

Research Article

Trajectory Planning for Nonholonomic Mobile Robot Using Extended Kalman Filter

Leonimer Flavio de Melo and Jose Fernando Mangili Junior

Department of Electrical Engineering, State University of Londrina, Londrina, PR, Brazil

Correspondence should be addressed to Leonimer Flavio de Melo, leonimer@uel.br

Received 26 August 2010; Revised 10 November 2010; Accepted 28 December 2010

Academic Editor: Miguel A. F. Sanjuán

Copyright © 2010 L. F. de Melo and J. F. M. Junior. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the mobile robotic systems, a precise estimate of the robot pose with the intention of the optimization in the path planning is essential for the correct performance, on the part of the robots, for tasks that are destined to it. This paper describes the use of RF digital signal interacting with beacons for computational triangulation in the way to provide a pose estimative at bidimensional indoor environment, where GPS system is out of range. This methodology takes advantage of high-performance multicore DSP processors to calculate ToF of the order about ns. Sensors data like odometry, compass, and the result of triangulation Cartesian estimative, are fused for better position estimative. It uses a mathematical and computational tool for nonlinear systems with time-discrete sampling for pose estimative calculation of mobile robots, with the utilization of extended Kalman filter (EKF). A mobile robot platform with differential drive and nonholonomic constraints is used as a base for state space, plants and measurements models that are used in the simulations and validation of the experiments.

1. Introduction

The planning of trajectory for the mobile robots, and consequently its better estimative of positioning, is the reason of intense scientific inquiry. A good path planning of trajectory is fundamental for optimization of the interrelation between the environment and the mobile robot. A great diversity of techniques based on different physical principles exists and different algorithms for the localization and the planning of the best possible trajectory.

The localization in structuralized environment is helped, in general, by external elements that are called of markers. It is possible to use natural markers that already existing in the environment for the localization. Another possibility is to add intensionally to the environment artificial markers to guide the localization of the robot.

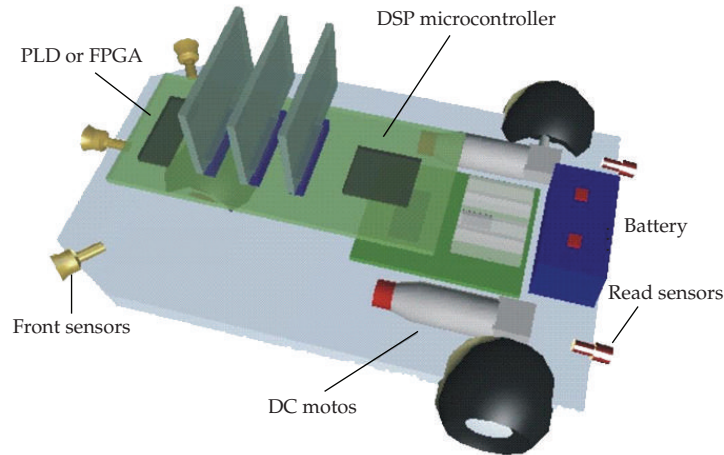


Figure 1: Mobile robot platform and elements.

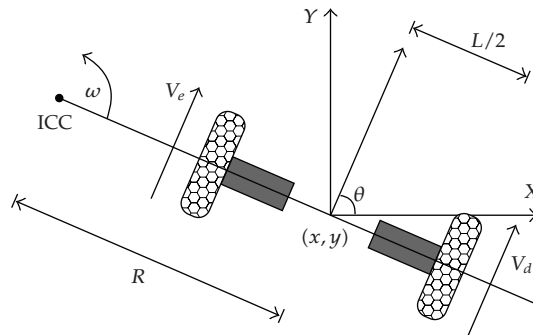


Figure 2: Direct kinematics for differential traction in mobile robots.

This work uses an important mathematical and computational tool for the calculation of the data fusing collected by the sensors and the disturbances caused for the errors, with the purpose of estimate the mobile robot pose, that is the Extended Kalman filter (EKF). A mobile robot platform with differential traction and nonholonomics restrictions is used for experiments validation.

In the direct kinematics the system of mobile robot positioning is presented. A model for state space, plants and measurements are presented, that are needed for the development of the necessary attributes to the positioning estimates made with the Extended Kalman filter. Finally, we present the experimental and simulation results obtained from the models created.

2. Direct Kinematics for Differential Traction

This paper focuses on the study of the mobile robot platform, with differential driving wheels mounted on the same axis and a free castor front wheel, whose prototype used to validate the proposal system is depicted in Figures 11 and 1 which illustrate the elements of the platform.

We assume that the robot is in one certain point (x, y) directed for a position throughout a line making an angle θ with x axis, as illustrated in Figure 2.

Through the manipulation the control parameters v_e and v_d , the robot can be led at different positionings. The determination of the possible positionings to be reached, once given the control parameters, is known as direct kinematics problem for the robot. As illustrated in Figure 2, in which the robot is located in position (x, y, θ) , we have for the trigonometrical relations of the system

$$\text{ICC} = [x - R \sin(\theta), y + R \cos(\theta)], \quad (2.1)$$

where ICC is the robot instantaneous curvature center.

As v_e and v_d are time functions and if the robot is in the pose (x, y, θ) in the time t , and if the left and right wheel has ground contact speed v_e and v_d , respectively, then, in the time $t \rightarrow t + \delta t$ the position of the robot is given by

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \delta t) & -\sin(\omega \delta t) & 0 \\ \sin(\omega \delta t) & \cos(\omega \delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - \text{ICC}_x \\ y - \text{ICC}_y \\ \theta \end{bmatrix} + \begin{bmatrix} \text{ICC}_x \\ \text{ICC}_y \\ \omega \delta t \end{bmatrix}. \quad (2.2)$$

Equation (2.2) describes the motion of a robot rotating a distance R about its ICC with an angular velocity given by ω [1]. Different classes of robots will provide different expressions for R and ω [2].

The forward kinematics problem is solved by integrating (2.2) from some initial condition (x_0, y_0, θ_0) , it is possible to compute where the robot will be at any time t based on the control parameters $v_e(t)$ and $v_d(t)$. For the special case of a differential drive vehicle, it is given by

$$\begin{aligned} x(t) &= \frac{1}{2} \int_0^t [v_d(t) + v_e(t)] \cos[\theta(t)] dt, \\ y(t) &= \frac{1}{2} \int_0^t [v_d(t) + v_e(t)] \sin[\theta(t)] dt, \\ \theta &= \frac{1}{L} \int_0^t [v_d(t) - v_e(t)] dt. \end{aligned} \quad (2.3)$$

A question more interesting, and at same time more difficult to answer, is the following how can the control parameters could be selected in a way the robot obtain a specific global position or follow a specific trajectory? This is known as the task of determining the vehicle's *inverse kinematics*: inverting the kinematic relationship between control inputs and behavior. It is also related to the problem of trajectory planning.

2.1. Inverse Kinematics for Differential Drive Robots

Equation (2.3) describe a constraint on the robot velocity that cannot be integrated into a positional constraint. This is known as a *nonholonomic constraint* and it is in general very difficult to solve, although solutions are straightforward for limited classes of the control

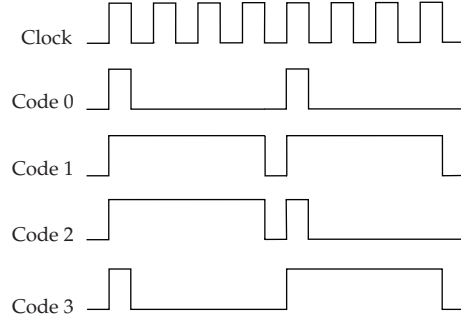


Figure 3: Communication protocol codes.

functions $v_e(t)$ and $v_d(t)$ [3]. For example, if it is assumed that $v_e(t) = v_e$, $v_d(t) = v_d$ and $v_e \neq v_d$, then (2.3) yields

$$\begin{aligned}
 x(t) &= \frac{L}{2} \frac{v_d + v_e}{v_d - v_e} \sin \left[\frac{t}{L} (v_d - v_e) \right], \\
 y(t) &= -\frac{L}{2} \frac{v_d + v_e}{v_d - v_e} \cos \left[\frac{t}{L} (v_d - v_e) \right] + \frac{L}{2} \frac{v_d + v_e}{v_d - v_e}, \\
 \theta(t) &= \frac{t}{L} (v_d - v_e),
 \end{aligned} \tag{2.4}$$

where $(x, y, \theta)_{t=0} = (0, 0, 0)$. Given a goal time t and goal position (x, y) . Equation (2.4) solves for v_d and v_e but does not provide a solution for independent control of θ . There are, in fact, infinity solutions for v_d and v_e from (2.4), but all correspond to the robot moving about the same circle that passes through $(0,0)$ at $t = 0$ and (x, y) at $t = t$; however, the robot goes around the circle different numbers of times and in different directions.

3. Position Estimation with RF Signal ToF

The communication system between the mobile robot and the beacons is follow described. The mobile robot, and each one of the beacons, have a module of control and reception of the address codes and a module of transmission. The communication protocol between the embedded control system, located in the mobile robot, and the beacons, that are located in strategical points in the environment, are composed of a frame formed for five quaternary codes.

3.1. Communication Protocol

The timing diagram shown in Figure 3 illustrates as each one of the codes in function of clock signal is formed.

Each half clock period correspond to a time about $896 \mu s$. Each code has a time period composed of 8 clocks cycles, that is $14,336 \text{ ms}$. Table 1 depicts in a logic way the formation of the codes.

Table 1: Logic formation of each code.

Code	Logic sequency
C0	$A_C + B_L + A_C + B_L$
C1	$A_L + B_C + A_L + B_C$
C2	$A_L + B_C + A_C + B_L$
C3	$A_C + B_L + A_L + B_C$

Table 2: The timing of the logic codes.

Logic signal	Meaning	Duration	Time
A_C	Logic level "1" with short duration	(1/2) clock	0,896 ms
A_L	Logic level "1" with long duration	3(1/2) clock	6,272 ms
B_C	Logic level "0" with short duration	(1/2) clock	0,896 ms
B_L	Logic level "0" with long duration	3(1/2) clock	6,272 ms

Each code is configured by a logic signals sequence, each one with a determined period. Table 2 shows how each logic signal of each code is composed.

The idea is to mount a quaternary codifier using binary logic levels, associates in such way that the logic levels alternate and the total period of each code is the same. The codification implemented was conceived here aiming the minimizing of the errors, such as the transmission of the one exactly signal level is transmitted without transitions of level for long time periods. In this case, the receiver tends to put out of the way itself and to perform the reading out the correct point, originating errors. In this way, RF transmission of the codes is sufficiently robust and trustworthy, practically extinguishing errors of signal decoding signal inside the area of system range.

3.2. The Communication Frame

The communication frames used between the mobile robot and the beacons are composed for five quaternary codes. The Figure 4 illustrates an example of a communication protocol frame. As each code has a period about 14,336 ms, the all frame has transmission time about 71,68 ms.

The maximum number of possible combination is given by

$$N_e = 4^5 = 1024. \quad (3.1)$$

Each beacon has it own address, composed by five codes. In this way, the system is able to deal with up to 1024 beacons, with their own individual address.

3.3. The RF Link

The coded signal is transmitted in RF modulated by BASK-OOK technic. The carrier signal frequency is about 433,92 MHz (UHF band).

The RF link uses a half-duplex channel between mobile robot and beacons. The mobile robot control system is previously programmed with quantity and address of each beacon.

C2	C0	C1	C3	C0
----	----	----	----	----

Figure 4: Example of a communication protocol frame.

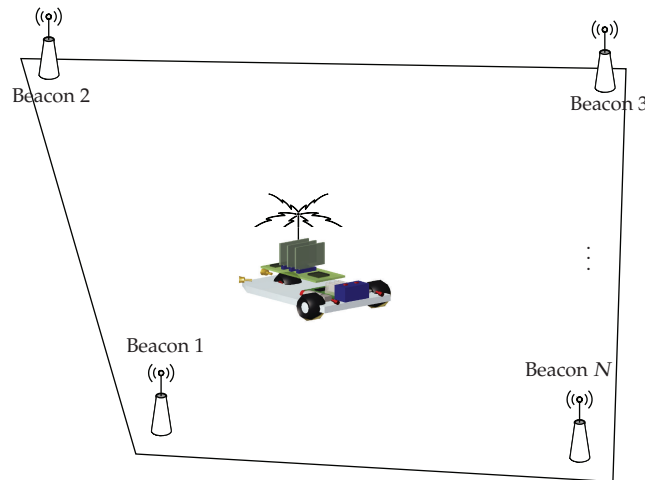


Figure 5: An example of environment beacons arrangement and the communication system.

Figure 5 depicts an example of environment configuration of the communication between the robot and beacons.

3.4. Beacon Transceiver System

The beacon embedded system is composed basically by two modules. One is responsible for RF signal receive and make all the concerned computation. This module has a 16F630 PIC microcontroller, operating at 4 MHz clock frequency. The other one is the RF signal transmitter. This module is equipped with 12F635 microcontroller and also operates at 4 MHz. The system is able to operate in autonomous way, been programmed with specific address. In other hand, the mobile robot must be programmed with the amount and the address of all operative beacon inside the navigating environment. Figure 6 depicts the block diagram of the RF transceiver at mobile robot and at beacons.

Figure 7 shows the beacons RF transceiver modules. The transmission module (Figure 7(b)) is able to function in asynchronous independent way, emitting a address code frame in a certain period of predetermined time, or synchronous way commanded by the reception and control module (Figure 7(a)). In the first case, a battery 12 V A23 model is used which allows autonomy of more than 3 months of continuous use, due to ultra low power energy consumption given by the embedded microcontroller with nanowatt technology. In second case the power supply and transmission command are made by the reception control module, illustrated in Figure 7(a). This second one is the mode utilized by this work.

The mobile robot, as each one of the beacons, have a transceiver control system composed by reception module and transmission module. As the objective of our system is to provide a triangulation between the mobile robot position and the beacons, the transmission

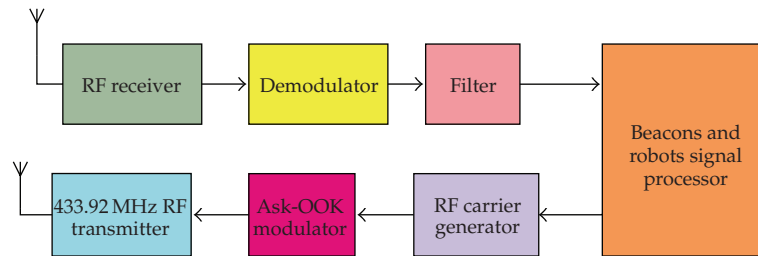
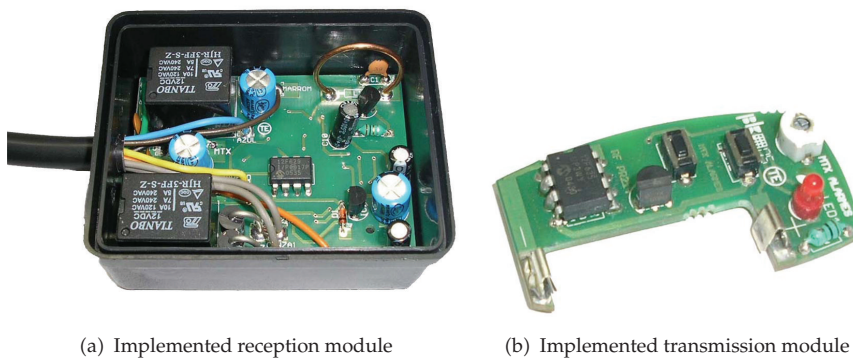


Figure 6: Block diagram of the RF transceiver at mobile robot and at beacons.



(a) Implemented reception module

(b) Implemented transmission module

Figure 7: Beacons RF transceiver modules.

modules work in synchronous way. It is assumed that the module of control of reception-transmission of the mobile robot has been previously loaded with the amount of existing beacons in the environment and with its respective addresses codes. The functioning of the system goes to the following procedure.

- (1) The mobile robot emits a address code-frame for first beacon. In this instant it sends an interrupt control signal to the central processing unit for triggering and starts a timer counter. The robot then, waits the return of the signal. This return must occur in up to 100 ms.
- (2) If the signal returns, means that the beacon recognized the code and sent back the same code. In this instant is sent a signal to the robot embedded central processing unit for stops the timer and calculation of the signal return delay time, that could be about ns.
- (3) If the signal was not returned, means that the beacon is out of area reach or occurred some error in signal transmission-reception.
- (4) Increment the number of beacon and go to the loop first item.

The distance between the robot and a certain m beacon is computed with base of the delay time in the reception of the same transmitted code. The total elapsed time between the code final transmission, sent by the robot, and the reception of the same code, sent back to the beacon, can be calculated by

$$T = t_{is} + t_{pm} + t_{rs} + t_q + t_{pr}, \quad (3.2)$$

where t_{is} is the travel signal time between leaves robot transmitter and reach beacon reception, t_{pm} is the processing signal time by the beacon, t_{rs} is the signal return elapse time, t_q is the frame code period and t_{pr} is the processing time of the sent back signal received by robot.

It is well known that RF signal cover one meter in about 3,3 ns because its velocity is about 0,3 m/ns in air. We can considering that the linear speed of the robot is so small that the displacement of the robot could be considering as being zero during the time T . In this way, the distance in meters between the mobile robot and the beacon m can be given by

$$d_m = \frac{0,3(t_{is} + t_{rs})}{2}, \quad (3.3)$$

where t_{is} and t_{rs} are given in ns.

The elapsed time T is computed with a 64 bits timer of the Texas Instrument TMS320C6474-1200 dualcore robot embedded processor. The instruction cycle time of it is about 0,83 ns (1,2 GHz clock Device), allowing timer calculations in order of ns, essential for our case of study. The times t_{pm} and t_{pr} are determined empirically and $t_q = 71,68$ ms. In this way, the covered distance between the robot and beacon m should be done by

$$d_m = \frac{0,3(T - t_{pm} - t_{rm} - t_q)}{2}. \quad (3.4)$$

Algorithm 1 depicts the computation method for distance d calculation using RF ToF.

4. Triangulation

Triangulation refers to the solution of constraint equations relating the pose of an observer to the positions of a set of landmarks. Pose estimation using triangulation methods from known landmarks has been practiced since ancient times and was exploited by the ancient Romans in mapping and road construction during the Roman Empire.

The simplest and most familiar case that gives the technique its name is that of using *bearings* or distance measurements to two (or more) landmarks to solve a planar positioning task, thus solving for the parameters of a triangle given a combination of sides and angles. This type of position estimation method has its roots in antiquity in the context of architecture and cartography and is important today in several domains such as survey science. Although a triangular geometry is not the only possible configuration for using landmarks or beacons, it is the most natural [1].


```

input: The mobile robot is initialized with total number of active beacons
         in the environment ( $N_m$ ) and theirs respective address.
output: The distance between the robot and the beacon  $m$  based at delay RF signal time.
System setup;
while The system is active do
  for  $n = 1$  until  $n = N_m$  do
    To transmit a frame-coded for beacon  $n$ ;
    To initiate timer for period  $T$  calculation;
    if The same frame-coded signal returns.AND.  $t < 100$  ms then
      Stops the timer and calculate the time  $T$ ;
      Calculate the distance  $d_m = 0,3(T - t_{pm} - t_{rm} - t_q)/2$ ;
    else
      Some transmission/reception RF signal error occurred;
      Try next active beacon;
    end
  end
end

```

Algorithm 1: Computational method for distance d_m calculation using RF ToF.

Although landmarks, beacons and robots exist in a three-dimensional world, the limited accuracy associated with height information often results in a two-dimensional problem in practice; elevation information is sometimes used to validate the results. Thus, although the triangulation problem for a point robot should be considered as a problem with six unknown parameters (three position variables and three orientation variables), more commonly the task is posed as a two-dimensional (or three-dimensional) problem with two-dimensional (or three-dimensional) landmarks [4].

Depending on the combinations of sides (S) and angles (A) given, the triangulation problem is described as “side-angle-side” (SAS), and so forth. All cases permit a solution except for the AAA case in which the scale of the triangle is not constrained by the parameters. In practice, a given sensing technology often returns either an angular measurement or a distance measurement, and the landmark positions are typically known. Thus, the SAA and SSS cases are the most commonly encountered. More generally, the problem can involve some combination of algebraic constraints that relate the measurements to the pose parameters. These are typically nonlinear, and hence a solution may be dependent on an initial position estimate or constraint [5]. This can be formulated as

$$\mathbf{x} = F(m_1, m_2, \dots, m_n), \quad (4.1)$$

where the vector \mathbf{x} expresses the pose variables to be estimated (normally, for 2D cases $[x \ y \ \theta]$), and $\mathbf{m} = m_1, m_2, \dots, m_n$ is the vector of measurements to be used. In the specific case of estimating the position of an oriented robot in the plane, this becomes

$$\begin{aligned} x &= F_1(m_1, m_2, \dots, m_n), \\ y &= F_2(m_1, m_2, \dots, m_n), \\ \theta &= F_3(m_1, m_2, \dots, m_n). \end{aligned} \quad (4.2)$$

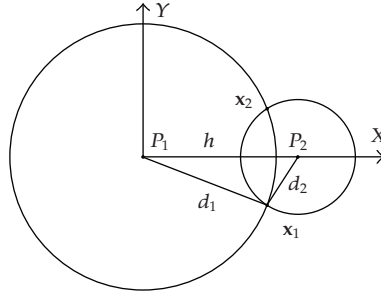


Figure 8: Simple triangulation example. A robot at an unknown location x_1 .

If only the distance to a landmark is available, a single measurement constrains the robot's position to the arc of a circle. Figure 8 illustrates perhaps the simplest triangulation case. A robot at an unknown location x_1 senses two beacons P_1 and P_2 by measuring the distances d_1 and d_2 to them. This corresponds to our case of study in which active beacons at known locations emit a signal and the robot obtains distances based on the time delay to arrive at the robot. The robot must lie at the intersection of the circle of radius d_1 with center at P_1 , and with the circle or radius d_2 with center at P_2 . Without loss of generality we can assume that P_1 is at the origin and that P_2 is at $(h, 0)$. Then we have

$$\begin{aligned} d_1^2 &= x^2 + y^2, \\ d_2^2 &= (h - x)^2 + y^2. \end{aligned} \tag{4.3}$$

A small amount of algebra results in

$$\begin{aligned} x &= \frac{h^2 + d_1^2 - d_2^2}{2h}, \\ y &= \pm \sqrt{d_1^2 - x^2} = \pm \sqrt{d_1^2 - \left[\frac{h^2 + d_1^2 - d_2^2}{2h} \right]^2}, \end{aligned} \tag{4.4}$$

resulting in two solutions x_1 and x_2 .

In a typical application, beacons are located on walls, and thus the spurious (in our example, the x_2) solution can be identified because it corresponds to the robot's being located on the wrong side of (inside) the wall.

Although distances to beacons provide a simple example of triangulation, most sensors and landmarks result in more complex situations [6]. The situation for two beacons is illustrated in Figure 9(a). The robot senses two known beacons and measures the bearing to each beacon relative to its own straightahead direction. This obtains the difference in bearing between the directions to the two beacons and constrains the true position of the robot to lie on that portion of the circle shown in Figure 9(a). We can note that the mathematics admits

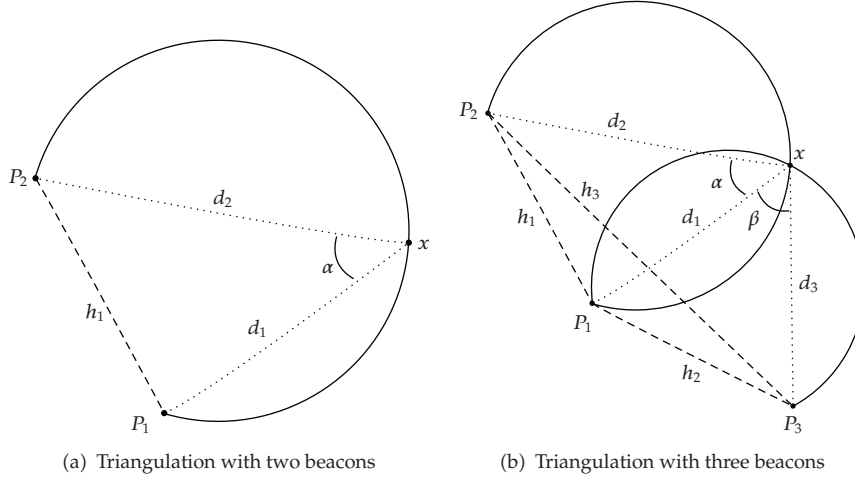


Figure 9: Location estimative of the mobile robot based on beacons triangulation.

two circular arcs, but one can be excluded based on the left-right ordering of the beacon directions. The loci of points that satisfy the bearing difference is given by

$$h_1 = \sqrt{d_1^2 + d_2^2 - 2|d_1||d_2|\cos\alpha}, \quad (4.5)$$

where d_1 and d_2 are the distances from the robot's current position x to beacons P_1 and P_2 , respectively. The visibility of a third beacon, as can be seen at Figure 9(b), gives rise to three nonlinear constraints on d_1 , d_2 and d_3 :

$$\begin{aligned} h_1 &= \sqrt{d_1^2 + d_2^2 - 2|d_1||d_2|\cos\alpha}, \\ h_2 &= \sqrt{d_1^2 + d_3^2 - 2|d_1||d_3|\cos\beta}, \\ h_3 &= \sqrt{d_2^2 + d_3^2 - 2|d_2||d_3|\cos(\alpha + \beta)}, \end{aligned} \quad (4.6)$$

which can be solved using standard techniques to obtain d_1 , d_2 , and d_3 . Knowledge of d_1 , d_2 and d_3 leads to the robot's position [7].

The geometric arrangement of beacons with respect to the robot observer is critical to the accuracy of the solution. A particular arrangement of beacons may provide high accuracy when observed from some locations and low accuracy when observed from others. For example, in two dimensions a set of three collinear beacons observed with a bearing measuring device can provide good positional accuracy for triangulation when viewed from a point away from the line joining the beacons (e.g., a point that forms an equilateral triangle with respect to the external beacons). On the other hand, if the robot is located on the line joining the beacons, the position can only be constrained to lie somewhere on this (infinite) line.

4.1. Triangulation with RF Beacons

In our case of study the beacons's position at 2D environment are known and thus, the distances between the beacons. If the N beacons are positioned at points $(x_{s1}, y_{s1}), (x_{s2}, y_{s2}), \dots, (x_{sN}, y_{sN})$ and the robot's position is given by $\mathbf{x} = (x, y)$, then, (4.6), that express the robot triangulation with three beacons, yield

$$\begin{aligned} d_1^2 &= (y - y_{s1})^2 + (x - x_{s1})^2, \\ d_2^2 &= (y - y_{s2})^2 + (x - x_{s2})^2, \\ d_3^2 &= (y - y_{s3})^2 + (x - x_{s3})^2, \end{aligned} \quad (4.7)$$

where the robot's position $\mathbf{x} = (x, y)$ can be inferred by numerical methods.

5. Data Fusion

The question of how to combine data of different sources generates a great quantity of research in the academics ambients and at the research laboratories. In the context of the mobile robotic systems, the data fusing must be effected in at least three distinct fields: arranging measurements of different sensors, different positions, and different times.

5.1. State Space Models

A system for which the state vector can be fully determined from enough number of measurements is described as being *perceivable*. As used by Bar-Shalom et al. [8], to describe the estimate of state to be computed, is used $\hat{\mathbf{x}}(k | k_1)$ to denote the estimate of the vector \mathbf{x} in the time step k using collected data in a period of time where k is including the time step k_1 . Using remarks until step k , however abstaining k , to form a prediction, this is in general expressed as $\tilde{\mathbf{x}}(k)$ which denotes *prediction* of state vector $\mathbf{x}(k)$ based in the information availability strict before the time k . Being based on the availability of information until, and including themselves, the time k , it forms an updated date state *estimated* $\hat{\mathbf{x}}(k)$, which denotes the *estimative* of the state vector $\mathbf{x}(k)$ in the time k .

5.2. Plant Model

A plant model that describes how a state of the system $\mathbf{x}(k)$, which in ours particular case represents the position of the mobile robot, changes in function of the time, control input $\mathbf{u}(k)$ and the noise $\mathbf{v}(k)$ can be expressed by

$$\mathbf{x}(k + 1) = \Psi(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{v}(k), \quad (5.1)$$

where $\Psi(\mathbf{x}(k), \mathbf{u}(k))$ is the state transition function and $\mathbf{v}(k)$ is the noise function. One of most common and conventional way to represent the noise model is using the Gaussian noise model with zero average with covariance $\mathbf{Q}(k)$ (Gelb also utilizes the notation

$\mathbf{v}(k) \sim N(0, \mathbf{Q}(k))$ to represent the Gaussian noise model with zero average with covariance $\mathbf{Q}(k)$ [9].

A model of linear plant, from (5.1), can be written as

$$\mathbf{x}(k+1) = \mathbf{\Psi}\mathbf{x}(k) + \mathbf{\Upsilon}\mathbf{u}(k) + \mathbf{v}(k), \quad (5.2)$$

where the matrix $\mathbf{\Psi}$ tells how the system evolves from a state to another one in the absence of entrances (that is frequently given as a identity matrix) and the matrix $\mathbf{\Upsilon}$ tells how the entrances of the control modifies the state of the system.

Considering a linear omnidirectional mobile robot with displacements restricted to the plan, a simple model of plant can be considered, as follow presented.

The robot state is given by $\mathbf{x}(k) = [x(k) \ y(k)]^T$ where $(x(k), y(k))$ describes the robot pose at global coordinates system. As suppose that the robot is equipped with some omnidirectional locomotion system, then the control input $\mathbf{u}(k) = [\Delta x \ \Delta y]^T$ can be described as an independent change in the robot x and y localization. If the error in the movement of the robot is independent in the directions x and y , and if this error can be modeled by some noise function $v_x(k)$ and $v_y(k)$, then the robot plant model is given by (5.1), and at ours particular case becomes

$$\begin{bmatrix} x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + \Delta x(k) + v_x(k) \\ y(k) + \Delta y(k) + v_y(k) \end{bmatrix}, \quad (5.3)$$

which the robot moves to where it will be commanded with each movement being corrupted by the noise process. Equation (5.3) describes a model of linear plant.

5.3. Measurement Model

The measurement model describes how the sensors data change in function of the system state. If the sensor model is inverting (hypothetical case) it will allows that the sensor data can be used for state calculations. As explained for Leonard and Durrant-Whyte [10], which developed works with mobile robots equipped with ultrasonic sensors, the measurement model tells the sensor observation for robot position and the target geometry (bulkhead) that produces the observation, and it has the form of

$$\mathbf{z}_i(k) = \mathbf{h}_{st}(\mathbf{x}(k), \mathbf{p}_t) + \mathbf{w}_i(k), \quad (5.4)$$

where $\mathbf{w}_i(k)$ is the noise function for $\mathbf{w}_i(k) \sim N(0, \mathbf{R}_i(k))$ so that represents the Gaussian noise model with zero average with covariance matrix $\mathbf{R}_i(k)$, and \mathbf{p}_t is the target state vector and change accordingly the aim shape, that can be basically corners, edges, cylindrical surfaces or plain surfaces. The measurement function $\mathbf{h}_{st}(\mathbf{x}(k), \mathbf{p}_t)$ express a observation $\mathbf{z}(k)$ about the sensor s for the target t with a vehicle localization function $\mathbf{x}(k)$ and the target geometry.

As well as in the plant model, a linear measurement model is particularly interesting. From (5.4), this takes the form of

$$\mathbf{z}_i(k) = \Lambda_E \mathbf{x}(k) + \mathbf{w}_i(k), \quad (5.5)$$

where Λ_E is the matrix which express how the measurements are derived with the state linear transformation. This simple case illustrate how a stare estimative cam be recover for measurement:

$$\hat{\mathbf{x}}(k) = \Lambda_E^{-1} \mathbf{z}_i(k), \quad (5.6)$$

if it is assumed the Λ_E matrix is reversibly.

The majority of the mobile robots traction by wheels (MRTW) cannot be modeled in a linear way, and it is necessary to consider plant, model and not linear estimative processes, as ours case. Considering a nonlinear system, where the robot can be modeled as a punctual robot with independent control of the orientation and the speed, as the case of the synchronous transmission mobile robots. Then, for this type of robot, the control input $\mathbf{u}(k) = [T(k) \ \Delta\theta(k)]^T$ that is, in the period k until $k + 1$ the robot moves from a distance $T(k)$ forward, to direction that it is pointed, and then rotate itself $\Delta\theta(k)$. The system state is given by $\mathbf{x}(k) = [x(k) \ y(k) \ \theta(k)]^T$ and the nonlinear plant model is given by

$$\Psi(\mathbf{x}(k), \mathbf{u}(k)) = \begin{bmatrix} x(k) + T(k) \cos[\theta(k)] \\ y(k) + T(k) \sin[\theta(k)] \\ \theta(k) + \Delta\theta(k) \end{bmatrix}. \quad (5.7)$$

Each movement of the robot has a noise process parcel $\mathbf{v}(k)$, which has a covariance matrix known or estimable $\mathbf{Q}(k)$. It is assumed for this process of noise that the same satisfies the necessary conditions that assures the use of the Kalman filter(presented in Section 5.4). If, in the practical way, the robot moves in distinct steps composites of pure rotations or pure translations (i.e., only one between $\Delta\theta(k)$ and $T(k)$ is different of zero), then only two versions of $\mathbf{Q}(k)$ are necessary.

Assuming now that the robot is equipped with a sensor that can determine the distance from robot to one determined target marker in the environment. For example, the target can emit an only sound with a known frequency, and the robot is equipped with a receiver that captures the sound. If the robot and the sound emitted at the target have synchronized clocks, the distance between the target and the robot can be estimated. If the target is located at (x_s, y_s) , measurement model for this robot comes from (5.4) and is given by

$$\mathbf{z}_1(k) = \sqrt{[x(k) - x_s]^2 + [y(k) - y_s]^2} + \mathbf{w}_1(k). \quad (5.8)$$

This model of measurement has the parcel of degradation given by the noise process $\mathbf{w}_1(k)$ with covariance matrix $\mathbf{R}_1(k)$.

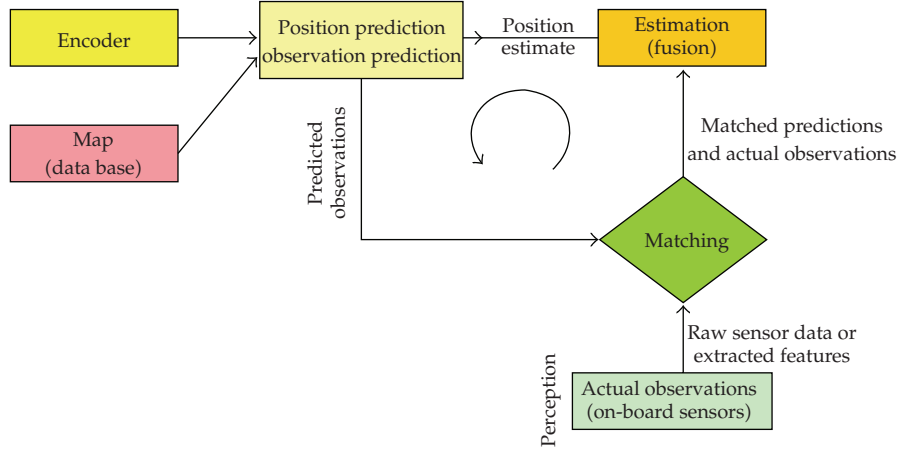


Figure 10: Schematic for Kalman filter mobile robot localization.

5.4. Kalman Filter

To control a mobile robot, frequently it is necessary to combine information of multiple sources. The information that comes from trustworthy sources must have greater importance about those one collected by less trustworthy sensors.

A general way to compute the sources that are more or less trustworthy and which weights must be given to the data of each source, making a weighed pounder addition of the measurements, are known with Kalman filter [11]. It is one of the methods more widely used for sensorial fusing in mobile robotics applications [12]. This filter is frequently used to combine data gotten from different sensors in a statistical optimal estimate. If a system can be described with a linear model and the uncertainties of the sensors and the system can be modeled as white Gaussian noises, then the Kalman filter gives a optimal estimate statistical for the casting data. This means that, under certain conditions, the Kalman filter is able to find the best estimative based on correction of each individual measure [13]. Figure 10 depicts the particular schematic for Kalman filter localization [14].

The Kalman filter consists of the stages follow presented in each time step, except for the initial step. It is assumed, for model simplification, that the state transition matrix Ψ and the observation function Λ_E remain constant in function of the time. Using the plant model of (5.2) and computing a system state estimate in the time $(k + 1)$ based on robot position knowledge in the instant of time k , we have how the system evolves in the time with the input control $\mathbf{u}(k)$:

$$\hat{\mathbf{x}}(k + 1) = \Psi\hat{\mathbf{x}}(k) + \Upsilon\mathbf{u}(k). \quad (5.9)$$

In some practical equations the input $\mathbf{u}(k)$ is not used. It can also, to actualize the state certainty as expressed for the state covariance matrix $\mathbf{P}(k + 1 | k)$ through the displacement in the time, as:

$$\mathbf{P}(k + 1 | k) = \Psi\mathbf{P}(k)\Psi^T + \mathbf{Q}(k). \quad (5.10)$$

Equation (5.10) express the way which the system state knowledge gradually decays with passing of the time, in the absence of external corrections. The Kalman gain can be expressed as

$$\mathbf{K}(k+1) = \mathbf{P}(k+1)\mathbf{\Lambda}_E^T\mathbf{R}_i^{-1}(k+1), \quad (5.11)$$

but, how it did not compute $\mathbf{P}(k+1)$, this can be computed by

$$\mathbf{K}(k+1) = \mathbf{P}(k+1 | k)\mathbf{\Lambda}_E^T \left[\mathbf{\Lambda}_E\mathbf{P}(k+1 | k)\mathbf{\Lambda}_E^T + \mathbf{R}_i(k+1) \right]^{-1}. \quad (5.12)$$

Using this matrix, a estimate of revised state can be calculated that includes the additional information gotten by the measurement. This involves the comparison of the current sensors data $z(k+1)$ with the data of the foreseen sensors using it state estimate. The difference between the two terms

$$\mathbf{r}(k+1) = \mathbf{z}(k+1) - \mathbf{h}_{st}(\hat{\mathbf{x}}(k+1 | k), \mathbf{p}_t), \quad (5.13)$$

or, at the linear case

$$\mathbf{r}(k+1) = \mathbf{z}(k+1) - \mathbf{\Lambda}_E\hat{\mathbf{x}}(k+1 | k) \quad (5.14)$$

is related as the innovation. If the state estimate is perfect, the innovation must be not zero only which the sensor noise. Then, the state estimate actualized is given by

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k+1 | k) + \mathbf{K}(k+1)\mathbf{r}(k+1), \quad (5.15)$$

and, the up-to-date state covariance matrix is given by

$$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{\Lambda}_E]\mathbf{P}(k+1 | k), \quad (5.16)$$

where \mathbf{I} is the identity matrix.

5.5. Extended Kalman Filter

In many robotic applications with sensor data fusing, the system to be modeled fails for having a nonlinear Gaussian noise distribution. While the errors are *approximately* Gaussian, the Kalman filter can be used, even so, probably will not be optimal. For nonlinear systems, is used the *Extended Kalman filter* (EKF). This involves the linearization of the plant, (5.1) and, if necessary, the linearization of the measurement (5.4) cancelling high order terms of the Taylor expansion [15].

The model plant linearization involves the jacobian calculation of the plant model $\nabla\mathbf{\Psi}(\hat{\mathbf{x}}(k), \mathbf{u}(k))$ and to use it as a linear estimate of $\mathbf{\Psi}$ in the Kalman filter. The model measurement linearization involves the jacobian calculation of the measurement model $\nabla\mathbf{h}_{st}(\mathbf{x}(k), \mathbf{p}_t)$ and to use it as linear estimative $\mathbf{\Lambda}_E$.

For our case of EKF use, takes the example of the plant model and nonlinear measurement model presented in Section 5.3. To simplify the exposition, it is assumed that $\mathbf{Q}(k) = \mathbf{Q}$ and $\mathbf{R}_i(k) = \mathbf{R}_i$. For each robot movement, the next stages are follow.

- (1) For represent the robot displacement, it's used (5.1) and (5.7). The known control parameters are utilized for the robot pose estimation at the time $k + 1$, as:

$$\hat{\mathbf{x}}(k + 1 | k) = \begin{bmatrix} x(k) + T(k) \cos[\theta(k)] \\ y(k) + T(k) \sin[\theta(k)] \\ \theta(k) + \Delta\theta(k) \end{bmatrix}. \quad (5.17)$$

- (2) A linearized plant model version is generated in the current estimative of the robot pose $\hat{\mathbf{x}}(k)$ as:

$$\nabla\boldsymbol{\Psi} = \begin{bmatrix} 1 & 0 & -T(k) \sin[\theta(k)] \\ 0 & 1 & T(k) \cos[\theta(k)] \\ 0 & 0 & 1 \end{bmatrix}. \quad (5.18)$$

- (3) The state uncertainty is generated by the state covariance matrix actualization, using measurements obtained until the time k , including itself, through:

$$\mathbf{P}(k + 1 | k) = \nabla\boldsymbol{\Psi}\mathbf{P}(k)\nabla\boldsymbol{\Psi}^T + \mathbf{Q} \quad (5.19)$$

which is the result of the linearization of (5.10).

- (4) The sensor model is linearized around current estimative of robot pose $\hat{\mathbf{x}}(k)$, as:

$$\boldsymbol{\Lambda}_E = \nabla\mathbf{h}_{st}(\mathbf{x}(k), \mathbf{p}_t), \quad (5.20)$$

If the sensor have a reference dot like target in (x_b, y_b) , then

$$\boldsymbol{\Lambda}_E = \begin{bmatrix} 2x(k)[x(k) - x_b] \\ 2y(k)[y(k) - y_b] \\ 0 \end{bmatrix}^T, \quad (5.21)$$

- (5) Using the value $\boldsymbol{\Lambda}_E$, the Kalman gain, that comes for (5.12), is computed like

$$\mathbf{K}(k + 1) = \mathbf{P}(k + 1 | k)\boldsymbol{\Lambda}_E^T \left[\boldsymbol{\Lambda}_E\mathbf{P}(k + 1 | k)\boldsymbol{\Lambda}_E^T + \mathbf{R}_i \right]^{-1}. \quad (5.22)$$

- (6) The innovation, like presented in (5.13), in this way is compute as

$$\mathbf{r}(k + 1) = \mathbf{z}(k + 1) - \mathbf{h}_{st}(\hat{\mathbf{x}}(k + 1), \mathbf{p}_t). \quad (5.23)$$

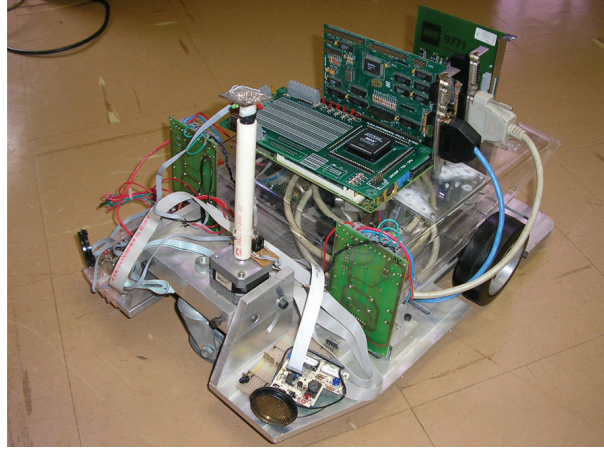


Figure 11: Experimental mobile robot prototype.

(7) Now, is possible to calculate the robot pose estimative, like shown at (5.15), like been

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k+1 | k) + \mathbf{K}(k+1)\mathbf{r}(k+1). \quad (5.24)$$

(8) Finally, the actualized covariance matrix, like illustrated by (5.16), is now calculate as

$$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{A}_E]\mathbf{P}(k). \quad (5.25)$$

After certain time interval, that is shortest possible, the derivatives used in the linearization model must be recalculated through the estimated current state. This backwards a deficiency in the EKF: if the estimated state is very far from the current state, the linear approach of the system behavior will not be enough precise.

6. Mobile Robot Rapid Prototyping

Figure 11 illustrates the mobile robot prototype developed by a research team led by authors and used for experiments validation and as base for the models and simulations.

The use of the rapid prototyping technique in mobile robotic systems differs from the traditional target used in mechanics engineering and enters in new field of research and development for projects of mobile robots mechatronics systems. In this way, the rapid prototyping of these systems is associated not only with the project of the physical system, but mainly with the experimental implementations in the fields of hardware and software of the robotic system. It is fundamental that the architecture of hardware of the considered system be opened and flexible in the way of effecting the necessary modifications for system optimization. A proposal of open architecture system was presented in [16].

The software of the embedded control system of the mobile robot, in the context of the rapid prototyping, can be elaborated in simulators and tested all the parameters for

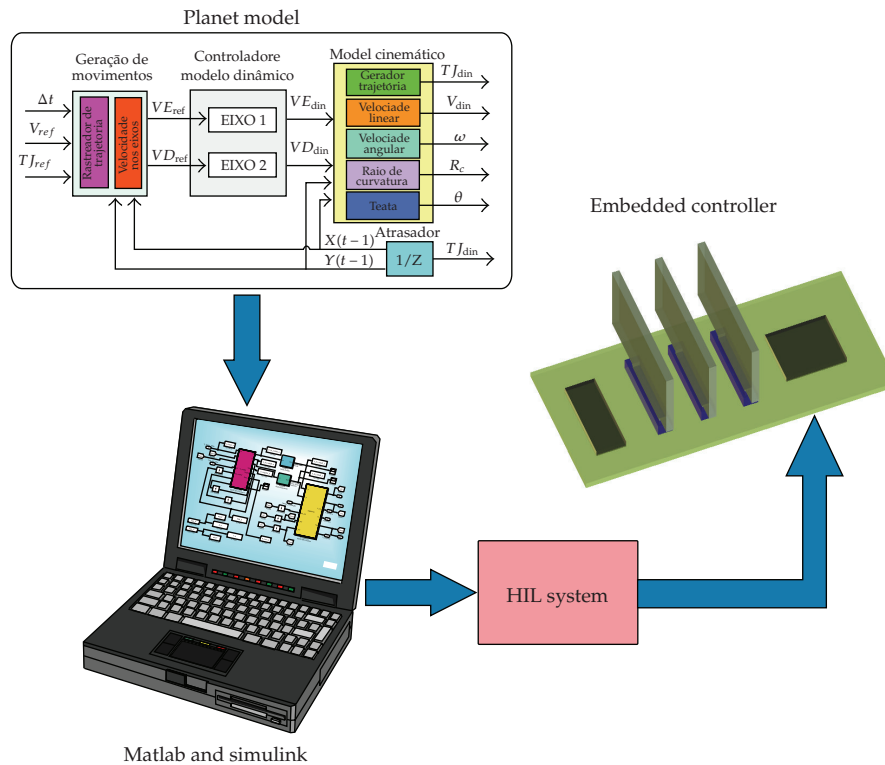


Figure 12: HIL technique for mobile robot system.

adjustments that makes necessary in accordance with the physical system to be implemented, the hardware architecture, the actuators and the sensors. In this way, in the context of this work, the rapid prototyping is then the methodology that allows the creation of a virtual environment of simulation for the project of a controller for mobile robots. After being tested and validated in the simulator, the control system is programmed in the control board memory of the mobile robot. In this way, a economy of time and material are obtained, sooner validating all the model virtually and later operating the physical implementation of the system.

6.1. HIL (Hardware-in-the-Loop)

The HIL technique is used in development and tests for real-time embedded systems. HIL provide a platform accomplish of development for adding the complexity of the plant under control to the tests platform. The control system is enclosed in the tests and developments through its mathematical models representations and all the respective dynamic model [17].

The Figure 12 illustrates the use of the HIL simulation technique for real-time simulation and experimental validation of the considered mobile robotic system. With the utilization of HIL it's possible to implement de Kalman filter methodology with others embedded control techniques for improve the mobile robot localization.

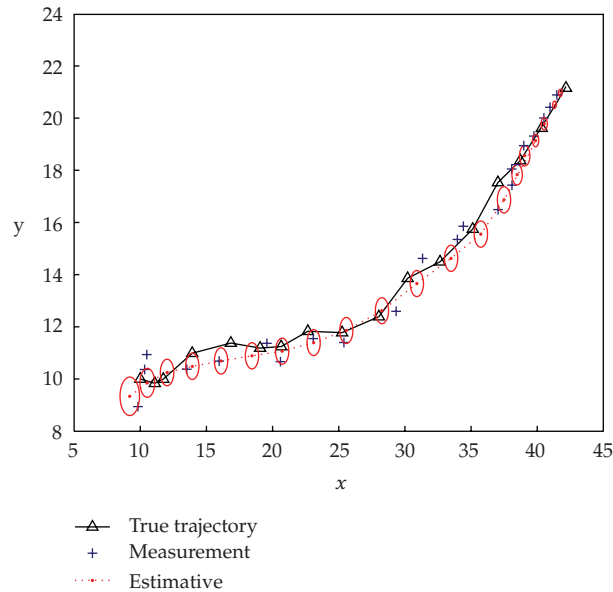


Figure 13: Almost linear trajectory.

7. Experimental Results

Figures 13 and 14 depicts the result of EFK pose estimative applied in the trajectory of the mobile robot prototype (showed in Figure 11) accomplishing different routes.

As illustrated in Figure 13 it can be seen that the mobile robot starts at point (10,10) moving itself with constant linear speed. The ellipses delimit the area of uncertainty in the estimates. It can be observed that these ellipses are bigger in the trajectory extremities, because in these points less measurements data are computed. The average quadratic error varies depending on the chosen trajectory. It can be noticed that the pose estimative improves for more linear trajectories and with high frequency of on-board sensors measurements.

Figure 14 depicts another mobile robot trajectory example. In this case, the route is more irregular and curvilinear. The uncertainty of pose estimative are great than the first example (Figure 13). That's because in this trajectory the on-board sensors measurements becomes more unprecise and with low-frequency samples.

8. Conclusions

Once given the nonlinearity of the system in question, the use of the EKF become necessary. It does not have therefore, at beginning, theoretical guarantees of optimality nor of convergence of this method. Therefore, it was implemented a model of simulation that allows, underneath of next conditions of the reality, to verify the performance of this technique. Among others parameters that were looked to realistic model the increasing error in the measure of the position of a marker (reference object in the environment) to the measure meets that in the distance between the robot and it increases. This effect also was introduced in the estimate of the observation covariance matrix to allow a more coherent performance of the filter. A factor

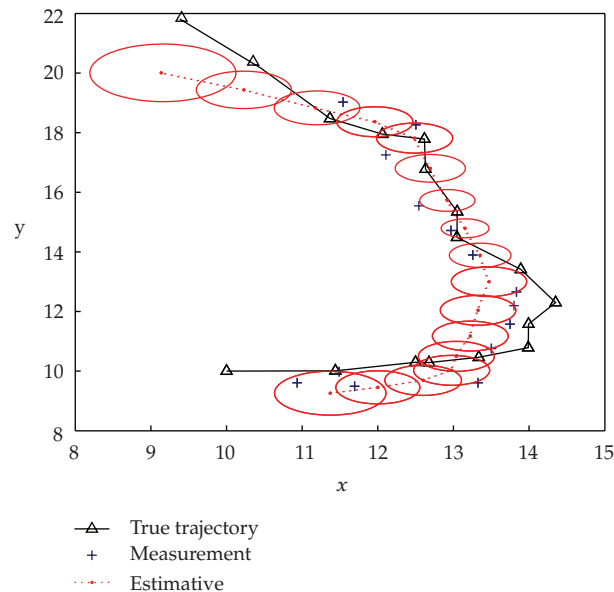


Figure 14: Irregular curvilinear trajectory.

extremely important is the characterization of the covariances matrices of the presents signals in the system.

The results of the simulations associated with experimental validations confirm that this technique is valid and promising so that the mobile robots, in autonomous way, can be able to correct its own trajectory. The consistency of the data fusing relative to the odometry of the mobile robot and the markers it is obtained even after inserted disturbances in the system.

The presented method does not make an instantaneous absolute localization, but successive measurements show that the estimative state converges for the real state of the robot.

References

- [1] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*, vol. 1, Press Syndicate of the University of Cambridge, Cambridge, UK, 2000.
- [2] H. S. Shim, J. H. Kim, and K Koh, "Variable structure control of nonholonomic wheeled mobile robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1694–1699, 1995.
- [3] Y. Zhao and S. L. BeMent, "Kinematics, dynamics and control of wheeled mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 91–96, 1992.
- [4] H. Gonzalez-Banos and J.-C. Latombe, "Robot navigation for automatic model construction using safe regions," in *Proceedings of the 7th International Symposium on Experimental Robotics*, vol. 271 of *Lecture Notes in Control and Information Sciences*, pp. 405–416, 2001.
- [5] M. Kam, X. Zhu, and P. Kalata, "Sensor fusion for mobile robot navigation," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 108–119, 1997.
- [6] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, Wiston, Washington, DC, USA, 1977.
- [7] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 251–263, 1997.
- [8] Y. Bar-Shalom, T. E. Fortmann, and P. G. Cable, "Tracking and data association," *Acoustical Society of America Journal*, vol. 87, pp. 918–919, 1990.

- [9] A. Gelb, *Applied Optimal Estimation*, The MIT Press, Cambridge, Mass, USA, 1974.
- [10] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*, Springer, New York, NY, USA, 1st edition, 1992.
- [11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME, Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [12] H. F. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.
- [13] D. Gregory, M. Jenkin, E. Miliotis, and D. Wilkes, *Reflections on Modelling a Sonar Range Sensor*, World Scientific Press, 1996.
- [14] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents)*, The MIT Press, Cambridge, Mass, USA, 2004.
- [15] P. S. Maybeck, "Stochastic models, estimation, and control," in *Mathematics in Science and Engineering*, vol. 1, Academic Press, New York, NY, USA, 1979.
- [16] L. F. de Melo, J. M. Rosario, and C. R. E. Lima, "A reconfigurable control architecture for mobile robots," in *Proceedings of the 2nd Internacional Symposium on Mutibody and Mechatronics (MuSMe '05)*, pp. 101–108, 2005.
- [17] L. F. De Melo and J. F. Mangili Jr., "Virtual simulator with mobile robot rapid prototyping for navigation systems," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '09)*, pp. 899–904, 2009.