula for the probabilities of the hypothesis

4.3.3 Kalman update step after the fusion

4.4 Example of the tracking process

4.4.1 Track process

4.4.2 Track life cycle

4.4.2.1 Track life cycle with only laser detector

4.4.2.2 Track life cycle with fusion

4.4.3 The different states of the tracks

5 Real experiments with the robot

5.1 Implementation parameters of the Robust multi-hypothesis tracker

5.1.1 Person confirmation

5.1.2 Person deletion

5.1.3 Person association distance

5.2 Implementation parameters of the Robust multi-hypothesis tracker fusing diverse sensor information

5.2.1 Person confirmation with fusion

5.2.2 Person deletion with fusion

5.2.3 Confirmation of the track state of fusion

5.2.4 Deletion of the track state of fusion

5.2.5 Association distance in fusion

5.3 Experimental results in robot interaction with people

5.3.1 Rosbag experimental results

5.3.1.1 Experimental results with rosbags of the FME

5.3.1.2 Experimental results with the rosbags in the Barcelona robot lab

5.3.2 Real experimental results

5.3.2.1 Results of real experimental in the laboratory

5.3.2.2 Experimental results with the real robot in the FME

5.4 Experimental results in Cargo ANTs project
5.4.1 Tracker improvements for Cargo ANTs ........................................ 72
  5.4.1.1 Filter static objects by velocity in a local tracking ................. 73
  5.4.1.2 Group Association by Distance. ....................................... 73
5.4.2 Experimental results .............................................................. 74
  5.4.2.1 Tracker in false positives filtering configuration ................. 74
  5.4.2.2 Tracker in false negatives filtering configuration ............... 74

6 Conclusions and future work ...................................................... 77
  6.1 Conclusions ............................................................................ 77
  6.2 Future work ............................................................................ 78

Annex .............................................................................................. 81

A Annex ........................................................................................... 82
  A.1 Coordinate change for local tracking ........................................ 82
1. Introduction

Thanks to the great evolution of robotics in the recent years, humanoids and social robots are available nowadays. The humanoid and social robots are endowed with human appearance and behaviour to get them to interact with us in different ways: guide/find people, work in cooperation with human beings, etc. For these robots are very important being able to detect and follow people in a robust way. Also, The ability to distinguish one person from other can be very important for this types of robots. We can check the importance of the tracking seeing the great number of different tracking systems implemented during years. In order to accomplish these important and complex following tasks it is necessary to have a good and robust tracking system which obtains the current position of the people in relation to the robot. These tasks are very challenging due to the unpredictable behaviour of people.

In order to obtain a good and robust tracker, we implement the multi-hypothesis tracker of Reid [28] and modify it including some improvements, in order to increase its robustness. Then, the tracker was modified adding some parts which deal with false positives, false negatives, and crossing situations. Also, the tracker was modified to be used in autonomous vehicles, in addition to social robots. Finally, the tracker was improved including the fusion of detections. The fusion of the tracker combines the different characteristics detected of the same person. The multi sensor fusion is a technique which refers to the combination of information extracted from different sensors. If we combine different characteristics from different types of sensor, we reduce the global uncertainty and limitations in the detection of people. Also, at the same time we have complementary perceptions which increases the accuracy and robustness of the detections.

This behaviour which uses different sensor information to obtain a better differentiation and recognition of the people is similar to the human sensory perception of the world. We have five senses to detect and recognize the world which surrounds us. Our five senses complement each other, and allows us to have a better perception and understanding of our environment. Given that, It is better for the robot to have the information from different sensors to recognize the world or the people around it. This allows the robot to have several measurements which ensure reliability of the data, avoid errors, improve the time response and improve the interpretation of what happens in the environment which surrounds the robot.
In our fusion case, we combine the different detections from different sensors (laser and vision). The laser and vision sensors give us some different characteristics of the person. The combination of these different characteristics allows us to get more robust people detections. In particular, allows us to distinguish a specific person to interact or follow.

The vision sensor contributes with a more accurate and specific differentiation of people, by using its appearance. The people appearance can be get it by different kinds of detectors, such as: color, shape or person skeleton detectors. I.e, with vision we can obtain a set of features which uniquely distinguish one people from another, and we can use it to differentiate between different people and also to focus in one specific person.

The laser sensor contributes to the detection with position accuracy, faster detection rate and less false negatives. The laser has a better position accuracy and less false negatives due to the type of vision detector which is used. And, the laser has faster detection rate, because the laser data processing does not consume much time. Due to the laser sensor characteristics, we can know where people are and follow them. Also, with laser we can cover a larger detection area than with vision, since the camera has a less vision area.

Our vision detector is a detector focused on detects a tag worn by the person interacting with the robot and the laser detector is a detector focused on detects the person’s legs. The tracker is designed to handle asynchronous inputs from laser and vision detectors which operate at different rates with a slight variation. The tracking algorithm has some weights assigned to the different object/people detectors. These weights reflect the confidence in the detector’s performance and behaviour to determine if the track is a real person or not. Furthermore, for us these weights serve to distinguish the person that interacts with the robot or not, because they introduce different levels in the probabilities of the tracks. Also, the tracker prioritizes keep the position detections which are more reliable. In our case the tracker has more weight for the laser position than for the vision position. The tracker has a general and expandable structure in the implementation of the fusion, e.g. the number of detectors could be changed or extended easily. Additionally, the tracker is able to use any detector which gives the people position in the ground plane. Finally, the tracker can adapt its behaviour online, changing its fusion parameters to obtain a better tracking performance in different environment situations.

### 1.1 Motivation

All social robots need to interact and coexist with people. In order to carry out these interaction tasks, robots have to know in each moment the location of the people. Also, robots must have to distinguish one person from another, specially in the cases that the robot is interacting with a specific person. Due to this, robots need a good and robust tracker which follows people in a robust way and also be able to distinguish them.
Also, the autonomous vehicles need to use the fusion to obtain a better differentiation between dynamic and static objects, using the different characteristics which brings the Lidar and radar sensors. Although, The adaptation of the algorithm to use Lidar is not implemented yet, because until recently we did not have a radar detector that could be used.

1.2 Project objectives

The main objective of the project is the extension of the implemented tracker, including the fusion of different object detections from different types of detectors. This fusion implementation aims to obtain a more robust tracking and to distinguish the person which interacts with the robot. Some times, we say object detections in general, because the tracker can detect people and mobile vehicles.

The specific objectives to fulfill are:

* Study and reformulate the previously proposed fusion theory for the tracker.
* Implement fusion in the tracker.
* Make the implemented fusion general and adaptable, so it can be used by different types of detectors. (The number and/or type of detectors used can be modified easily).
* Verify the good behaviour of the tracker with fusion in simulation, using rosbags.
* Verify the good tracking performance with fusion in the robots.

1.3 Outline of the Thesis

This final master thesis is organized in 6 chapters. In Chapter 1, we present the motivation, objectives and the outline of the thesis. In Chapter 2, we present the different hardware and software used. In Chapter 3, we present the state of the art in the tracking field, the fundamental theory for our tracker and the reasons for which we chose this theory. In Chapter 4, we present the theory of the improvements included in our tracker algorithm. Then, in Chapter 5, we present the obtained results with the social robots and the autonomous vehicles, which show the good tracking behaviour. In Chapter 6, we present the conclusions and future work of the thesis. Finally, in the Annex we explain the change of coordinates for local tracking.
2. Hardware and Software

This chapter contains a brief explanation of the software and hardware used. In this hardware section, we only explain the hardware used in the Tibi and Dabo robots, Figure 2.1. The hardware used for autonomous vehicles in the Cargo ANTs project will be briefly explained in the experiments of this project, Section 5.4. The software section has a brief explanation of the ROS (Robot Operating System), software which is used in our robots, and a brief explanation of some ROS nodes that the tracker uses.

Figure 2.1: Tibi and Dabo robots.

2.1 Hardware of Tibi and Dabo

Since the Tibi and Dabo robots are used, their characteristics will be enlisted, as well as the hardware these need to carry out their social tasks.

2.1.1 Tibi and Dabo

Tibi and Dabo are the social robots of the IRI. As seen in Figure 2.1. They have a sympathetic and friendly appearance to improve their interaction with people. The robots
have a similar appearance, their only visible differences being in their color and arms. At IRI, we attempt to assign a different behaviour and personality to each one; while Tibi represents the female version, Dabo represents the male counterpart. Nonetheless, both robots are able to use the tracking algorithm implemented.

2.1.2 The sensors

These robots use a group of different sensors to obtain the information of the environment that surrounds. The goal of these sensors is to interact with people mainly through detection and tracking, and to interact with the environment through navigation, mapping, and collision avoidance. The necessary sensors for the tracking approach are: two horizontal laser scans, the eye cameras or the LADYBUG2 and the robot odometry in the segway.

In Figures 2.2 and 2.3, we can see the position of all the sensors in the Dabo robot.

This part presents a summary of the characteristics of only the most relevant sensors considered for this project; for further information, We have the links of the datasheets of all sensors included in the final of each part.

2.1.2.1 Horizontal laser (Front: UTM-30LX, Rear: UTM-30LX-EW)

These are the two horizontal lasers that both robots contain. One of these located in the front and the other one in the rear of the robot. We can see the position of the sensors in
Figure 2.3: Dabo rear view

Figure 2.4: Horizontal laser
the figures pertaining to sensor placement, Figures 2.2 and 2.3. These sensors facilitate location and mapping through wall detection, allowing navigation and obstacle avoidance, as well as the detection of people.

![Figure 2.5: range scan of the horizontal laser](image)

The laser UTM-30LX-EW is an area laser scan sensor with a wavelength of $\lambda = 905$ nm and with security class 1. The laser allows the scanning of a semi circle of 270 degrees, divided in steps of 0.25 degrees (360 degrees /1,440 steps) of resolution, this allows to obtain a resolution in the detection of 1 mm respect to the real position of it, in distances between 0.1 m and 30 m. Also, the laser can detect up until 60 m but with less accuracy. The minimum detectable amplitude of the laser at 10 m of distance is of 130 mm. We can see the scan in Figure 2.5. The scan velocity is of 25 msec/scan (Motor speed : 2400 rpm). The laser returns the distance of each detected obstacle between its scan range. The diameter of the laser beam is 400 mm at the 30 m distance. The laser scan precision was of $\pm 30$ mm inside the range from 0.1 m to 10 m, and of $\pm 50$ mm inside the range from 10 m to 30 m. We can find the datasheets of these sensors in [5] and [6].

### 2.1.2.2 Eye cameras (Bumblebee2)

![Figure 2.6: Bumblebee2 cameras](image)

Both robots have this type of camera located in the eyes. The camera will serve to identify the required images to a specific person. The most relevant data for the image detector
is: The images have 640x480 pixels. The camera uses the color RGB compression format and the frame rate is the maximum that the camera can give us, 7.5 FPS. Furthermore, the image processing is made by the camera itself. We can find the datasheet of the sensor in [1].

2.1.2.3 LADYBUG2

![Image of LADYBUG2](image_url)

The Ladybug2 has six 0.8 MP cameras which enable video perception of 360°. Its six cameras are closely-packed Sony 1024x768 CCDs placed within 20 mm of each other. With high quality 2.4 mm microlenses enabled to collect video from more than 75% of the full sphere. We can obtain videos at 30 FPS. We can have multiple image output formats: JPEG, BMP, PNG, TIFF and more. Also we can have multiple video output formats: AVI, MP4 and FLV. We can obtain more information about this software in the official page [19]. With this camera we obtain a vision detection in all of the 360° and with it we can fuse with the laser detection, and obtain a more robust track of any person in the robot’s environment.

2.1.2.4 Segway (RMP 200)

The two-wheeled Segway, which use Tibi and Dabo, provides great mobility. This Segway allows them to move freely through the same places as people. This Segway also provides us with some information about the movement of the robot. This information is sent by the status message of the segway, which is sent each 0.01 sec. The structure of the status message can be seen in the Interface_Guide_for_Segway_RMP_2.0.pdf. And in the case of the odometry, we only want to know the displacement from the before robot position to the new robot position and the velocity of this displacement, both data is obtained from the two wheels of the Segway. This displacement and velocity is used to obtain the position odometry and the linear velocity of the odometry. Furthermore, we want to know the 3 angles of orientation: roll, pitch and yaw, to obtain the angular velocity of the odometry. We can find the user guide of the Segway in [3] and some specifications in [2]
2.2 Software

Throughout the implementation of the tracking algorithm, matlab, C++, and ROS were used. First, the adequate behaviour of the equations and algorithms are tested using matlab. Then, we transform and improve these algorithms in C++, making the libraries that our node uses. Finally, we create the ROS node that encapsulate the algorithms performed in C++, so these can be interpreted and used by the robot. For security, before the direct test of the code in the robot, we simulate the robot configuration in the experiments using rosbags, our node, the Rviz visualization tool and the dynamic-reconfigure tool which allows the modification of some node parameters online. Each of these tools will be better explained later. The last step is the real experiments with the robot, to validate the appropriate behaviour of our algorithm.

Given the widespread knowledge of Matlab and C++, the focus in the remainder of this chapter will be given to ROS (Robot Operating System); this tool is being used more and more in the context of robotic investigation.

2.2.1 Robot Operating System (ROS)

ROS is an open-source Software that can run in any operative system and robotic hardware. ROS provides libraries and tools to develop applications for robots. The system is based on a modular concept, that consists in dividing a complex task in simpler and reduced subtasks, these subtasks are encapsulated in executable files called nodes, which communicate amongst each other to generate the final behaviour of the robot. Each node has a unique name that distinguishes it from the rest of the existent nodes. The nodes can communicate with one another through three different communication procedures: publishing or subscribing it to a topic, providing or using one service, or using actions. One robot can have many independent nodes working in a cooperative manner with the global goal to achieve a complex behaviour of the robot. With this modularity we obtain
an important advantage that is a easy error correction, due to the fact that we can localize and resolve easily the errors inside small functions, reducing the complexity in comparison with the monolithic codes. Also, ROS is used since it allows the sharing of information between all kind of nodes, and it is possible to use previously developed processes and driver can easily be encapsulated in the nodes. Furthermore, it offers powerful tools for debugging, which saves time and allows the programmer to correct any errors found. Also, it offers the Rviz that is a visualization tool which allows to observe the behaviour that has been implemented in the programs before these are tried on the actual robot, this also saves time.

As we already presented in the previous paragraph, there are three communication protocols that the nodes use to communicate in ROS:

**Topics:** These are data buses that are used by nodes to exchange messages among them. The nodes can be of two different types: Publishers or Subscribers. The publishers generate data of one topic; for example the nodes that corresponds to the encapsulated drivers of the sensors are publishers that publish messages that contain the sensed values. The subscribers are nodes that are subscribed to the topics that publish another node. All the nodes can be at the same time publishers and subscribers of different topics, also we can also have multiple publishers and subscribers of the same topic. The nodes can be subscribed or can published topics in anonymous form, therefore, the production of information is independent of its use. In general, nodes do not know with whom they are communicating. The unit that performs communication and knows whether nodes are published or subscribed to a topic is the roscore.

**Services:** These allows communication between nodes that have requests and answers. One service is defined by two types of messages, one for the request and other for the answers. In these situations one node takes up the role of the client and sends the request to obtain a service, and waits until the server node sends the answer.

**Actions:** These are based on the same principle as the services, a request is sent and a response is received. The difference is that the action adds the ability to cancel the service, and therefore the nodes do not need to wait until it gets the answer. An action is defined by three messages: goal, feedback and result. The first contains the reason of the request, the second periodically provides the information of the state of the system, and the last is the result of the request.

**Rosbags:** These are data files where the publisher messages are stored, in these files we only save the information of the publisher nodes that we need to replay the real robot experiment offline in the computer. This data can be stored in topics of one node that performs a complex behaviour of the robot or can be only nodes that encapsulate one driver of one sensor and return the sensed data. These files allows us to recreate real situations offline and provide the option to verify or improve our created algorithms. Therefore, the rosbags may be designated for two functions: use our node in the environment rviz of our
computer to finish the debug of some errors, or record the experiments on the robot to analyse them further.

**Rviz:** A 3D graphical visualization environment for ROS. This graphical visualization environment allows us to draw the sensor outputs and create markers that show the output variables of our node, we can show only the parameters that we want to see. Also, it allows us to draw a robotic model at the point where the robot is located in the experiments, as well as, several coordinate axes that allows to know where are the position of each sensor and how the sensor is oriented in the robot. Finally, we can move the camera within the graphical environment to see different views of the simulation, so we can do a thorough check of the satisfactory behaviour of our programmed node. The procedure to check the nodes or see the good results of the nodes is: When the node is completed and ready to be tested, we create an executable file where we include the commands to run all the nodes that are needed, including our created node. In this executable we include the commands to run the rosbag file previously recorded, to replay the real robot experiments. And finally we give the commands to activate the rviz.

![Rviz graphical visualization](image)

**Figure 2.9:** Rviz 3D graphical visualization environment for ROS with the tracker. We can see the tracks with their id’s, the tracks in green are people detected using only the laser, and the track in blue is the track with fusion. Also, we can see an arrow in one person which represents its velocity. The green points that form lines are the laser detections of the walls.

**Dynamic-reconfigure:** The dynamic reconfigure is a package which provides a means to
change node parameters at any time without having to restart the node. In Figure 2.10 you can see the dynamic reconfigure for the tracker case, in it we can change all the modifiable parameters of the tracker.

![Dynamic-reconfigure with the tracker](image)

**Figure 2.10: Dynamic-reconfigure with the tracker**

**Node relations for tracking**

In Figure 2.11 we can see the implemented tracker and the relation between the tracker and the nodes that uses, as well as, the topics that these nodes use to communicate. The nodes that we can see in the figure are: the implemented tracker, `mht`; the laser people detector, `lpd`; the visual tag detector, `vdm`; and the laser people map filter, `laser_people_map_filter`, that is a filter that removes the laser people detections in the wall when we have a map. The nodes of the laser detector and the vision detector give their detection messages to the tracker. Then, the tracker node processes these detections and obtains the tracks. Finally, the tracker gives the tracks marker messages to the rviz, for the visualization of the real situations. The tracker can show different objects that are: tracks, detections, predictions, track velocities, group of tracks in crossing situations and groups of tracks.

### 2.2.2 ROS nodes that the tracker uses

In this subsection, we explain briefly the different ROS nodes implemented by other people that the tracker uses. First, we are going to explain the detector nodes that the tracker
Figure 2.11: In the image we can see: the $mht$ that is the tracker implemented, the $lpd$ that is the laser people detector, the $vmd$ that is the vision tag detector, the $laser\_people\_map\_filter$ that is a filter that removes the laser people detections in the wall when we have a map. We can see also the rosbag file and the reconfigure gui that is the dynamic reconfigure. The other names are topics that leave or enter in the nodes, according with the direction of the arrow.

uses to obtain the position of the detected objects in each moment. Furthermore, we explain other node that the tracker uses to make the prediction of each track. This node allows use the linear prediction with constant velocity and other predictions which takes into account the people behaviour. Finally, we briefly explain the odometry node, which is used to obtain the robot odometry, in the cases that track people in local way.

### 2.2.2.1 Laser people detector

This detector is able to detect people using the shape produced by a person’s legs, when they are detected by the horizontal laser. The laser pattern corresponding to the detected legs appears as two semicircles very close to one another. Also, in two consecutive detections people can not travel long distances. The node searches and identifies said pattern using certain features, such as: the number of points that are associated with the detection of the person, the standard deviation, the distance between one detection and the detection in the next instant, the circularity, the radius, the curvature, the average speed, etc. We can see the mentioned pattern in the Figure 2.12. The theory of this detector can be found in [8].

Figure 2.12: Laser leg detection.
The tracker uses this detector to find the positions and covariances of all the possible people detections. These positions are located in an x-y plane which corresponds to the floor.

2.2.2.2 Vision tag detector

This detector can detect tags by vision. We use this detector to detect people that wear a tag. We can find more information about the tag’s in [29]. The detector finds the tag in the image and gives its position in the space, then we only use the position in the plane of the floor to associate the vision detection with the track, with the purpose to adequately distinguish the person that is interacting with the robot. A tag example can be seen in Figure 2.13, this tag is the tag that we use in our experiments to detect the people.

![Tag example](image)

Figure 2.13: Tag example.

The tracker uses this detector to make the fusion between laser detections and vision detections. This detector allows the tracker to differentiate a specific person by the fusion improvement. This fact allows the robot to interact with a specific person. We can find this detector in [4]

2.2.2.3 Cargo ANTs laser object detector

This detector can group near laser points which compose a detected objects, said grouping is achieved through a k-nearest neighbours search, performed in the Euclidean space. Also, the object detector is able to detect possible moving objects by comparing the position of the objects in different laser frames. Furthermore, the detector sends the tracker a bounding box of the objects detected, which is why these detections have a square form. We can see the Rviz representation of said detections in Figure 2.14. The red boxes are all of the object detections, the blue boxes are the detections considered as possible moving objects, and the green box is the tracked object. In the Cargo ANTs the tracked object corresponds to the only real object which is moving in the scene.
Figure 2.14: Cargo ANTs laser object detector. The red boxes are all of the object detections, the blue boxes are the detections considered as possible moving objects, and the green box is the tracked object.

The tracker uses this detector, in the Cargo ANTs project to obtain the object detections and to know a first approach of the object mobility. The first approach of the objects mobility is useful for the tracker in order to adequately filter the static objects that detects in a local way. We need to filter the static objects because when we detect this objects in a local way, we have some velocity induced by the own movement of the vehicle that makes them appear as moving objects.

The laser object detector for Cargo ANTs project is submitted in a paper to the ROBOT’2015 - Second Iberian Robotics Conference.

2.2.2.4 People prediction

This node predicts the trajectories which perform the people [17]. These predictions are made by defining and estimating the intention of people when describing trajectories in social environments. That is, given the previous track position corresponding to a existent track which the tracker is following, this node makes the prediction of this track (calculates the new position of the track with a certain covariance after a certain time interval). These predicted positions are calculated taking into account two things: the interaction forces between the obstacles, other people and the robot; and the possible destination where people often go into a scene, some of them may be: doors, corridors, stairs, elevators, etc.

The tracker uses this node to obtain the predicted position of all the tracks at the current time. To do the prediction the node uses the previously corrected position, the average velocity of the track and the time interval between detections. The first time that the tracker has a new detection, it generates a new track which is sent to this node. In the next iteration, this node predicts the new track position in the current instant of time. The prediction is done in order to compare the current detections
with the tracks that the tracker has previously. At this moment, this prediction is made like a constant linear velocity propagation.

2.2.2.5 Odometry node

This node gets the wheel velocity of the left and right wheel of the segway status message, as well as the rotation in the roll, pitch and yaw angles. Using these values, the node calculates the linear and angular velocities and the translation in position of the segway odometry.

This node publishes a message with the segway's odometry and this message is used in local tracking. Only the odometry and the information obtained from the sensors are used in local tracking.
3. State of the art

In this chapter a brief summary on the state of the art in tracking is presented. First, we present an overview of the existent tracking techniques. Then, we present the multi-hypothesis tracking algorithm implemented by Reid, which is the foundation of our tracker, and the reasons to select this algorithm.

3.1 Introduction

People detection and tracking are very important for social robots, autonomous vehicles, surveillance system etc. Because of this, we can found many ways to track people which use different techniques. Make an exhaustive study of it is a hard work. Because of that, we will focus on showing some of the algorithms found which could be implemented at IRI. The differences between the tracking algorithms can be found in three different fields: The types of sensors and detectors used, which determine the accuracy and reliability of the data, along with the types of characteristics of the objects detected; the association techniques between detections and tracks used, which determine the accuracy of the association resulting in fewer changes of id’s and lost tracks; and the tracking algorithm used, which is important in the prediction of the people movement. Also, each tracking algorithm may have other features which can not be compared easily with other trackers.

Regarding the sensors and detectors the use of laser and vision detectors are very extensive, but there are also some algorithms which use different sensors. This field mainly depends on which sensors are available or compatible with our system, as well as the aim of the project. The sensors and types of detectors available in the state of the art are: laser scanners [7], [25], [26], [23] and [30]; vision (Monocular cameras [16]; stereo cameras [9], [18] and [22]; thermal cameras [13]; RGB-D Cameras [10], [12], [20] and [27]) and sonar [31]. Moreover, some of the implementations use fusion of some of the previous sensors to obtain more characteristics of the detected objects and to increase robustness and accuracy. These cases were found in [11], [13] and [31].

In the association techniques we can found the Nearest Neighbour technique which can use the Euclidean distance, the Mahalanobis distance or some other type of likelihood [9], [10], [27] and [31]; The Multi-hypothesis tracking (MHT) [28], [7], [16], [25], [26], [23], on which our tracker is based and the MHT will be explained in more detail in Subsubsection 3.2.2; the Joint Probabilistic Data Association [15]; an adaptation of the Joint
Probabilistic Data Association in [30]. Some of the works use a combination of these techniques, such as the case of Reid and the works which come from their, where a combination of Mahalanobis distance and Multi-Hypothesis is used to achieve the association between detections and tracks. Each method comes with advantages and disadvantages. The Multi-hypothesis method generates compatible hypotheses assignments which works very well in the detection-track association and continues these assignments in time to correct the \textit{id} changes between tracks. Also, the method is able to deal with an unknown number of tracks creation, confirmation, occlusion and deletion events in a probabilistic way. Besides, initial events can be extended by adding events such as fusion confirmation. As a drawback, the Multi-hypothesis tracking has a computational cost which grow exponentially with the grow of the number of tracks. This may prevent the use of the algorithm in real time. Then, the Joint Probabilistic Data Association can overcome this problem although it requires to know the number of tracking objects; this proves to be a significant drawback since the number of individuals in a scene is initially unknown. In the adaptation of the Joint Probabilistic Data Association, this last drawback is addressed with a way to estimate the number of tracks, but this is only an estimation used to simplify the tracking problem.

Regarding the tracking algorithm for the prediction of the people behaviour, with the aim of tracking their adequately, three main fields of study can be found: the Alpha-beta filters [18]; the Kalman filter strategies, (the regular Kalman filter [28], [7], [16], [25], [24]; the Extended Kalman Filter [20]; the Unscented Kalman filter [10] and [11]); the Particle filters [12], [13] and [30]. These methods have its advantages and disadvantages. The main advantage of the Kalman filter is in its recursive structure which allows the algorithm execution in real-time without storing detections or past estimations. The disadvantage of the Kalman filter is that only works well for a models without strong non-linearities. However, the Particle filters can work with models with non-linearities, but we have to do a prediction for each particle.

### 3.2 Basic theory of the implemented tracker

In this section we present the theory of the multi-hypothesis tracking algorithm implemented by Reid [28]. Also, we present the motivation for choosing this algorithm as foundation for the tracker.

#### 3.2.1 Motivation

In the sensors and detectors part, our tracker uses a laser leg detector and a vision tag detector. This was decided since these sensors are already available at IRI. Also, we have a tracking algorithm which can work with different types of detectors. Because of that, we could change or extend the types and numbers of detectors if is needed.
In the association techniques, we found three kind of tracking tendencies in the studied papers: the Nearest Neighbour, the Multi-hypothesis tracking and the Joint Probabilistic Data Association. The tracker uses a Euclidean or Mahalanobis distance combined with a Multi-hypothesis tracking. The choice of the Multi-hypothesis was because we do not know the number of people that we have to track (it is also desired for the tracker to perform in different environments). Also, we chose this method for the before showed advantages in Section 3.1. And during our tracking implementation we found the exponential drawback, but only when our tracker exceed much more the number of 50 tracked people, and we do not have in our environments so many people detected. Due to this, the tracker implementation using the MHT works well for what we need.

In the field of the tracking algorithm used, we found two big tendencies: the Kalman filter and the particle filter. Or tracker uses the Kalman filter. The basic advantage to choose the Kalman filter was its recursive structure which allows the algorithm execution in real-time without storing detections or past estimations. This advantage allows the track to go faster and also work in real time. Furthermore, if we used the particle filter in conjunction with the multi-hypothesis the computational time grows exponentially with less number of tracker people, and we would not be able to run in real-time. The main disadvantage of the Kalman filter is that only works well for a models without strong non-linearities. The models with non-linearities can be overcome by the particle filter. But, in our case this disadvantage of the Kalman filter does not affect us, because of the algorithm speed of 0.04 seconds which allows the consideration of a linear propagation for the people.

For all of these reasons, we choose the Multi-hypothesis tracker [28] as foundation of our tracker. Also, the Multi-hypothesis tracker [28] is the foundation of all the Multi-hypothesis trackers.

### 3.2.2 Multi-hypothesis tracker

This subsection presents the theory of the multi-hypothesis tracking algorithm implemented by Reid [28]. Our tracking algorithm is based on it, although only the most likely hypothesis is selected. Furthermore, we modified a little the tracking nomenclature of Reid because we improved the Reid algorithm including some new functions and the formulation subindexes could be confusing.

#### 3.2.2.1 Detections, tracks and the relationship between them

The Multi-hypothesis tracker of Reid established the existence of a set of multi dimensional detections at the moment of time $k$, $z(k) \equiv \{z_m(k), m = 1, 2, \cdots M\}$, and a set of tracks which was represented by their state, $x(k) \equiv \{x_n(k), m = 1, 2, \cdots N\}$. In position the tracks (Eq. 3.3) and the detections (Eq. 3.1) are points in the ground plane and have some matrices of covariances associated to each one. Also, the tracks have velocities in this plane. The covariance matrices of both are to take into account the possible detection
and prediction errors in the coordinates of the plane. We can see the covariance matrix for the detections in the Equation 3.2 and the covariance matrix for the tracks in the Equation 3.4). Finally, in the tracker each person is represented by a track and can be distinguished by the id’s of the tracks.

\[ z(k) = \begin{bmatrix} x \\ y \end{bmatrix} \]  

\[ R(k) = \begin{bmatrix} \sigma^2_{x} & \sigma_{x} \cdot \sigma_{y} \\ \sigma_{y} \cdot \sigma_{x} & \sigma^2_{y} \end{bmatrix} \]  

\[ x(k) = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \]  

\[ P(k) = \begin{bmatrix} \sigma^2_{x} & \sigma_{x} \cdot \sigma_{y} & \sigma_{x} \cdot \sigma_{v_x} & \sigma_{x} \cdot \sigma_{v_y} \\ \sigma_{y} \cdot \sigma_{x} & \sigma^2_{y} & \sigma_{y} \cdot \sigma_{v_x} & \sigma_{y} \cdot \sigma_{v_y} \\ \sigma_{v_x} \cdot \sigma_{x} & \sigma_{v_x} \cdot \sigma_{y} & \sigma^2_{v_x} & \sigma_{v_x} \cdot \sigma_{v_y} \\ \sigma_{v_y} \cdot \sigma_{x} & \sigma_{v_y} \cdot \sigma_{y} & \sigma_{v_y} \cdot \sigma_{v_x} & \sigma^2_{v_y} \end{bmatrix} \]  

The track state \( x(k) \) are related with one detection \( z(k) \) according to the equation:

\[ z(k) = H \cdot x(k) + v(k) \]

Where, \( z(k) \) is the current detection; \( H \) is the matrix which indicates the relation between the detections and the states of the tracks at an instant \( k \), in the ideal case that we do not have noise in the detections; \( v(k) \) is the white noise with zero mean and covariance \( R \) in the \( k \) instant of time.

### 3.2.2.2 Prediction and correction of tracks (Kalman filter)

The detections can be associated only with one track. Also, each time the positions of all the tracks must be propagated to be associated with the current detections (prediction step). Then, the detections are associated with only one tracks (association process). Finally, the tracks have to be corrected in position and covariance, taking into account the associated detection (correction step). The prediction and correction steps are achieved by the steps of the Kalman filter [21]. The association between detections and tracks will be explained in Subsection 3.2.2.3. In Fig. 3.1 is depicted a visual example of these three steps: track propagation, association and correction.
Figure 3.1: Track prediction, association and correction. The green triangles are the positions of the tracks with its id’s (1), the detections are the red squares and the ellipses denote the area of association between tracks and detections.

Prediction step: This step serves to predict the position in the current instant of time of the all tracks that we have. We have to calculate the new position and covariance of the tracks in the current time $k$, to associate them with the detections. The prediction is make taking into account the previous track position, velocity and covariance, as well as the time between the current time and the last time that we correct the tracks positions. Furthermore, In our tracker we consider a window of previous track positions, velocities and covariances to make the track prediction.

* We calculate the position of the tracks in the current instant of time, using the state transition matrix and the previous position of the tracks: \[
\bar{x}(k) = \Phi \cdot \hat{x}(k - 1).
\]

* We calculate the covariance of the tracks in the current instant of time:
\[
P(k) = \Phi \hat{P}(k - 1) \Phi^T + \Gamma Q \Gamma^T.
\]

Correction step: After the association of each track with its corresponding detection, we have to correct the track position and covariance taking into account the actual position of the detection associated to the track. This second part serves to correct the error that introduces the prediction step in the position of the tracks.

* We correct the track position: \[
\hat{x}(k) = \bar{x}(k) + K[z(k) - H \bar{x}(k)]
\]

* We correct the track covariance:
\[
\hat{P}(k) = \hat{P}(k) - \hat{P}(k)H^T(H \hat{P}(k)H^T + R(k))^{-1}H \hat{P}(k).
\]

* Kalman Gain: \[
K = \hat{P}(k) \cdot H^T \cdot R(k)^{-1}
\]

For the prediction and correction steps we use the Kalman filter with the next matrices:

\[
\Gamma = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 1
\end{bmatrix}
\]

\[(3.5)\]
\[ 
\Phi = \begin{bmatrix} 
1 & 0 & T & 0 \\
0 & 1 & 0 & T \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 
\end{bmatrix} 
\]  \hspace{1cm} (3.6) 

\[ 
H = \begin{bmatrix} 
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 
\end{bmatrix} 
\]  \hspace{1cm} (3.7) 

\[ 
Q = \begin{bmatrix} 
\sigma_{wx}^2 & 0 \\
0 & \sigma_{wy}^2 
\end{bmatrix} 
\]  \hspace{1cm} (3.8) 

Where, \( T \) is the time interval between two consecutive detections (the prediction interval), \( \sigma_{wx}^2 \) and \( \sigma_{wy}^2 \) are equal to \( qT \) and \( q \) is the variance in \( x \) and \( y \) of tracks, \( \Phi \) is the state transition matrix, \( \Gamma \) is the disturbance matrix, \( H \) is the measurement matrix, \( Q \) is the covariance matrix of white noise with zero mean.

3.2.2.3 Association between tracks and detections, the association hypothesis and its probabilities:

The multi-hypothesis tracker has a set of detections \( z(k) \) and a set of track states \( x(k) \) which are propagated and corrected by Kalman equations. The tracker algorithm begins checking by distance if the detections \( z(k) \) are associated to the propagation of one of the existent tracks \( x(k) \). For the association distance we use the next formula:

\[ 
(Z_m - H\bar{x})^T(H\bar{P}H^T + R)^{-1}(Z_m - H\bar{x}) \leq \eta^2 
\]  \hspace{1cm} (3.9) 

Where, \( Z_m \) is the position of the current detection, \( \bar{x} \) is the track state which was propagated, \( H \) is the observation matrix, \( \bar{P} \) is the covariance matrix of the track, \( R \) is the covariance matrix of the detection and \( \eta^2 \) is the association threshold.

These distances are distances with covariances (Mahalanobis distance) and the threshold physically matches some circular area around the track which determine if this detection can belong to the track or not. After apply the association with the mahalanobis-distance we can have diverse detections inside the zone of association of one track, or in the intersection of the zone of association of diverse tracks. These subgroups of tracks and detections associated between them are called clusters. Using these clusters, we divide all the group of detections and tracks of a scene in smaller subgroups which are easy to treat.
independently from the rest. These clusters can be formed by: a single track, a single detection, one track and one detection, or several tracks and several detections related between them. We can see a cluster example in the Figure 3.2. Besides, when we finish the process of associations between detections and tracks, after the multi-hypothesis, each detection only can be associated to one track.

![Cluster Example](image)

Figure 3.2: Cluster example: The green triangles are the positions of the tracks states with their id’s (1 and 2), the detections are the red squares and the ellipses denote the area of association between tracks and detections.

For each cluster we generate a group of possible hypotheses which relate the tracks and detections that can be associated by distance, $\Omega^k \equiv \{\Omega_i^k, i = 1, 2, \cdots, I_k\}$. To generate these hypotheses we have in mind two things: the hypothesis for the track in the previous time instant and the actual case of possible relations between detections and tracks. The possible hypotheses and clusters situations that we can have in each instant of time are the following. We can have only two hypotheses for an unassociated detection: false alarm or new track. For a detection associated to a track we can have 3 hypotheses: false alarm, detection associated to the track with id 1 (the detection is inside the association region of the track 1), or new track. For a detection which can be associated to two tracks, we can have 4 hypotheses: false alarm, detection associated to the track with id 1, detection associated to the track with id 2, or new track with a new id. In this last case, when we associate the detection to one of the two tracks if the other track does not have any detection associated, this track is a unassociated track. For better understanding of the hypothesis and its probabilities in the Table 3.2.2.3 is generated an example of the matrix of hypothesis and its corresponded probabilities for the associations case of the cluster which is shown in Figure 3.2. In the table, each hypothesis is a row and each column corresponds with one detection.
After this, we calculate the probabilities associated to each hypothesis in order to choose the best case of associations between tracks and detections. The probabilities are calculated with the next formula:

\[
P_i^k = P(\Omega_i^k | Z^k) = P(\Omega_g^{k-1}, \psi_h | Z(k)) = \eta P(Z(k) | \Omega_g^{k-1}, \psi_h) P(\psi_h | \Omega_g^{k-1}) P(\Omega_g^{k-1}) (3.10)
\]

Where, \( P_i^k \) is the probability of the actual hypothesis \( \Omega_i^k \) given some detections \( Z^k \) in the instant of time \( k \). Also, we can see \( \Omega_i^k \) as the joint hypothesis between the previous hypothesis \( \Omega_g^{k-1} \) and \( \psi_h \) which is the hypothesis of associations between the set of current detections and the predicted tracks. Finally, by means of the properties of the Bayes equation and Markov we can obtain the recursive equation which allows us to calculate the probability of each new hypothesis. This probability equation has three important parts: \( P(Z(k) | \Omega_g^{k-1}, \psi_h) \) is the likelihood of the measurements \( Z(k) \) given the association hypothesis; \( P(\psi_h | \Omega_g^{k-1}) \) is the probability of the current data association hypothesis given the prior hypothesis \( \Omega_g^{k-1} \); and \( P(\Omega_g^{k-1}) \) is the probability of the previous hypothesis. In addition, we have a normalization term \( \eta \) which serves to make the sum of all the probabilities of the current hypotheses equal to 1.
The likelihood of the measurements given the association hypothesis:

\[
P(Z(k) \mid \Omega_g^{k-1}, \psi_h) = \prod_{m=1}^{N_{det}} \mathcal{N}(Z_m - H\tilde{x}, B) \frac{1}{\sqrt{N_{\text{new}} + N_{\text{fal}}}}
\] 

(3.11)

\[
B = H\tilde{P}(k)H^T + R(k)
\] 

(3.12)

This part of the probabilities indicates the degree of similitude between detections and tracks which are being associated. This part \(\frac{1}{\sqrt{N_{\text{new}} + N_{\text{fal}}}}\) corresponds with the cases of false alarm or new track. Where \(V\) is the volume (or area in our case) which covers the sensor, \(N_{\text{fal}}\) is the number of detections associated to false alarms and \(N_{\text{new}}\) is the number of detections associated to new tracks. In the case that a detection is associated to a track, we have this part \(\prod_{m=1}^{N_{det}} \mathcal{N}(Z_m - H\tilde{x}, B)\) which corresponds to the normal distribution. Since, when a detection is associated to a track we consider that it has a function of Gaussian probability distribution with average equal to the relation between the detection and the propagated track and with matrix of covariances which relates the matrices of covariance of these detection and track. Therefore, this Gaussian gives us a measure of the similitude between the associated track and detection.

Probability of the current data association hypothesis given the prior hypothesis:

\[
P(\psi_h \mid \Omega_g^{k-1}) = \frac{1}{c} \frac{N_{\text{fal}}!N_{\text{new}}!}{M_K!} \cdot P_{\text{det}}^{N_{\text{det}}}(1 - P_{\text{det}})^{(N_{\text{TGT}} - N_{\text{det}})} \beta_{\text{fal}}^{N_{\text{fal}}} \beta_{\text{new}}^{N_{\text{new}}} \cdot (V^{N_{\text{new}} + N_{\text{fal}}})
\] 

(3.13)

This part performs the probability which fulfils the actual case of associations between detections and tracks in the actual hypothesis. Therefore, it includes all the terms of the probability which correspond to the possible cases of association. \(P_{\text{det}}\) and \(1 - P_{\text{det}}\) are the probability of detection and the probability of no detection, respectively. \(N_{\text{det}}\) is the number of detections associated to existent tracks, \(N_{\text{TGT}}\) is the number of the existent tracks, \(M_k\) is the total number of detections. \(\beta_{\text{fal}}\) and \(\beta_{\text{new}}\) are the Poisson probability distributions for the cases of false alarms and new tracks.

Probability of the previous hypothesis:

\[
P(\Omega_g^{k-1}) = P_i^{k-1}.\] 

This factor is the probability of the previous hypothesis from which derives the current hypothesis.
Final equation of probability for the hypotheses:

$$P^k_i = \eta P^{N_{det}}_i (1 - P_{det})^{N_{TGT} - N_{det}} \beta^{N_{fal}}_{fal} \beta^{N_{new}}_{new} \prod_{m=1}^{N_{det}} N(Z_m - H\bar{x}, B) P^{k-1}_i \quad (3.14)$$

$$B = H\bar{P}(k)H^T + R(k) \quad (3.15)$$

Where all the terms are previously explained.

This final equation is used to obtain the probabilities of each possible hypothesis of associations in each cluster every instant of time. For the purpose of this project, only the hypothesis with the highest probability obtained in each iteration is considered. Doing this, we remain with the best hypotheses and we reduce the computational cost of saving a large number of hypotheses and calculating many derivative hypothesis (it has an exponential grown). In addition to that, the worst hypotheses can be discarded because of the big difference between them and the higher probability. Therefore, we prune the tree of hypotheses by means of choosing a very restrictive quality threshold in the probabilities, by under of that we delete these hypotheses since they have been classified as unlikely.
4. Robust multi-hypothesis tracker fusing diverse sensor information

In this chapter we present the theory of the tracker implemented, The Robust multi-hypothesis tracker fusing diverse sensor information. This theory includes all of the improvements that we make in the theory of the multi-hypothesis tracking algorithm implemented by Reid, which we use as a fundamental theory of our tracker.

This tracker is used to track and distinguish people with Tibi and Dabo robots, in addition to differentiate and track dynamic and static objects in Cargo ANTs. The tracker can work in local coordinates using the robot odometry [14] (for Cargo ANTs) and in global coordinates using an existent localization map (for Tibi and Dabo robots). The local tracking is briefly explained in Section A.1.

In the field of the social robots, the tracker is able to follow multiple people independently without losing their id’s in crossing situations with other people or when being occluded by other people or objects of the urban surroundings. The improvements in crossing situations will be explained in Section 4.2. Also, the tracker is able to eliminate many false positives and false negatives, which will be explained in Section 4.1. Using the fusion part, the tracker is able to be more confident with the detections of people and can distinguish an only person which interact with the robot, which will be explained in Section 4.3. This differentiation of one person is necessary to allows the robot interaction with the same person. The tracker uses [17] for the tracks propagation, which was briefly explained in Subsubsection 2.2.2.4.

In the Cargo ANTs project the tracker is able to distinguish between dynamic and static objects and tracking the dynamic objects without lose it. Furthermore, the tracker can group different detected parts of the same dynamic object. These improvements will be explained in Subsection 5.4.1.

4.1 Filtering of false positives and false negatives

This part include some necessary improvements to have a better control of the confirmation, hold and deletion of tracks (This part was broadened in my PFC thesis ??). The
tracker includes two new parts: one to eliminate the false positives and other to eliminate the false negatives. The false positives are object detections that the laser confuses with people detections, for example a rubbish can be confused with a person detection. In these cases, we obtain a unconfirmed track which will be eliminated soon. The false negatives are punctual no detections of a existent person in the scene by the detector, in these cases we have a confirmed track which is remained in the scene during some time, to have time to be associated to one detection in the case that the track was truly a false negative.

4.1.1 Filtering false positives (confirmation function)

In order to reduce the number of false positives the confirmation function (Eq. 4.1) was added in the general probability function (Eq. 3.14) of the tracker. The confirmation function allows control whether a track is confirmed or not by growing the probability in a slower way than other tracker approaches. The elimination of false positives is possible because these detections are not real people and have a least constant detection than the real people. Due to that, we can eliminate the most part of the false positives by increasing the probability in a slower way and using the deletion function (Eq. 4.2). The deletion function will be explained in Subsection 4.1.2. The elimination of these detections is necessary because they can introduce errors in the tracking of real people, in addition to that, the robot has to interact with real people and not with objects which the robot can believe that are people. This part only applies when the track is still unconfirmed.

$$f_c(t) = \begin{cases} 
\beta_c & \text{if } l = 1 \text{ and the track is unconfirmed} \\
\beta_c + (\Delta \beta_c) \times (l - 1) & \text{if } l > 1 \text{ and the track is unconfirmed} \\
1 & \text{if the track is confirmed}
\end{cases} \quad (4.1)$$

Where, $\beta_c = 0.02$ is the confirmation constant which is obtained experimentally to provide continuity to the track probability, which has to continue from the value for the probability of new track (the probability of new track is 0.5). The $\Delta \beta_c = 0.01$ (can be inside of $[0, 1]$) is the confirmation increment which grow each time the track has been associated with one detection (this is incremented by the parameter $l$, where $l = 1$ for the first association after the generation of a new track). This function allows the slow increasing of the probability when the tracks has not been confirmed and has a detection associated. The lower the value of $\Delta \beta_c$, the lower the probability grow, and vice versa.
4.1.2 Filtering false negatives (deletion function)

In the case of false negatives, the tracks do not have any detection associated to it, in some time. In these cases, the tracker holds these tracks during short time by means of the new deletion function (Eq. 4.2). Also, this equation was added in the probability function (Eq. 3.14) of the tracker. The deletion function decreases slowly the track probability, so the tracks take longer time to be deleted and therefore the id’s are kept longer. If the detection reappears and is associated to the track, this keeps its id. But if the probability of the track falls below the no detection threshold, this track is eliminated. In the cases where the tracks have been deleted, we are considering that these people were out of the scene or these people were a false positives which they have not arrived neither to be confirmed. The deletion function is necessary because our people detector does not provide always a detection of the existent people (false negatives). The no detection of the people are due to different causes, such as: the occlusion of one leg by the other leg or the clothes that people dress make difficult the detection. In these situations the people are not detected because the dot pattern detected by the laser does not have the characteristics of the two legs.

\[
f_d(t) = \begin{cases} 
(\beta_d)^{\Delta t} & \text{if the track is unassociated} \\
1 & \text{if the track is associated}
\end{cases}
\] (4.2)

Where, \(\Delta t\) is the interval of time in seconds without detection associated with the track and \(\beta_d = 0.99\) is the parameter which allows the slow decreasing of the track probability. The behaviour of this function is as a decreasing exponential which decreases very slowly. Larger the value of \(\beta_d\) is, slower the tracking probability decrease, in the cases which we do not have detection associated with the track. Thanks to this slow decreasing we can keep the tracks id’s when we have false negatives.

4.1.3 Final probability formula

Including the previous two new parts, we improve the formula (Eq. 3.14) and obtain the new probability formula for each single detector (Eq. 4.3). This formula is used for calculate the probability of each possible hypothesis in an existing cluster.

\[
P_i^k = \frac{\eta^* \cdot f_c(t) \cdot P_{det}^{N_{det}} \cdot (1 - P_{det})^{N_{TGT} - N_{det}} \cdot \beta_{fal}^{N_{fal}} \cdot \beta_{new}^{N_{new}} \cdot \prod_{m=1}^{N_{det}} \mathcal{N}(Z_m - H\bar{x}, B) \cdot P_i^{k-1} \cdot f_d(t)}{\prod_{m=1}^{N_{det}} \mathcal{N}(Z_m - H\bar{x}, B) \cdot P_i^{k-1} \cdot f_d(t)}
\] (4.3)
With this final formula (Eq. 4.3), we obtain the probability behaviour for the tracks which are detected only with one detector. In the Section 4.4.2.1 we will be explained better the probability behaviour with only one detector.

### 4.2 Dealing with tracking people in crossing situations

This Section talks about other improvements which were introduced in the probabilities of the hypothesis, to take into account the orientation of the velocity of the tracks in the crossing situations. With this improvement we will be able to reduce the number of changes in the tracks’ ids during the crossing situations.

#### 4.2.1 Using distance and the orientation of the velocity in crossing situations

In the crossing situations the track differentiation by position was solved in the multi-hypothesis, but when the detections are very close we can fail in our association and in some cases we change the ids of the tracks. To correct this, we include a new part in the probabilities of the hypothesis which takes into account the orientation of the velocity in the association process during the crossing situations. This new part is important since in the cases where the detections are very close, the association by distance between tracks and detections is not sufficient to do a good association. In these cases, the use of the orientation of the velocity can solve in a good way more crossing situations, when the people which are crossing have different orientation in the velocity. We only compare the velocity orientation, since this is a very distinctive quality of each person in the crossing situations. In our tracker, the orientation of the velocity is more robust than the value of the velocity. This is due to the fact that we use the average speed of the track, calculated using a window of detected object positions. The average speed sometimes can not be a very distinctive quality for the people because they can walk at the same speed and they can reduce their velocity to do not collide in crossing situations. For these reasons, we prefer to use the orientation of the velocity to better distinguish the people in crossing situations.

Nevertheless, if in the future we use fusion with other sensors like radar which gives a more accurate value of velocity, also we can add a part in this function which takes into account the value of the velocity and not only the value of the orientation.

The tracker performs the next process to make the crossing differentiation by velocity. The first time that the tracks enter into a crossing situation we keep the initial orientations...
of the velocities of the tracks. During the crossing situation, we compare the stored orientations of each track with the current orientations of all the tracks. Then, we use the differential angle (Eq. 4.4) between the orientations to calculate the probability of the velocity orientation with the Equation 4.5. The probability value of this function allows to correct the changes of the id’s in the crossing situations because the value of this function changes the value of the probability of the hypothesis to make the good association wins, in the cases which the detections are very near and are difficult to associate well. It serves only in the case which the tracks does not change a lot their orientation in the crossing situation. In the case of groups where the people was very near we can have changes of id’s, but these cases are different than changes of id’s in the crossing situations. These changes of id’s in groups can be corrected with the improving of the treatment of groups in the tracker.

In Figure 4.1, we can see the selected function for the probability of the difference of the velocity orientation. The top of the curve is for the velocities orientation which is inside a margin of 0.20 rad up and 0.30 rad down from the initial orientation of the track. Inside this margin we consider that the track is the same initial track and in this case we improve the probability of the good association hypothesis to make it win. Then, if the velocity orientation is outside of this margin, we decrease the track probability in a slow way and with different potential functions to improve more different crossing situations. These margins and values was adjusted experimentally in the tracker and was the values which better work for our system, but we can change it in the future if is necessary for a better work of the tracker.

Figure 4.1: Probability for velocity orientation
The new function added in the probability to take into account the velocity orientation of the track in crossing situations is 4.5. This function is make with different parts because we want a function with parts which improve the probability of the hypothesis of the tracks that have similar initial orientation of the velocity in the crossing situations. Also, we want to have more control of the decreasing of the probability when the difference in the orientation of the velocity were out of this margin.

\[ \Delta V \theta = V \theta_f - V \theta_i \]  

Where, \( \Delta V \theta \) is the differential angle between \( V \theta_f \) and \( V \theta_i \), which are the final/actual velocity orientation of the track and the initial velocity orientation of the track when enters in the crossing situation, respectively.

\[ f_v(\Delta v \theta) = \begin{cases} 
3^{-|\Delta V \theta|} & \text{if } \Delta V \theta < -0.8 \\
\frac{0.337}{|\Delta V \theta|} & \text{if } \Delta V \theta > -0.8 & \Delta V \theta < -0.4 \\
\left| \frac{\log(|\Delta V \theta|)}{\log(3)} \right| & \text{if } \Delta V \theta > -0.4 & \Delta V \theta < 0.4 \\
\frac{0.337}{|\Delta V \theta|} & \text{if } \Delta V \theta > 0.4 & \Delta V \theta < 0.8 \\
3^{-|\Delta V \theta|} & \text{if } \Delta V \theta > 0.8 
\end{cases} \]  

### 4.2.2 Final probability formula

Finally, with the improvement of this section, the previous formula (Eq. 4.3) is again improved and become in the new probability formula (Eq. 4.6).

\[ P_i^k = \eta^k \cdot f_c(t) \cdot P_{\text{det}}^{N_{\text{det}}} \cdot (1 - P_{\text{det}})^{N_{\text{TGT}} - N_{\text{det}}} \cdot \beta_{\text{fal}}^{N_{\text{fal}}} \cdot \beta_{\text{new}}^{N_{\text{new}}} \cdot \prod_{m=1}^{N_{\text{det}}} \mathcal{N}(Z_m - H \bar{x}, B) \cdot P_i^{k-1} \cdot f_d(t) \cdot f_v(\Delta v \theta) \]  

Where all the parameters were explained previously in the Equations 3.14 and 4.3. Also, we include the orientation velocity function \( f_v(\Delta v \theta) \), (Eq. 4.5), which performs the improvement of the actual section.

### 4.3 Fusion of different person/object detections

Finally, our tracker makes fusion of detections obtained from different object detectors to improve our people tracking algorithm. With the fusion we obtain more robust tracks with
greater probability to be the objects which we are following. The objects which we can actually follow are: people and vehicles (moving objects). In addition, we can distinguish one person from other to allows the robot to interact with a specific person.

We introduced some changes to adapt the robust multi-hypothesis tracker to the fusion. Our fusion is made in the multi-hypothesis and its probabilities. We created a new hypothesis layer for the fusion case. In these hypothesis we can have only two possible values for each detector: 0 (the tracker does not detect the person, no detection case), 1 (the tracker detects the person, detection case). Also, we have the previous multi-hypothesis for each different detector because the multi-hypothesis between the detections of the same type serve to assign the observations of the same detector to only one track. The changes in the hypothesis will be explained in Subsection 4.3.1. Also, we changed the probabilities of the hypotheses. We created for each detector a layer of independent probability, each probability layer is equal to the formula of the probability for a single detector and its parameters are tuned taking into account the type of the detector. It is to say, each detector has its independent probability which is different of the probabilities for the others detectors and each detector has its different parameters to model its detections, no detections, new tracks, false alarms, tracks confirmed and deleted tracks. To obtain the final track probability we make a weighted sum with all of the probabilities obtained in the multi-hypothesis case of each detector. The use of the weighted sum for the final probability is because we want to improve the probability and do not penalize if one of the detectors does not detect one person, because we have people which only are detected by laser, due to the fact that they are not interacting with the robot. The change in the probabilities will be explained in Subsection 4.3.2. Finally, we have to take into account the position of the associated detection from each detector in the Kalman update. The change in the Kalman update step will be explained in Subsection 4.3.3.

4.3.1 Changes in the hypothesis

In the fusion, each detector is treated as a detector of the case with only one detector. Therefore, the matrix of hypothesis has its hypotheses in its rows and its columns correspond with each one of the different detectors. In the tracker with fusion we can give n cases, depending of the number of detectors that we have. These cases can be: we do not detect the person with any detector, we detect the person with only one detector, we detect it with two of the detectors, ..., we detect it with n detectors. To represents each one of these cases, we create a matrix of different hypothesis and the probabilities of these hypotheses depend on the combination of detectors that we have in the hypothesis. Since, it is not the same to detect the person with one sensor than with other sensor. In our case, only with laser and vision detection, the position of the detections inside the columns of the matrix of hypothesis is the following: the first column is assigned to the laser leg detector; the second column is assigned to the vision tag detector.
The possible hypotheses in the cases of fusion for each detector will be: false alarm or the tracker does not have detection with this detector, hypothesis 0; the tracker has detection associated to the track with this detector, hypothesis 1. The value of the hypothesis is placed in the corresponding column of this detector. In the Figure 4.2 we show a possible fusion case for the two detectors and in the Table 4.1 we show all the possible hypothesis of a fusion case with only two detectors. In the concrete case of the Figure 4.2, the hypothesis which wins is the H3. This hypothesis has one detection of each different detector associated to the same track. In the case that we only have laser detection, the hypothesis which wins would be the H1; in the case that we do not have any detection the hypothesis which wins would be the H0, etc. In each case only wins one of the hypothesis that we can have, since the hypothesis for the other probabilities have less value. If we do not have vision detection associated, the vision detection probability is 0 because the association distance is infinite. The probabilities of each hypothesis will be explained in Subsection 4.3.2.

Figure 4.2: Fusion example. The black triangle is the position of the track 1 (T1), the red square is the laser detection (7), the purple square is the vision detection (3). The green ellipse is the area of association for the laser detector, the blue ellipse is the area of association for the vision detector. The detection numbers are to remark that we can have several detections from each detector and also we can have several tracks.
Table 4.1: Possible hypothesis in the fusion case

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Laser</th>
<th>Vision</th>
<th>Meaning of the hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>H0</td>
<td>0</td>
<td>0</td>
<td>No detection associated to the track for all detectors</td>
</tr>
<tr>
<td>H1</td>
<td>1</td>
<td>0</td>
<td>Only laser detection associated to the track</td>
</tr>
<tr>
<td>H2</td>
<td>0</td>
<td>1</td>
<td>Only vision detection associated to the track</td>
</tr>
<tr>
<td>H3</td>
<td>1</td>
<td>1</td>
<td>Laser and vision detection associated to the track</td>
</tr>
</tbody>
</table>

4.3.2 Changes in the probabilities

For the probabilities of the fusion hypothesis we make a weighted sum of the probability of each different detector. This ensures that the non-detection of the person by one of the detectors does not completely eliminate the track of this person. We do not want the complete elimination of the tracks in the cases which the tracks are only detected by one detector because we can have people only detected by laser. Also, the detection of the person for all detectors improve the probability of the track and we can differentiate some people or have more robust tracking of people. Furthermore, this probability formula can be used in the case that we have fusion for all the existent people, only increasing the confirmation threshold to the fusion threshold. Now, the probability formula for the fusion case is 4.7. The probability behaviour of one track in the fusion case and the track states will be explained in Section 4.4.

\[
P_i^k = \sum_{i=1}^{n} W_{detector} \cdot f_c_{detector}(t) \cdot P_{det}^{N_{det}} \cdot (1 - P_{det})^{N_{TGT} - N_{det}} \cdot \beta_{fal}^{N_{fal}} \cdot \beta_{new}^{N_{new}} \cdot N_{detector}(Z_m - H\hat{x}, B) \cdot f_v(\Delta v, \theta) \cdot f_d_{detector}(t)
\]

(4.7)

Where, \(n\) is the number of different detectors that we have. Also, the values of the parameters inside the formula are different for each type of detector.

4.3.2.1 Changes in the confirmation of tracks

The function that we use to model the confirmation of the tracks in the case of fusion is the same as in the case of a single detector, but we have one confirmation function for each detector. When the probability of the track for one of the detectors surpass the confirmation threshold the person is confirmed, because we consider that this person was sufficiently continuously detected and is a existent person no a false positive. In general the detector which confirms the person is the laser detector because has a big detection
rate and a better precision in position, than the vision detector. The equation for this part is (Eq. 4.10).

\[
    f_{c\, \text{detector}}(t) = \begin{cases} 
    \beta_c & \text{if } l = 1 \text{ and the track is unconfirmed} \\
    \beta_c + (\Delta \beta_c) \ast (l - 1) & \text{if } l > 1 \text{ and track no confirmed} \\
    1 & \text{if } \text{the track is confirmed}
    \end{cases}
\] (4.8)

Where, the factors are the same presented previously and the subindex detector can be a \(L\) for the laser detector and a \(V\) for the vision detector.

### 4.3.2.2 Changes in the elimination of the tracks

In the case of elimination of the tracks in fusion, we consider another time the same previous deletion probability formula than in the case of a single detector, but we have one deletion function for each detector. This is to decrease the fusion probability in a slow way if we do not have vision detection but we have laser detection. This fact can be understood better in the probability graph for fusion of the Figure 4.4. The deletion function in the fusion is (Eq. 4.9). The decreasing of probability when we do not have vision detection is used to obtain different probability levels in the different cases that we can have: only laser detection or fusion. And with these probability levels we can distinguish when we have fusion or not in the tracks. Also, when we delete a track in the fusion, we do not have any detection associated to the track (we do not have neither laser detection nor vision detection) and both deletion functions (laser and vision deletion functions) are applied to the decreasing of the probability.

\[
    f_{d\, \text{detector}}(t) = \begin{cases} 
    (\beta_{d\, \text{detector}})^{\Delta t} & \text{if } \text{the track is unassociated} \\
    1 & \text{if } \text{the track is associated}
    \end{cases}
\] (4.9)

Where, the factors are the same presented previously and the subindex detector can be a \(L\) for the laser detector and a \(V\) for the vision detector.

### 4.3.2.3 Changes to confirm fusion tracks with probability

To confirm the tracks as tracks with fusion, we included other increasing function similar to the confirmation function. This was due to several reasons. One of the reasons is that, in our system is different the confirmation of a track as a real person and the confirmation of a track as the person which has the fusion of detectors, since we can have people detected only with laser and the robot has to detect it and follow it well. Other
reason is because we have more control of this two different facts with a function which has different parameters, namely we can make it grow in different way the confirmation as a person and the confirmation as a person with fusion. Also, the track confirmation is associated to each detector separately and the fusion is only associated for the unique case which we detect the person with all of the detectors.

This parameter $\Delta \beta_{fc}$ allows the slow grow of the probability in the case where the probability of the fusion surpass the value of 1 (The probability value for a person detected only with one detector is 0.988). When the fusion probability surpass the value of 1, means that the track has associated laser and vision detections at the same time. The slow grow of probability in the case of fusion is to eliminate some false positives due to the imprecise vision tag detection (Also, we can have tag detections in the walls or other places, like in the laser case). When the track probability surpass the probability value of 1.4, the track is considerate a track with fusion.

We have two parameters associated to the tracks to differentiate the tracks with fusion from the other tracks. These parameters are: the track type (type=laser=0 in the case that we have only laser detection and type=fusion=1 in the case that we have fusion) and the probability of the track (0.988 in the case that the track has only laser detection associated and 1.7 in the case that we have fusion). We need this two parameters because we can have some false positives in fusion which are very stable. With the type and the probability we can differentiate the person who actually has the tag with the track probability. We can distinguish this person from the others because the association of the tag detection with the track which corresponds to the person that worn the tag is more stable (the track has higher probability).

$$f_{\text{fus}}(t) = \begin{cases} 
\beta_{fc} & \text{if } l = 1 \text{ and track no fusion confirmed} \\
\beta_{fc} + (\Delta \beta_{fc}) \ast (l - 1) & \text{if } l > 1 \text{ and track no fusion confirmed} \\
1 & \text{if track fusion confirmed}
\end{cases} \quad (4.10)$$

Where, the $f_{\text{fus}}(t)$ is the fusion confirmation function; $\beta_{fc}$ is the beta no fusion confirmation; $\Delta \beta_{fc}$ is the increment of beta no fusion confirmation and $l$ is the number of fusion associations to the track.
4.3.2.4 Final formula for the probabilities of the hypothesis

For the implementation of the tracker with fusion we use the final probability formula (Eq. 4.12). This formula has the improvements of the previous improved formula (Eq. 4.6) with the improvements of the fusion section. This formula includes fusion of detections obtained from laser and vision detectors. In the fusion case the confirmation threshold is 0.8, the fusion confirmation threshold is 1.4, the no fusion threshold is 1 and the deletion threshold is 0.4. These thresholds allows the different track states and the behaviour of the probabilities in the fusion case. The track states and the tracks probability behaviour will be explained in Section 4.4. For the laser and vision weights we have the values of \( w_L = 1 \) and \( W_V = 0.7 \), this weights make a value of 0.988 for the track probability when the track is only detected by laser and a value of 1.68 when the track is detected by vision and laser (fusion). These probability levels are to have confirmed people and no deleted people only for laser detections associated to tracks. And with the vision detection we only make increase a little the probability in the cases which we have fusion for distinguish the person that interacts with the robot.

\[
f(P) = P_{\text{det}}^{N_{\text{det}}} \cdot (1 - P_{\text{det}})^{N_{\text{TGT}} - N_{\text{det}}} \cdot \beta_{\text{fal}}^{N_{\text{fal}}} \cdot \beta_{\text{new}}^{N_{\text{new}}} \cdot \mathcal{N}_{\text{detector}}(Z_m - H\bar{x}, B) \cdot P_i^{k-1}
\]

(4.11)

Where the factors of this part were explained in the Subsection 3.2.2.3, and we include all of these factors in this function \( f(P) \) to make understandable the explanation of the fusion probability function.

\[
P^k_i = f_{\text{fus}}(t)[W_L \cdot f_{L}(t) \cdot f_L(P) \cdot f_d L(t) \cdot f_v L(\Delta v\theta) + W_V \cdot f_{V}(t) \cdot f_V(P) \cdot f_d V(t) \cdot f_v V(\Delta v\theta)]
\]

(4.12)

Where, \( f_{\text{fus}}(t) \) is the previous fusion confirmation function; \( W_L \) and \( W_V \) are the laser and vision weights respectively; \( f_{L}(t) \) and \( f_{V}(t) \) are the confirmation functions for the laser and vision detectors respectively; \( f_d L(t) \) and \( f_d V(t) \) are the deletion functions for laser and vision respectively; \( f_v L(\Delta v\theta) \) and \( f_v V(\Delta v\theta) \) are the function for the velocity crossing situations for the laser and vision detectors respectively; and \( f_L(P) \) and \( f_V(P) \) are the probability parts of Reid that we use for the laser and vision detectors separately.

4.3.3 Kalman update step after the fusion

The prediction step is made in the same form that in the case of only one detector, because we have only one track for each person. After the association of the track with its corresponded detections obtained from each sensor, we have to correct the track position and covariance using all the associated detections. The fusion in the correction step serves
to correct the real position of the track with all the detected positions obtained from each sensor.

We make the Kalman updates in a chained way. We tested in matlab that the order in which we were doing the Kalman correction with each detector did not change the final result. Because of this, we choose to do first the correction with the laser detector and then with the vision detection, since we had tracks which have only laser detection.

1) We correct the position of the track with the laser detection:

\[
\hat{x}_1(k) = \bar{x}(k) + K_1[z_1(k) - H\bar{x}(k)] \tag{4.13}
\]

Where, \( \bar{x}(k) \) is the current prediction of the track state, \( \hat{x}_1(k) \) is the correction of the track state with the laser detection and \( z_1(k) \) is the laser detection.

2) We correct the covariance of the track with the laser detection:

\[
\hat{P}_1(k) = P(k) - \bar{P}(k)H^T(HP(k)H^T + R_1(k))^{-1}HP(k) \tag{4.14}
\]

Where, \( \bar{P}(k) \) is the current covariance of the track state which was predicted, \( R_1(k) \) is the covariance of the laser detection and \( \hat{P}_1(k) \) is the track covariance corrected with only the laser detection.

3) Kalman Gain:

\[
K_1 = \bar{P}(k) \cdot H^T \cdot (HP(k)H^T + R_1(k))^{-1} \tag{4.15}
\]

Where, \( K_1 \) is the Kalman gain for the laser case.

4) We correct the position of the track with the vision detection:

\[
\hat{x}_2(k) = \hat{x}_1(k) + K_2[z_2(k) - H\hat{x}_1(k)] \tag{4.16}
\]

Where, \( \hat{x}_1(k) \) is the previous correction using only the laser detection, \( \hat{x}_2(k) \) is the track state corrected with the laser detection and the vision detection and \( z_2(k) \) is the vision detection.
5) We correct the covariance of the track with the vision detection:

\[
P_2(k) = \hat{P}_1(k) - \hat{P}_1(k)H^T(HP_1(k)H^T + R_2(k))^{-1}HP_1(k)
\] (4.17)

Where, \(\hat{P}_1(k)\) is the previous covariance corrected by laser, \(R_2(k)\) is the covariance of the vision detection and \(\hat{P}_2(k)\) is the covariance corrected with the laser and vision detections.

6) Kalman Gain:

\[
K_2 = \hat{P}_1(k) \cdot H^T \cdot (HP_1(k)H^T + R_2(k))^{-1}
\] (4.18)

Where, \(K_2\) is the Kalman gain for the laser and vision case.

Where the matrices used are:

\[
z_1(k) = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}
\] (4.19)

\[
z_2(k) = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}
\] (4.20)

\[
R_1(k) = \begin{bmatrix} \sigma_{x_1^2} & \sigma_{x_1 \cdot y_1} \\ \sigma_{y_1 \cdot x_1} & \sigma_{y_1^2} \end{bmatrix}
\] (4.21)

\[
R_2(k) = \begin{bmatrix} \sigma_{x_2^2} & \sigma_{x_2 \cdot y_2} \\ \sigma_{y_2 \cdot x_2} & \sigma_{y_2^2} \end{bmatrix}
\] (4.22)

\[
\bar{x}(k) = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}
\] (4.23)

\[
H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}
\] (4.24)

\[
\hat{P}(k) = \begin{bmatrix} \sigma_{x^2} & \sigma_{x \cdot y} & \sigma_{x \cdot v_x} & \sigma_{x \cdot v_y} \\ \sigma_{y \cdot x} & \sigma_{y^2} & \sigma_{y \cdot v_x} & \sigma_{y \cdot v_y} \\ \sigma_{v_x \cdot x} & \sigma_{v_x \cdot y} & \sigma_{v_x^2} & \sigma_{v_x \cdot v_y} \\ \sigma_{v_y \cdot x} & \sigma_{v_y \cdot y} & \sigma_{v_y \cdot v_x} & \sigma_{v_y^2} \end{bmatrix}
\] (4.25)
4.4 Example of the tracking process

For a better understanding of the hole behaviour of the tracker. Now, I’m going to explain the track process, the tracks life cycle and the different states of tracks that the tracker has.

4.4.1 Track process

In the first iteration of the tracker we obtain a group of different detections. With each detection we generate a new track. We assign to the new tracks the position and covariance of the detections which create them. In the next iterations we predict these tracks, so we can compare with the positions of the actual group of obtained detections (This was explained in Subsubsection 3.2.2.2). At this moment, we associate each detection with an unique track by means of the association distance. Also, for each cluster of detections and tracks we select the best hypothesis of associations among them (This was explained in Subsection 3.2.2.3), taking into account the probability of the hypothesis (We select the hypothesis with the higher probability). Such that, each detection is associated with a single track. Finally, we correct the track predicted position with the position of the associated detection (This was explained in Subsubsection 3.2.2.2). We make it process in a chained way for all the detectors that we have in the fusion system. With these procedure we can associate the different characteristics provided by each sensor to the fusion track. Namely, if one track has associated one laser and one vision detection this track has the qualities which provide the laser and the vision sensor in the multi-hypothesis, the probabilities and in the Kalman correction part. The propagation is done independently of the detections (out of the fusion) and it is a linear constant velocity propagation.

Actually, we only generate new tracks with the laser detections and with the vision detections we only distinguish between people due to the fact that the vision detections are less constants and reliable in position. Also, for the vision detections we only correct the position of these tracks with the position of these detections when they are very close to the tracks.

4.4.2 Track life cycle

The track life cycle is very related with the track probabilities and the different states of tracks. In the first iteration the track starts as a New track. Then, the track changes its state to an Unconfirmed track before the track can be confirmed as a Confirmed track (real person) or eliminated as a false positive. The track changes its state to Confirmed track when the probability of the track surpass the Confirmation threshold which is 0.8 and the track is eliminated if the probability falls below the Elimination threshold which is 0.4. When the track is confirmed the track probability oscillate inside some range of probabilities [0.8-0.988]. These oscillations in the probability of the track are due to changes in the distance of association between detection and track. When the track is
a Confirmed track, it can become a Fusion track if it has associated a vision detection during some iterations because the track probability will surpass the Fusion threshold. If finally the track surpass the Fusion threshold the track changes its state to Fusion track. Then, some times in these before cases the track can be an Unassociated track with the laser detection, with the vision detection or with both. In the cases which the track is unassociated, it can be an Unassociated track by laser, by vision or by both. In these situations the track probability decays taking into account the time that the track is unassociated and it can become a No fusion track if its probability falls below the No fusion threshold or an Eliminated track if its probability falls below the Deletion threshold. This slow decrease of the probability is to deal with the false negatives in the both cases: to deal with false negatives in fusion and to deal with false negatives in the case which the real person is not detected by both sensors. We can see better this behaviour in the next probability graphs for the case of a person only detected by laser in the Figure 4.3 and for the case of a person detected with fusion in Figure 4.4.

4.4.2.1 Track life cycle with only laser detector

When we only have a laser detector, we use the final formula (Eq. 4.3) and we obtain the probability behaviour for the tracks which can be detected only with a laser detector. In the probability graph of the Figure 4.4, we can see the probability behaviour of two different tracks for the case which we only have the laser detector. This graph shows the probability of the tracks in each iteration of the tracker. The iterations can be translated in time taking into account that each iteration of the tracker takes 0.04 seconds.

Regarding the track 1, we can see how is initially created with the probability value for the new tracks. Then, we can see how the track one is in the state of Unconfirmed track during this state the track one increase exponentially its probability, because the track 1 has most of the time a laser detection associated to it. Finally, the track 1 surpass the Confirmation threshold and is confirmed as a Confirmed track which corresponds to an existent person. The Confirmation threshold has a probability value of 0.8. To confirm the track we use the confirmation function (Eq. 4.1).

For the track 2, we can see how is initialized as in the case of the track 1, but this track only has the first laser detection associated to it. Doe to that, the track enter in the state of Unassociated track and the probability of this track decrease in a exponential way using the deletion function (Eq. 4.2). When the probability of the track falls below the Deletion threshold, the track becomes a Deleted track and it is eliminated. The Deletion threshold has the probability value of 0.4.
4.4.2.2 Track life cycle with fusion

In the Figure 4.4, we can see the probability life cycle of a track which has fusion. This graph shows the probability of the track in each time. First, the track is initialized as a New track with the probability of 0.5, equal than in the case that we only have a laser detector. Then, in the cases which the track is associated to a detection, the track increases its probability using the confirmation function. When, the probability of the track surpass the Confirmation threshold of 0.8, is confirmed and become an existent track/person with the track state of Confirmed track. During some time, the track is only a person detected by laser, that is to say the track has not a tag detection associated. After this, the track has a tag detection associated and increase its probability in exponential way using the fusion confirmation function. When the track surpass the fusion Confirmation threshold of 1.4 become a Fusion track. The track has this state during some time. After this, the track has not a tag detection associated and its probability decrease slowly in an exponential way using the deletion function of the vision detector. Now, the probability of the track falls below the No fusion threshold of 1 and become a track only detected by laser during some time, with the track state of Confirmed track or existent person. Finally, the track is unassociated with a laser detection and its probability decrease slowly in an exponential way using the deletion function for the laser detector. When the probability of the track...
falls below the Deletion threshold of 0.4, become a Deleted track and it is eliminated. The times to the confirmation and deletion can be seen in the Section 5.1.

![Figure 4.4: Probabilities in the cycle of life of one track with fusion and its possible tracks states taking into account its probability.]

### 4.4.3 The different states of the tracks

In our tracker we have different states of tracks and some of them are related. For example, the track can be at the same time a No fusion track and an Unassociated track. All of these states of tracks are differentiated by their corresponding probability. Our probability represents the probability that the track is a real person. Also, in the fusion case the track probability represents the probability which distinguish the specific person who interacts with the robot from the rest of people. Our probability margins for a real person detections are between the 0.4 until 1.7. We extend the probability over the 1, because we want to remain the person confirmation for only the laser detection and also we have more ranges to distinguish better the different states of the tracks than if we try to remain between 0 and 1. These different states are: New tracks, Unconfirmed tracks, Confirmed tracks, Fusion tracks, No fusion tracks, Unassociated tracks and Eliminated tracks.

**New tracks:** Are tracks which corresponds to the first detection of one person. They are the initialization of the existent tracks and have a initial probability of 0.5.

**Unconfirmed tracks:** Are existent tracks which use the part added in the probabilities to
filter the false positives and its probabilities are inside the margin [0.5,0.8]. If the tracks corresponds to a real person, these tracks are in this state until their probability surpass the **Confirmation threshold** (*Confirmation threshold* = 0.8) and become confirmed tracks; or if the tracks corresponds to a false positives, these are eliminated using also the deletion function. The tracks can be confirmed by laser or vision. The track confirmation means that the track is considered as an existent person.

**Confirmed tracks:** Are tracks which their probability surpass the *Confirmation threshold*. When the probability of the tracks surpass the *Confirmation threshold* are considered existent people. These tracks use the probability without the two added parts to deal with false positives and false negatives. The probability range of the **Confirmed tracks** is [0.8,1]. The oscillations of the probability inside this margin is due to changes in the association distance between the detection and the track.

**Fusion tracks:** Are tracks which have associated observations from different types of detectors at the same time, in our case laser and vision. These tracks use the probability extension for the fusion and their probabilities are inside [1,1.68]. The tracks are considered **Fusion tracks** when their probability surpass the *Fusion threshold* (*Fusion threshold* = 1.4). This threshold is not directly 1 to eliminate some false positives in the fusion case.

**No fusion tracks:** Are tracks which were considered **Fusion tracks**. But now, these tracks only have the laser detection associated to they. This tracks appear when the probability of the tracks considered as **fusion tracks** falls below the *No fusion threshold* (*No fusion threshold* = 1).

**Unassociated tracks:** Are tracks which are not associated with any detection at this time. These tracks use the deletion function. These tracks are eliminated in a slow way to deal with false negatives. And they are remained in the scene if they are truly false negatives of one person detection, but they are eliminated if correspond to a false positives or a person which goes out of the scene. Their probabilities are between the range of [0.4,1.65] and their probabilities go down taking into account the time that these tracks are not associated to one detection. Inside this state of tracks can be different unassociations: unassociated by laser or unassociated by vision or unassociated by both (the **Unassociated tracks**). In each case applies the unassociation function of the unassociated detector. Although, this state of **Unassociated track** usually is associated with tracks which do not have any detection associated.

**Eliminated tracks:** Are tracks which are eliminated by the deletion function of the probability formula. The tracks are eliminated when the probability of the track falls below the *Elimination threshold* (*Elimination threshold* = 0.4).
5. Real experiments with the robot

In this chapter we show the good performance of the tracker implemented which was tested with different kind of data. First, We tested the algorithm with rosbangs and finally we tested the algorithm in the real robot. Also, we made the experiments in the two fields where the tracker can work: the autonomous vehicles (Section 5.4) and the social robots Tibi and Dabo (Section 5.3). As well as, we show some important parametrization of the tracker (Section 5.1). In Section 5.1, we present the parametrization of the tracker in the case of the robust multi-hypothesis tracker and in Section 5.2 we present the parametrization of the tracker in the case of the Robust multi-hypothesis with fusion. Then, we will proceed to show the results obtained with the rosbags in the Faculty of Mathematics and statistics (FME) in Section 5.3.1.1 and in the Barcelona robot lab in Section 5.3.1.2. Finally, we show the results in the experiments with the real robot in Section 5.3.2.2.

5.1 Implementation parameters of the Robust multi-hypothesis tracker

We told that the implemented tracker is a adaptable and a flexible tracker which is able to perform a good tracking in a multiple situations. To obtain this behaviour the tracker has different parameters which can be changed taking into account the environmental situation. When we are in a specific environment these parameters are fix, but these parameters can be adaptable if the scene conditions change. We have tested two groups of parameters for indoor (inside robot lab) and outdoor (FME and Barcelona robot lab). Few of these parameters can be the same for both cases.

In this section, we are going to present the values of the parameters to confirm, associate and delete tracks. The parameters of this section are focused in the case which we only have one detector and this detector is a laser detector. The Robust multi-hypothesis tracker was only implemented using the laser detection. In Section 5.2 we show the values of the parameters for the Robust multi-hypothesis tracker fusing diverse sensor information, which is the tracker with fusion of the laser and vision detections.
5.1.1 Person confirmation

In the experiments when a track appear at the output of the tracker is because the track was confirmed. If the track is unconfirmed only exists inside the tracker node. The other states of the tracks can be seen at the tracker output. We can see the tracks obtained in the output of the tracker with the Rviz (the ROS visualization tool). In this case, we only have a laser detection and the track only can be confirmed by laser.

For the confirmation of the tracks, we use the confirmation function included in the probabilities of the hypothesis. The confirmation function was explained in Sections 4.1.1 and 4.3.2.1. This function allows the slow growing of the probability and the slow growing of the probability is done using the beta increment confirmation of the confirmation function. In the Table 5.1, we show some different values for the beta increment confirmation of the laser detector and the corresponding values for the time which takes the tracks to be confirmed (confirmation time). The time to confirm the tracks can be changed to adapt the tracker to different situations. The time to confirm the tracks can be changed by changing the beta increment confirmation. If we want a fast reaction of the tracker, confirming faster the tracks, we have to increase the value of the beta increment confirmation. When we increase the beta increment confirmation, we decrease the confirmation time. On the other hand, if we want to eliminate more false positive, we have to decrease the value of the beta increment confirmation. And with this we increase the confirmation time. When we confirmed a track, we consider that the track is the representation of a real person.

<table>
<thead>
<tr>
<th>Beta confirmation for a person detected with laser</th>
<th>time for a track confirmation</th>
</tr>
</thead>
<tbody>
<tr>
<td>laser beta increment confirmation</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>0.03</td>
</tr>
</tbody>
</table>

Table 5.1: Some values for beta confirmation in the case of a person detected by laser

Due to the differences of the indoors and outdoors environments, we use a different beta increment confirmation for each environment. We use a value of 0.005 for indoor, where we can have more false positives and these can be more stable. We use this value because we can filter more false positives with it. Otherwise, we use a value of 0.03 for the beta increment confirmation in cases where we have a very clean outside environment, when we have a few false positives and these are not very stable. Also, we can have a faster confirmation of the tracks when we have a laser people detector with a map filtration, due to the fact that we eliminate all the false positives which may be in walls or things that we have over the map.
5.1.2 Person deletion

The track is deleted when the probability of the track falls below the elimination threshold which is 0.4. That is to say, we use the probability formula only with laser, Eq. 4.3 and this formula has a deletion function for the laser detector. This deletion function has a parameter which can have different values for different environment situations. The deletion function was explained in Sections 4.1.2 and 4.3.2.2. With the beta deletion we can have more control in the elimination of the tracks. This is to say, by changing the value of the beta deletion we can remove a track diminishing its probability in a slower way or in a faster way.

Then, in the Table 5.2 we can see the different values of the beta deletion for a person detected only by laser and the time which takes the track to be eliminated. Also, we can control and change these values to obtain different tracking behaviours. If we have a detector very continuous and precise in position we can eliminate faster the tracks when we do not have detection associated to the track. But if we have a detector which has false negatives, as in our case, we have to keep more time the tracks to do not change its id’s in the cases of false positives.

In the case of a person only detected by laser, in the Table 5.2 we can see some values of the beta deletion which we usually use in some situations. The value of 0.99 is used in indoor because we have more false negatives. We have more false negatives since we have more objects in the environment. This value allows us to do a slower elimination, to deal with more false positives. Also, this value can be used in outdoor where we have columns or objects which can occlude or difficult the person detection, to deal with a little occlusions and not only with false negatives. Then, the 0.97 value is used in outdoor where we have less false negatives. The intermediate values are used taking into account the difficulty of the environment.

<table>
<thead>
<tr>
<th>Beta deletion only person detected with laser</th>
<th>time for a track deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td>laser beta deletion</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>89.96 sec = 1.48 min</td>
</tr>
<tr>
<td>0.985</td>
<td>59.82 sec</td>
</tr>
<tr>
<td>0.98</td>
<td>44.75 sec</td>
</tr>
<tr>
<td>0.97</td>
<td>29.68 sec</td>
</tr>
</tbody>
</table>

Table 5.2: Some values for beta deletion of a person detected only by laser

5.1.3 Person association distance

Regarding the association distance, in the laser case we have a association distance equal to 1.5 meters. This distance is to associate the laser detections with tracks. For this distance in the laser case we can have until 2 or 3 meters of association distance, due to the fact
that, the laser is very precise in position and we have the multi-hypothesis and the velocity orientation for the associations in crossing situations. The augment of this distance allows us to recuperate more tracks in cases of cluttered environments with lot of objects and where we have more false negatives. The association distance normally can oscillate inside this margin $[0.5, 2]$. Also, the association distance depends of the association function that we use (Euclidean or Mahalanobis). In some cases is good the use of the Mahalanobis distance or the use of the Euclidean distance. Normally, for association distances bigger we use the Euclidean distance and for association distances small we use the Mahalanobis. We use each distance in each case because when we use the Mahalanobis we use also the increment of the covariances of the tracks, which we have when we do not have associated detection for the track. But in the case that we use the Euclidean distance we do not use the increment of the covariances of the tracks. Furthermore, the use of the Mahalanobis distance has the problem that if the track was long time with false negative detections and finally we have a detection associate to the track, we can obtain a big velocity and we can lose the track. Also, when we use the Mahalanobis distance, in the cases that we do not have a detection associated to the track we can change the tracks id’s when the tracks are near, due to the biggest covariance of the track.

5.2 Implementation parameters of the Robust multi-hypothesis tracker fusing diverse sensor information

In the fusion case, we also have different parameters which allows us to adapt the fusion to obtain different results. In our case, we can adapt the time to confirm a track with the state of Fusion track taking into account the number of associations of the vision detection with the track which we need to confirm the track. It is as the track confirmation for the laser. Also, we can control the elimination of the track state of Fusion track when during some time we do not have vision detection associated with the track. Finally, we can choose the track that we consider the Fusion track by the association distance between the track and the vision detection.

5.2.1 Person confirmation with fusion

In the fusion case, the track can be confirmed for both vision and laser detectors. But normally the track has more associations with the laser detector, since the laser detector is more constant and more precise in position than the vision tag detector. Due to this, the laser detector is which usually confirms the track. The behaviour of the track confirmation in the fusion case is the same as in the track confirmation with only laser. We use the same confirmation function for both detectors, but we can change its parameters. Now, the beta confirmation for laser and vision are the same as in the case of only laser and we have the same values for the confirmation time. We can see the values for beta increment confirmation and the confirmation time in the Table 5.3.
Table 5.3: Some values for beta confirmation in the case of one person detected by laser and vision

<table>
<thead>
<tr>
<th>Beta confirmation for a person detected by laser and vision detectors</th>
<th>time for a person track confirmation</th>
</tr>
</thead>
<tbody>
<tr>
<td>laser and vision beta increment confirmation</td>
<td></td>
</tr>
<tr>
<td>0.005</td>
<td>5.55 sec</td>
</tr>
<tr>
<td>0.01</td>
<td>2.77 sec</td>
</tr>
<tr>
<td>0.02</td>
<td>1.38 sec</td>
</tr>
<tr>
<td>0.03</td>
<td>0.9257 sec</td>
</tr>
</tbody>
</table>

Actually in the fusion case, we use the beta increment confirmation with value equal to 0.01 for both laser and vision detectors. We prove experimentally that this value is a good confirmation value for the tracks of our system. With this value we filter a lot of false positives and we have a reasonable confirmation time for the tracks. This is due because, the tracker runs at 24 Hz and we have at least one group of detections each 0.04 sec. Then, in 2.77 seconds the tracker obtain 69 groups of detections. With these amount of continuous detections we are sure that the track is a real person and also we have a small confirmation time.

5.2.2 Person deletion with fusion

The track is deleted now when the probability of the track with fusion goes under the elimination threshold, which is 0.4 as in the laser case. That is to say, we use the fusion probability formula (Eq. 4.12) and this formula has a deletion function for each detector which can have different values of the parameters for different detectors. This deletion function was explained in Sections 4.1.2 and 4.3.2.2. With different values for the beta deletion we can have a faster or slower track elimination (in the case that we have only laser detection or we have laser and vision detections) and we can have a faster or slower confirmation of the Fusion state. When we have a person who worn the tag, the track elimination is controlled by the laser and vision beta deletion. The elimination of the track state of fusion state is only controlled by the vision beta deletion because is the detection which distinguish the fusion cases. The deletion of the fusion state will be explained better in the Section 5.2.4.

Then, in the Tables 5.2 and 5.4 we can see the different values of beta deletion for a person detected only by laser and a person detected by both detectors. Also, in the tables we can see the times which takes the track to be eliminated. These values are calculated tacking into account the maximum probability of the track in each case. If the track elimination starts with less probability, this takes less time, but normally the values are near of the showed values. Also, we can change these values to obtain different tracking behaviours.

In the Table 5.4, we have the case of a person with fusion of laser and vision detections. For indoor and outdoor we use the same principle as in the case that we have only one
laser detector. We actually use the 0.97 for laser (deal with false positives in the person deletion) and the 0.985 for vision (deal with false positives of the vision detector in fusion). The value of the laser beta deletion is less than the value of the vision beta deletion, since the laser has a more constant detection with less false negatives. The value for the vision beta deletion is bigger because we want to remain the track state as a Fusion track in the cases of false positives in the vision detection. Also, we can change these values to obtain different tracking behaviours, but always the beta deletion for the vision tag detector will be greater than the beta deletion for the laser detector due to the fact that the vision has more false negatives.

<table>
<thead>
<tr>
<th>Beta deletion for a person detected with laser and vision</th>
<th>Laser beta deletion</th>
<th>Vision beta deletion</th>
<th>Time for a track deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.99</td>
<td>0.99</td>
<td>142.76 sec = 2.38 min</td>
<td></td>
</tr>
<tr>
<td>0.985</td>
<td>0.985</td>
<td>94 sec</td>
<td></td>
</tr>
<tr>
<td>0.97</td>
<td>0.97</td>
<td>71 sec</td>
<td></td>
</tr>
<tr>
<td>0.97</td>
<td>0.985</td>
<td>65 sec = 1.08 min</td>
<td></td>
</tr>
<tr>
<td>0.97</td>
<td>0.99</td>
<td>80 sec = 1.33 min</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>0.985</td>
<td>121 sec = 2.02 min</td>
<td></td>
</tr>
<tr>
<td>0.985</td>
<td>0.98</td>
<td>110 sec = 1.83 min</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4: Some values for beta deletion of a person detected by laser and vision detectors

### 5.2.3 Confirmation of the track state of fusion

We can control the confirmation of the fusion with the fusion confirmation function by changing the beta incremental fusion parameter. The fusion confirmation function was explained in the Section 4.3.2.3. In the Table 5.5 we can see some different values for this parameter and the time for the confirmation of the track as a Fusion track. Actually, we use the value of 0.001 for the beta confirmation fusion, since we have a lot of false positives in the vision tag detector. This value allows the elimination of some false positives. We prefer include this function to have a better control of the fusion confirmation and make the fusion confirmation independent of the track confirmation as a person. In this case we need 128.25 continuous vision associated detections to confirm a track with fusion. We emphasise that this value is for the huge amount of false positives that we have with the tag detector.
Table 5.5: Some values for beta fusion confirmation and its correspondent time for the fusion confirmation of the track.

<table>
<thead>
<tr>
<th>Beta fusion confirmation</th>
<th>Confirmation fusion track time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>5.13 sec</td>
</tr>
<tr>
<td>0.0013</td>
<td>4 sec</td>
</tr>
<tr>
<td>0.0017</td>
<td>3 sec</td>
</tr>
<tr>
<td>0.0026</td>
<td>2 sec</td>
</tr>
</tbody>
</table>

5.2.4 Deletion of the track state of fusion

We can control the no fusion consideration of the track by the beta deletion of the vision detector which is in the deletion function for the vision detection. With our two types of detectors the deletion of the fusion state of the track is made in this way, because the laser detector is more reliable and we can have people detected with only the laser detection and the vision detection is used only to differentiate a specific person from others. In the Table 5.6, we can see some different values for the vision beta deletion and the time which takes the deletion of the fusion state for the track. Actually, we use the value of 0.99 for the vision beta deletion (The laser beta deletion was fix at 0.97). Also we can change the vision beta deletion to other value for change the tracker behaviour.

Table 5.6: Some values for beta fusion deletion parameter and its correspondent time for the fusion deletion of the track.

<table>
<thead>
<tr>
<th>vision beta deletion</th>
<th>time for the deletion of the Fusion track state</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.999</td>
<td>524 sec</td>
</tr>
<tr>
<td>0.99</td>
<td>52 sec</td>
</tr>
<tr>
<td>0.985</td>
<td>34 sec</td>
</tr>
<tr>
<td>0.98</td>
<td>30 sec</td>
</tr>
</tbody>
</table>

5.2.5 Association distance in fusion

In fusion, we can control the association distance of the different detectors that we use in the track-detection association. The values of each detector for the association distance can be as the association values for the laser detector. The possible distances are inside [0.5-3] meters.

In the fusion case, we actually use 1.5 meters for the laser association and 0.5 meters for the vision association. In our case, we distinguish the track considered as the fusion track or not with the association distance of the vision detector. With this distance we can control the selection of the track with fusion by the proximity of the vision detection.
The vision detector has only 0.5 meters for the association distance, due to the fact that we have many false positives and the vision detector is not very precise in position in most of the cases.

For a detector very precise in position as the laser detector we can have a huge association distance to associate the detection with the track. If we have a detector very imprecise in position we can not have a huge association distance. A huge association distance goes bad, since we can associate this detection to other people and in the Kalman correction we correct the track position with a bad position of the detection. Because, if the detection is at one meter of the track and this position is not a real position we associate the track with a bad position and we can lose it. We need a detection precise in position to do a good Kalman correction. For this reasons, we select the 0.5 meters for the association distance of the vision detector and we try to keep more the fusion tracks to do not lose it in the cases which we have long periods without vision detection associated with the tracks.

5.3 Experimental results in robot interaction with people

5.3.1 Rosbag experimental results

In the first proves of the implemented algorithms, we used rosbags previously recorded which contains the real robot data. We had a lot of rosbags to prove the algorithm, due to the fact that a PHD student uses the same system for following a specific person with the two robots, but actually this system uses a less robust association than the association performed with fusion in the tracker. For this fact, The robag experiments which we used to prove the tracker were for finding and following one specific person. The use of these rosbags do not introduce any problem when we test the tracker. We obtained good results using only the laser and vision tag position recorded in the rosbags. But, we have to take in mind that the behaviour of the robot was recorded previously and the robot does not take into account the track with fusion of detections for following the specific person. For this fact, in some cases we can see a strange behaviour of the robot which does not follow the track with fusion, but this is not the tracker objective. The tracker objective is distinguish and follow the person with the fusion of detections and this was performer well. In these experiments with rosbags we tested the good behaviour of the tracker previously to the tests in the real robot.

5.3.1.1 Experimental results with rosbags of the FME

When the algorithm was ready, first we performed the experiments to test the good behaviour in the rosbags of the FME area outside the IRI. This area is a square yard free
of obstacles, of approximately 8x8 m², and we can add obstacles if is necessary. You can see the place in the Figure 5.1.

![Figure 5.1: FME scenario](image)

In the rosbags of the FME, we had a free of obstacles area and the two robots were detecting and following one person which wear the tag. In this experiments, we can proved and adapted the values of the tracker parameters for the association distance and the values of the tracker parameters to confirm and delete the fusion track. The values of the parameters are the previously explained in the Section 5.2.

In the Figures 5.2, 5.3 and 5.5, we can see how the person which has fusion are detected as a fusion track in blue and we detect as a person the Tibi robot. But for the tracker, Tibi is a person without the tag. Also, we can not detect the Tibi robot as a person if we know its position. The tracks which are people detected only by laser have green colour. The numbers are the id’s of each track. Furthermore, we can see the laser and vision detections which are yellow and orange cylinders and we can see the walls detected by the laser which are represented as green points. Also, in the images we can see some tracks which are false positives, due to the miss alignment of some parts of the map with the wall detected by the laser. These miss alignments of the map with the detected walls introduce some errors in the filtering of the false positives detected in the walls.

In the Figures 5.2, we can see the confirmation of the fusion track which takes the time that we present in the Section 5.2.3. Also, we can see how the track is initially a green cylinder which is a track considered as a person only detected by laser. Then, in the next image we can see how the track is a blue cylinder which is a the track considered a person detected by laser and vision (Fusion track). In the Figures we can see the robot models of Tibi and Dabo, Tibi is orange and Dabo is blue. The change of id’s respect the Dabo and Tibi views are because the images has been taken in two different executions of the rosbags, we can not execute the same rosbag offline for the two robots at the same time.
Figure 5.2: Confirmation of a track with fusion in the FME. The green cylinders are tracks without fusion, the blue cylinder is a track with fusion, the numbers are the id’s of the tracks, the yellow cylinders are the laser detections, the orange cylinders are the vision tag detections. Also, we can see the models of Tibi (orange) and Dabo (blue) robots.

In the Figures 5.3 and 5.4, we can see how the person walk around this area and the tracker holds the track with fusion all the time without change its id (eliminate the person) and without change its colour (lose the Fusion track state).

Figure 5.3: Holding the fusion track in the FME. The green cylinder is a track without fusion, the blue cylinder is a track with fusion, the numbers are their id’s, the yellow cylinders are the laser detections, the orange cylinders are the vision tag detections. Also, you can see the models of Tibi (orange) and Dabo (blue).(1)
Figure 5.4: Holding the fusion track in the FME. The green cylinder is a track without fusion, the blue cylinder is a track with fusion, the numbers are their id’s, the yellow cylinders are the laser detections, the orange cylinders are the vision tag detections. Also, you can see the models of Tibi (orange) and Dabo (blue).

In the images of Figure 5.5 we have Tibi very near of the person which has the fusion, roughly a distance of 0.5 meters. As Tibi is considered a person we can see the behaviour of the association distance for the laser and vision detections. In the images regarding the association distance of the laser we do not change the id’s and we do not lose the tracks of the people. In the images regarding association distance of the vision we can distinguish the person detected only by laser and the person detected by laser and vision.

Figure 5.5: Association distance in the FME. The green cylinder is a track without fusion, the blue cylinder is a track with fusion, the numbers are the id’s of the tracks, the yellow cylinders are the laser detections, the orange cylinder are the vision tag detections. Ans also we can see the models of Tibi (orange) and Dabo (blue).
5.3.1.2 Experimental results with the rosbags in the Barcelona robot lab

The Barcelona robot lab is situated in the UPC Nord Campus in Barcelona. The environment is a huge area which has 293.5 m$^2$, spread over: 147,500 m$^2$ of builded surface, 112,000 m$^2$ of free space and 34,000 m$^2$ of garden. The Barcelona robot lab has several levels and underpasses. We can find different static obstacles as columns, banks, ramps, stairs. Also, we can find different unpredictable mobile obstacles which are a lot of people which walk around. We can make the experiments in the different places of the campus nord, we can see the different experiments localizations in the Figure 5.6. We can see the Barcelona robot lab in the Figure 5.7.

![Figure 5.6: Different places of the campus nord for the experiments.](image)

![Figure 5.7: Barcelona robot lab scenario](image)

In the Figures 5.8 we can see the confirmation of the track with fusion. The confirmation time for fusion is configurable and in Subsection 5.2.3 we saw some different values for it. The figures show a confirmation case in the outside place of the classrooms of campus nord, this confirmation is with the parameters which takes 5 seconds in confirm the track.
with fusion in the case that we have a continuous vision detection associated to the track. But in the real case the fusion confirmation takes more time due to the false negatives which we can have. When we have a false negative the probability of the track decreases and then when we have detection associated to the track the probability regrows. Because of this, it is not possible know exactly the time which takes the confirmation of the track since it depends of the detector and the concrete situation. But in all the experiments the confirmation time for the fusion takes less than 10 seconds.

In the Figures 5.9 we can see how the tracker distinguish well the specific person which has the tag, even in situations where the specific person is together with other people and may be partially occluded by them. This good behaviour is done by the vision association distance and the multi-hypothesis of the vision detector. The association distance can be adaptable if it is necessary. In this case, the vision distance is 0.5 meters. This distance is 0.5 meters since the vision detection has less precision in position and we want to do a good Kalman correction with this detector. If we will use a huge distance and we will make the Kalman update we could lose the track. This track loses is because the association with a bad position of the detection which introduces a propagation velocity which is not real. This velocity make a bad track prediction which causes the lose of the track.
In the Figures 5.10 we can see how the tracker is holding the person which has the fusion, even when the tracker has some false positives in vision and also when we have a difficult environment with columns and different people walking around. The tracker does not lose the fusion status of the track and does not lose the track, keeps the track id.
In the Figure 5.11 we have a good example of the tracking performance in the case of the association distance. We can see a group of people and we only have one person who wear the tag. This person goes behind the previous group of two people which do not have tags. Even in this hiding situation the tracker is able to keep the fusion track and follow
it. The robot following behaviour that we can see in the images is recorded previously without the tracker test because this behaviour is performed by other node made by other person. We only tested the fusion behaviour with the people laser detections and the tag detection, we did not test the robot following behaviour.

![Figure 5.11: Group of people and we only have one person with the tag.](image)

### 5.3.2 Real experimental results

When we have the algorithm tested in rosbags and the algorithm was working well. We tested the node in the real robots. First, we tested inside the laboratory. As seemed that the algorithm was working well, then we went to the FME place and tested the algorithm more in detail.

#### 5.3.2.1 Results of real experimental in the laboratory

Inside the laboratory we tested the confirmation, hold and deletion of the track which had fusion. We keep the parameters selected in tests with rosbags since this parameters make a good tracking performance. These parameters are the explained previously in Section 5.2. The tracks were confirmed in a reasonable time, were hold out pretty well and were eliminated inside a reasonable time. Therefore, the algorithm works well. Due to that, after some proves we go out to the FME place.

In the Figures 5.12 we can see the real images of the tests which we made inside the laboratory. The rosbags are not available due to some problems in the record of these rosbags. But, then in the experiments outside we have the rosbags recorded and we can see in more detail the good results obtained in the real tests. Initially in the tests, we placed in front of the robot a person only detected by laser. Then, the person holds or worn the tag in front in a place where the robot can see the tag. A continuation, only for the test, the person made some artificial false negatives of the tag. This can be performed static or dynamic (The person walk around the robot). Finally, the person hide the tag behind him to see the deletion of the fusion state for this track.
5.3.2.2 Experimental results with the real robot in the FME

In the FME experiments to test the tracker, we tested the confirmation, holding and deletion of the tracks and also the association distance in groups of tracks.

First, we made that one person stay in front the robot without the tag, then this person put the tag in front of it, in one position which the robot can detect it. A continuation, the person walked around the robot making some false negatives in artificial way. Also, the person walked some times with the tag in front of it all the time to see the real false negatives situations. Finally, the person hide the tag to see the deletion time of the fusion state. These experiments were satisfactory and we have good results as we can see in Figure 5.13 for the confirmation of the fusion track, in Figure 5.14 for the holding of the track with fusion and in Figure 5.15 for the deletion of the fusion state of the track. The times taken for the confirmation and deletion were the expected taking into account the parameters which we use. These parameters was the explained previously.
In the Figures 5.13 we can see the fusion track confirmation. In the first two images the track is considered a person detected only by laser. Then, in the two next images the person holds the tag in front and the track is considered as a person detected by laser and vision.

Figure 5.14: Test the track fusion holding.
In the Figures 5.14 the person walks around the robot and we can see how the tracker is holding the track id with the laser parameters and it also is holding the fusion state of the track using the vision parameters.

Figure 5.15: Test the track fusion deletion.

In the Figures 5.15 we can see the deletion of the fusion state for the track. The person hide the tag behind it and the track takes the deletion time to change the track status from fusion to only laser detected.

Furthermore, we proved the value of the association distance. Also, we proved how influence the crosses and movements around the other tracks in the association of the track with fusion. We can see the association distance with a real subject in the Figures 5.16. Also, we can see a real group of people when one person is the track with fusion in the Figures 5.17.

Figure 5.16: Association distance when the fusion person is with other people.
Finally, we tested in more detail the association distance. We tested the distance making that the laser detect all obstacles as a people detection and filter the false positives of the walls. With this, we detected a paper bin as if it was a person and we walked around the bin. With this experiment we proved the association distance and how influenced in the association the crosses in front and behind other tracks. When we were behind the bin, we hid the tag as if the tag was occluded by the person which had between the robot and us. If we had more time the tag hid the fusion person status was lose it. But if we detect partially the tag we were holding the fusion status as in the rosbag cases in the campus Nord. Also, the association distance was the expected. We can see more or less the distances between the tracks and the results of this part in the Figures 5.18 and 5.19.
Figure 5.19: Test the track association distance with a bin.(2)
5.4 Experimental results in *Cargo ANTs* project

This experiment has been performed under the *Cargo ANTs* project. To do it, we used a dataset which has been created for testing the algorithms. Real sensor data was obtained from the CTT port terminal in Hengelo, the Netherlands (Fig. 5.20(b)), and has been captured using the TNO test vehicle that included specific sensors for this project (Fig. 5.20(a)). The dataset includes relevant data for *vehicle following* inside the port. For this dataset, a truck has been chosen as target (Fig. 5.20(c)), because *Cargo ANTs* is about mixed situations of AGV’s and automated driving trucks. The sensor setup of the TNO test vehicle is shown below:

- A TNO test vehicle from where the host tracking signals were logged (e.g. longitudinal velocity and acceleration).
- 3 Automotive radars (on top of the vehicle) which outputs are both object clusters and trackers.
- 6 Ibeo LUX laser scanners, mounted on the TNO test vehicle, which capture 360 deg field of view of the environment. The output of the sensor is a pointcloud data.
- Accurate GPS measurement system of OxTS.
- Video camera which captures the front view of the TNO test vehicle.

![Figure 5.20: TNO test vehicle containing all the sensors used, CTT environment and image of the front view with the corresponding output of the algorithm.](image)

5.4.1 Tracker improvements for *Cargo ANTs*

In the implementation for *Cargo ANTs* project we have to differentiate between static and dynamic objects and we can do it by the confirmation and elimination functions in the probabilities (Sections 4.1.1 and 4.1.2) and by filtering the static objects by velocity (Section 5.4.1.1). Also, we have to group different separated detections of the same moving
object due to the fact that the detector can give to the tracker more than one detection of the same object (Section 5.4.1.2). The improvements for Cargo ANTs project and the experiments carried out for this European project are submitted in a paper to the ROBOT’2015 - Second Iberian Robotics Conference.

5.4.1.1 Filter static objects by velocity in a local tracking

In order to filter the static objects, the tracker uses the following general formula:

\[ V_o^w = V_o^v + V_R^w + W_R^w \times r \]  

(5.1)

Where, \( V_o^w \) is the real velocity of the object in the world coordinates; \( V_o^v \) is the estimated local linear velocity of the object in the vehicle coordinates; \( V_R^w \) is the linear velocity of the vehicle in the world coordinates; \( W_R^w \times r \) is the linear velocity in the object induced by the rotation of the vehicle (in which \( W_R^w \) is the angular velocity of the vehicle in the world coordinates and \( r \) is the distance between the vehicle and the tracked object).

By imposing \( V_o^w = 0 \), then \( V_o^v = -(V_R^w + W_R^w \times r) \), and this is used to filter static objects.

5.4.1.2 Group Association by Distance.

The tracks of the same type are included in the same group using the following grouping distance (Eq. 5.2), this equation is the same that uses [28], but changing the detection by a track.

\[ (H\bar{x}_1 - H\bar{x}_2)^T(H\bar{P}_1H^T + H\bar{P}_2H^T)^{-1}(H\bar{x}_1 - H\bar{x}_2) \leq \eta^2 \]  

(5.2)

Where, \( \bar{x}_1 \) and \( \bar{x}_2 \) are the track states and \( P_1 \) and \( P_2 \) are its corresponding covariances; \( H \) is a measurement matrix defined in the same way as in [28]. Here, \( \eta^2 \) have the same meaning that in [28] but with biggest value, because is associating different targets of the same moving object.

When the tracker groups tracks of the same moving object which are detected in a separate way, it takes into account the shape of all of the grouped targets to get the real shape of the moving object. In this case, a new id for the final group is created.
5.4.2 Experimental results

In the Cargo ANTs project, we have only a laser object detector and we are tracking in local way using the vehicle odometry. At this moment, we only have one detector and we can not use the fusion part of the tracker.

With the data set provided, we tested two different configurations of the tracker which corresponded with two configurations of the detector. This two tracker configurations are in the next subsubsections 5.4.2.1 and 5.4.2.2. In both cases there exist a configurable track initialization time which impose a trade-off between performance and initialization time. When we have a longer initialization time we can filter more static objects considered as moving objects. The initialization time is the same as the confirmation time.

5.4.2.1 Tracker in false positives filtering configuration

In this case, the detector sent to the tracker a lot of false positives (static object considered as moving ones) and the tracker had to filter the false positives and tracked only the object which is moving. In this case, we chose a initialization time of 1.5 seconds and the beta confirmation of the detector was 0.01835. For the beta deletion we used a value of 0.98 which gives us a track elimination time of 44.75 seconds. The Figure 5.22 show the results of this case. In the figure we can see The global number of detections in red (static and moving detections), the detections considered as dynamic by the detector are in blue and the final dynamic object tracked is in green. This figure show how the tracker has a initialization time of the moving tracks of 1.5 seconds and how the tracker is able to eliminate all the false positives and keeps track only of the real object. The number of false positives filtered by the tracker are the detections in blue when we have more than one detection considered as moving objects and also the false negatives can be seen in the blue curve when we do not have any detection considered as moving object.
5.4.2.2 Tracker in false negatives filtering configuration

In this case, the detector sent to the tracker less false positives of the no moving objects detected but introduced more false negatives (moving objects not detected). Then, the tracker had to filter less false positives but had to filter more false negatives and keep track of the real moving object. In this case we had a initialization time less of 1 second and used the 0.03 value of the beta confirmation. For the beta deletion a value of 0.99 which gives to us a track elimination time of 89.96 sec equal to 1.48 min. We prefer to keep more time the before moving object detected to avoid collisions even in the cases which the track goes out of the driving area for the vehicle, than eliminate the moving tracks when they still exist. The Figure 5.22 show the results of this case. In the Figure we can see The global number of detections in red (static and moving detections), the detections considered as dynamic by the detector are in blue and the final dynamic object tracked is in green. The Figure show how the tracker have a initialization time of the moving tracks of 1 seconds and how it is able to eliminate all the false positives and the false negatives to keep track only of the real object without lose it. The false positives obtained are the detections in blue when we have more than one considered moving object and also the false negatives can be seen in the blue curve when we do not have any detection considered as moving object.
Robust multi-hypothesis tracker fusing diverse sensor information

In the Figures 5.23 we can see the moving object tracking results in the Rviz visualization tool. In the left image we can see how the tracker filters the false positives and groups the different moving detections contained in the single real moving object. In the right image we can see how the tracker holds the real dynamic object in a case which we do not have a detected object considered as dynamic object.

Figure 5.23: First image: the tracker is filtering static objects and grouping dynamic objects. Second image: the tracker is holding the real dynamic object in false negatives situations. We can see in green the real dynamic track with its identifier (12). Also, the red objects are the detections, the blue ones are the detection considered as a moving objects and the reference frame represents the vehicle.
6. Conclusions and future work

In this final chapter, we present the obtained conclusions of the work. Also, we evaluate the fulfilled objectives of the project. Furthermore, we present some possible future lines of work based on the open issues identified during the project.

6.1 Conclusions

In this work, we present the whole actual tracker of the IRI, but for the TFM we implemented the fusion part of the tracker. Obtaining with all the Robust multi-hypothesis tracker fusing diverse sensor information.

In the tracker implementation we meet all the initial fixed objectives and we obtain a tracker with fusion of different sensor information. We reformulated the before fusion formulation to obtain a good fusion behaviour for us. Also, we implemented a general tracker which can be adaptable to add more and different sensors. Furthermore, we tested the tracker with rosbags and in the robot proving its good behaviour.

With the tracker with fusion we can follow people in a more robust way. Also, we can distinguish specific persons which wear a tag. Furthermore, in the future if we will have a vision detector more precise in position, we could correct well the people position when the track is associated to the vision detector. Finally, we could adapt the tracker behaviour to different fusion situation, since we can change its parameters to obtain different time results in the confirmation and elimination of the tracks which has fusion. Also, we can change its parameters to obtain different association distances between the vision detection and the tracks. The tracker adaptability is very important to work with other nodes which for example has to find a specific person. The tracker adaptability is important since we can adapt the confirmation, elimination and association in fusion, for the necessities of the other algorithms. The tracker confirmation and elimination in the fusion case are very important due to the type of vision detector which we have, this vision detector can have false positives and false negatives, due to that, it is necessary these two functions in the fusion. Also, this detector is very imprecise in position and we need to adapt the association distance in some cases.
In the experiments part, we proved the good behaviour of the tracker in different environments. We saw how the tracker confirms in some time the *Fusion track* when the track had the tag associated. Also, we saw how the tracker removed the state of *Fusion track* in the case which the tag detection was not associated to the person. Also, We saw the importance of the association distance in crossing situations and in proximities with other people which did not have the tag. Furthermore, we saw how the tracker holds the fusion state when the tracker had false negatives. Also, we saw how the tracker distinguished the specific person when the person was in a group or when the person is partially occluded for other people. We distinguished the specific person in some difficult cases with some false positives, by the track type (fusion) and also by the track probability, due to the fact that the true specific people which had the tag, also it had always more probability than the false positives.

### 6.2 Future work

The future lines of research which we found are related with some directions that we could follow in the future of this investigation project. Some of the possible improvements that we can make in the tracker will be the next:

1. We could include some intelligent people interactions in the tracker behaviour.
2. We could implement the direct treatment of the occlusions in the tracker.
3. We could use people predictions more complex using models of people behaviour in the tracker prediction.
4. We could adapt the fusion for the future radar detector and the skeleton detector.
Bibliography


A. Annex

A.1 Coordinate change for local tracking

This part was implemented for an European project, because we have to work in local situations without map, only using the local laser detections and the vehicle/robot odometry. Of course, this part can work for vehicles and for Tibi and Dabo robots.

For the tracking work in local situations, when we receive the group of detections, we have to transform the tracks positions, velocity and covariances from the previous position to the current position of the mobile robot. We have to make a change of coordinate frames in position and orientation. If we not transform the tracks to the new robot position, we can not compare the actual local detections with the tracks that we have. For the implementation of this part, we use the next formulas to translate the tracks positions (Eq. A.2), velocities (Eq. A.3) and covariances (Eq. A.5) to the new position of the robot. Also, we can see graphically this process in the Figure A.1. We can find the theory of this part in [14].

![Figure A.1: Change between coordinate frames](image)

Translation of the Robot base frame: $R^t = v^t \cdot \Delta t$. Where, $R$ is the robot translation, $v$ is the linear velocity, $\Delta t$ is the interval of time between detections and $t$ is the current time.
Rotation of the Robot base frame: $\theta^t = w^t \cdot \Delta t$. Where, $\theta$ is the robot rotation, $w$ is the angular velocity, $\Delta t$ is the interval of time between detections and $t$ is the current time.

$$[H] = \begin{bmatrix} 
\cos(\theta^t) & -\sin(\theta^t) & R^t \cos(\theta^t/2) \\
\sin(\theta^t) & \cos(\theta^t) & R^t \sin(\theta^t/2) \\
0 & 0 & 1 
\end{bmatrix} \quad (A.1)$$

Transformation of the track position:

$$\begin{bmatrix} 
x \\
y \\
1 
\end{bmatrix}^t = [H^{t,t-1}]^{-1} \cdot \begin{bmatrix} 
x \\
y \\
1 
\end{bmatrix}^{t-1} \quad (A.2)$$

Where $x$ and $y$ are the track position in $x$ and $y$, $H$ is the transformation matrix, $t$ is the current time and $t - 1$ is the previous time.

Track velocity transformation:

$$\begin{bmatrix} 
v_x \\
v_y \\
0 
\end{bmatrix}^t = [H^{t,t-1}]^{-1} \cdot \begin{bmatrix} 
v_x \\
v_y \\
0 
\end{bmatrix}^{t-1} \quad (A.3)$$

Where $v_x$ and $v_y$ are the track velocities, $H$ is the transformation matrix, $t$ is the current time and $t - 1$ is the previous time.

Transformation of the track covariance:

$$[H] = \begin{bmatrix} 
\cos(\theta^t) & \sin(\theta^t) & 0 & 0 \\
-\sin(\theta^t) & \cos(\theta^t) & 0 & 0 \\
0 & 0 & \cos(\theta^t) & \sin(\theta^t) \\
0 & 0 & -\sin(\theta^t) & \cos(\theta^t) 
\end{bmatrix} \quad (A.4)$$

$$P^t = H \cdot P^{t-1} \cdot H^T + J \cdot O \cdot J^T \quad (A.5)$$

$$O = \begin{bmatrix} 
\sigma_R^2 & 0 \\
0 & \sigma_{\theta}^2 
\end{bmatrix} \quad (A.6)$$

$$[J] = \begin{bmatrix} 
-\cos(\theta^t) & -\sin(\theta^t) \cdot x^{t-1} + \cos(\theta^t) \cdot y^{t-1} + R^t \cdot \sin(\theta^t) \\
\sin(\theta^t) & -\cos(\theta^t) \cdot x^{t-1} - \sin(\theta^t) \cdot y^{t-1} + R^t \cdot \cos(\theta^t) \\
0 & -\sin(\theta^t) \cdot v_x^{t-1} + \cos(\theta^t) \cdot v_y^{t-1} \\
0 & -\cos(\theta^t) \cdot v_x^{t-1} - \sin(\theta^t) \cdot v_y^{t-1} 
\end{bmatrix} \quad (A.7)$$