23rd ABCM International Congress of Mechanical Engineering
December 6-11, 2015, Rio de Janeiro, RJ, Brazil

PROBABILISTIC ROBOTICS APPLIED TO SELF-LOCALIZATION INSIDE OIL WELLS OF AN AUTONOMOUS SYSTEM

Hugo Francisco Lisboa Santos
Petrobras/Cenpes - Av. Horácio Macedo, 950, Cidade Universitária - Rio de Janeiro - RJ. CEP: 21.941-915 e
PUC-RIO - Departamento de Engenharia Mecânica - Rua Marquês de São Vicente, 225 - Gávea, Rio de Janeiro - RJ, 22451-900
h.santos@petrobras.com.br

Marco Antonio Meggiolaro
PUC-RIO - Departamento de Engenharia Mecânica - Rua Marquês de São Vicente, 225 - Gávea, Rio de Janeiro - RJ, 22451-900
meggi@puc-rio.br

Abstract. *The use of robots in the petroleum industry is still in the beginning phase. There are several studies for using robots on topside installations, but few for inside wells. This paper describes part of a research project that aims to develop a robot capable of performing maintenance operations at producer and injector wells. The goal is to design an autonomous robot for well intervention. In this development, one of the most challenging problems is to define a reliable self-localization system. Normally, positioning inside wells is performed with cables or with pipe strings. Since the proposed robot is not connected to a cable, an alternative is needed. A possible solution is explored here, based on sensor fusion of a tachometer and a magnetic sensor. For this fusion, several probabilistic and non-probabilistic robotic techniques are considered. Kalman filters are implemented, with adaptations for the current problem. Particle and Histogram Filters are also evaluated, with good results despite their high memory and computational requirements. All techniques are evaluated and compared using field data.*

Keywords: *probabilistic robotics, Kalman, sensor fusion, particle filter, histogram filter*

1. INTRODUCTION

Petroleum and natural gas are normally extracted from reservoir rocks, located from hundreds to thousands of meters below the soil or below the sea level. To allow their production, several wells are constructed. It is performed using drilling rigs, which use drill pipes (DP) and drill collars (DC) to apply weight and to rotate the bit (Thomas, 2004). After drilling, casings are positioned between the formation and the inner part of the well, in order to guarantee the integrity of the system. Finally, completion equipment is positioned inside it, such as safety and control valves and artificial lift systems. That equipment is intended to allow a safe and controlled production (Bellarby, 2012).

During the productive life of oil wells, workovers are usually necessary, with the same type of rigs. It is usually performed in case of an equipment failure, a hole at the string, a production restriction due to scaling, high water cut, high gas production at oil wells, sand production, etc. An important remark is that the daily rates of those rigs range from US$ 260 to US$520 thousand. Besides, from the failure until the repair, the well is kept closed or with reduced production. So, there is a profit loss on wells capable of producing up to 20.000 barrels per day.

For this reason, there are many researches aiming to reduce that intervention cost. The present study is part of one of these researches, which aims to use an autonomous robot for performing some workover operations. The proposed configuration of this robot is briefly presented in Section 2.2.

The development of the self-localization system, presented at this article, is one of the most challenging parts of this project. The robot is supposed to travel a distance of up to 4,000m in pipes with oil and water inside, and it is supposed to reach its destination with at least 20m, or 0.5%, precision. Since there is oil between the wheels and the pipe, odometers may become unreliable, and since the robot is deep inside the soil, wireless or radio transmissions are not applicable. That's why it uses two sensors, an odometer and a magnetic sensor. The positioning sensors proposed in this project are presented in Section 3, and sensor fusion using probabilistic robotics techniques is presented in Section 5. An evaluation of the proposed system is also presented in these sections.

2. IN PIPE ROBOTS

In order to find a good configuration for the proposed robot, many were evaluated. Some of them are presented in Section 2.1. Based on these configurations and on the mission, a helical configuration with wall-pressed wheels was chosen. This configuration is presented in Section 2.2.

2.1 Common in-pipe robot configuration

There are many possible strategies for robot displacement inside pipes, but they can be classified into few groups (Figure 1): (a) PIG, (b) wheels, (c) caterpillar, (d) wall-pressing, (e) walking, (f) inchworm, (g) helical, (h) snake (Roh, 2005 and Kakogawa, 2010).

PIG type robots (Figure 1a) are impelled by the pressure differential on its sides. They are common commercially and don’t use their own energy for their movement. But their moving capacity is considerably limited. Wheeled in-pipe robots (Figure 1b) are one of the most researched and commercially used types, especially because of its capacity of ramification handling. Caterpillar type robots (Figure 1c) have the advantage of moving on non-uniform surfaces and of transposing obstacles inside the pipe. Wall-pressing robots (Figure 1d) are particularly interesting on vertical or near vertical pipes, since they allow the development of high axial forces. Walking robots (Figure 1e) are versatile, but their control is difficult, mainly because of the large number of actuators. Worm robots (Figure 1f) have advantage on vertical and curved pipes. Helical robots (Figure 1g) normally have simple easily controllable structures. Finally, snake robots (Figure 1h) are highly versatile and capable of wall-pressed movement as well as swimming; but their control is also very difficult.

These configurations may be merged as hybrid robots (Roslin, 2012). An example of this procedure is the caterpillar-pressed type. It allows good forward and backward traction, capacity of handling different diameters, and the possibility of changing the caterpillar angle. Other example is the wheel-pressed type. It has good capacity of performing curves, especially at pipe branches. They have good traction capacity with a simple structure and good mobility at high speeds. A third hybrid type is the helical one with wall-pressed wheels (Figure 1i). It has low drag compared with traditional screw-type ones, good traction capacity, and the possibility of using just one motor.

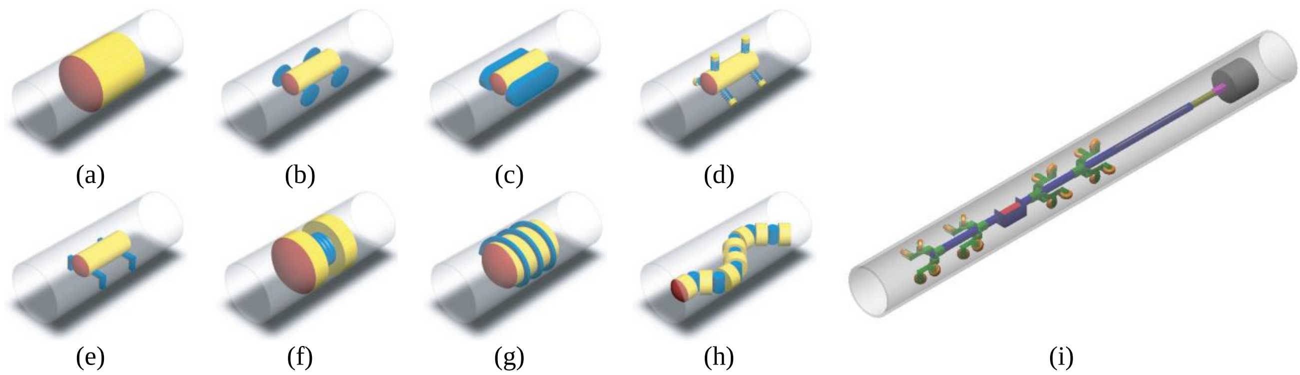


Figure 1. (a)-(h): Different types of robot for in-pipe moving (Kakogawa, 2010);
(i): Proposed configuration for the robot prototype.

2.2 The proposed robot configuration

The proposed robot configuration is based on the third hybrid type: the helical with wall-pressed wheels, see Figure 1i. It was chosen for several reasons. The first is the high traction need, as the robot is supposed to travel up to 4,000 m downwards and upwards and to perform operations that demand high axial force. The second is the low drag need, since it is supposed to be an autonomous robot that will travel a relatively long distance. The third and most important one is the possibility of using only one actuator, which increases reliability and allows the use of the robot in small diameters.

The general configuration is presented in Figure 1i. The robot uses magnetic wheels to reduce the oil film thickness and to allow a metal-to-metal contact. It also uses a system of leaf springs to press the wheels against the pipe. This system also allows using the robot at different pipe diameters.

3. IN PIPE SELF-LOCALIZATION

There are several ways for in-pipe localization. The simplest is odometry. It consists on measuring the speed or the number of turns of the wheel. Despite its simplicity and wide use, it may present slippage problems.

Other simple and common one is cable localization. In this type, the robot is linked to a cable and position is calculated by the number of rotations of the winch and the rope tension. Petrobras Girino’s localization system, for example, is performed in this way (Reis, 2000), as many other robots (Qi, 2010).

At small depth, it may be possible to use telemetry in the robot. One example is a wireless tracking and positioning system, which uses extremely low frequencies (Qi, 2009). Other systems using X-Ray or Co-60 have also been tried, but are not currently used because of its inherent risks.

In well engineering, there are some ways of in-pipe positioning (Thomas, 2004). The first one is using a steel pipe column, a common method during well drilling. It consists on adding the length of the drill pipes used at that activity. The second is cable positioning. It is commonly used on well logging and workover operations. A third is the Gamma

Ray profile. It uses the natural radiation of the formations to perform depth correlation. The last one is Casing Collar Locator: it is a magnetic sensor designed to allow the detection of Casing Collars.

In the present project, a fusion of two techniques will be used: odometry and magnetic sensing. The last one is presented in further detail in Section 3.1, with a treatment algorithm. Furthermore, when a single sensor doesn't show enough precision for a certain task, it is common to use sensor fusion, which is the fusion of different sensors in order to improve the estimate. That approach is presented in Section 3.2, and it will be used in this work.

3.1 Magnetic Sensing

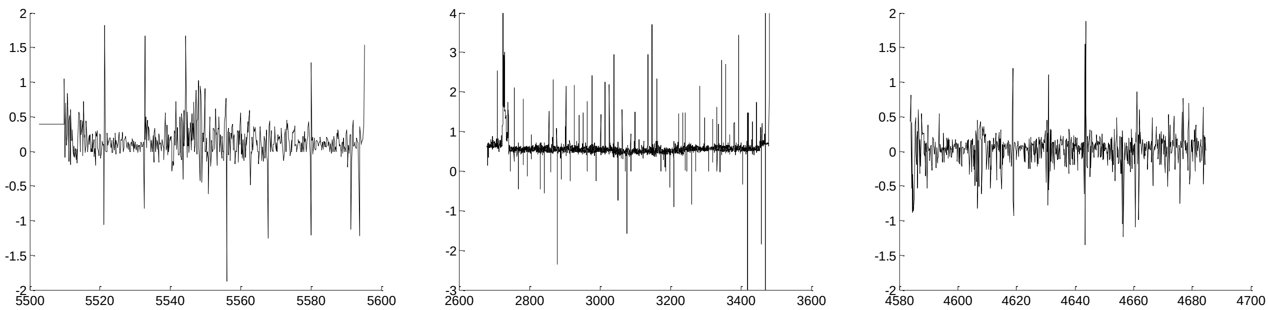
In well engineering, it is common to use magnetic sensors for depth correlation, among different sensors. Those magnetic sensors are called Casing Collar Locators, or CCL. Normally, it is combined with other one, the Gamma Ray Profile, to allow correlation between an open hole reading and a cased hole one. This is essential for subsequent operations such as perforation. As it is used at the primary depth control, CCL is run on almost every cased hole well.

Pipes inside wells are united by threaded collars, as presented in Figure 2a. As collars are thicker than the other pipes, there is a change in magnetic permeability. The CCL takes advantage of this change and uses it to identify a pipe joint. This tool is shown in Figure 2b and presented schematically in Figure 2c. It is normally composed of coils and magnets. The most sensitive of these arrangements is two magnetic poles positioned on either side of a central coil, see Figure 2c. The magnetic flux lines are distorted when the tool passes a location at which the metallic casing is enlarged by a collar. This distortion gives rise to a change in the magnetic field around the conducting coil, within which current is induced. The signal is amplified and recorded in the form of a voltage spike.

(a) (b) (c)

Figure 2.(a) Illustration of a casing collar; (b) Photo of a casing collar locator; (c) Schematic of the casing collar locator.

Typical CCL readings are presented in Figure 3. The peaks represent the Casing Collars. In Figure 3a there are 8 collars identified by the peaks and by the almost uniform peak spacing. In Figure 3b, a long track, there are more than 50 collars, some of them easily identifiable, some of them eventually interpreted as noise. In Figure 3c, there are again 8 not easily identifiable collars.



(a) (b) (c)

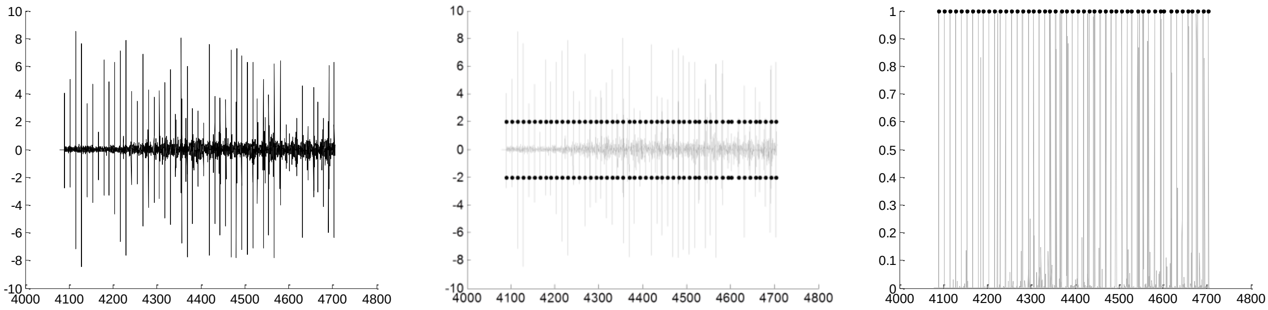
Figure 3. Example CCL readings used to test the system (abscissa: depth; ordinate: CCL reading).

To allow a probabilistic identification of the collars, a procedure based on Bayes Rule was used (see Section 4.1). It consists on calculating the probability $p(c|V_{ccl})$ of the captured value V_{ccl} being a collar c or a noise $\bar{c}$:

$$p(c|V_{ccl}) = \frac{p(V_{ccl}|c)\,p(c)}{p(V_{ccl}|c)\,p(c) + p(V_{ccl}|\bar{c})\,p(\bar{c})} = \frac{f(V_{ccl},\mu_c,\sigma_c)\,p(c)}{f(V_{ccl},\mu_c,\sigma_c)\,p(c) + f(V_{ccl},\mu_{\bar{c}},\sigma_{\bar{c}})\,p(\bar{c})} \tag{1}$$

$$p(\bar{c}|V_{ccl}) = \frac{p(V_{ccl}|\bar{c})\,p(\bar{c})}{p(V_{ccl}|c)\,p(c) + p(V_{ccl}|\bar{c})\,p(\bar{c})} = \frac{f(V_{ccl},\mu_{\bar{c}},\sigma_{\bar{c}})\,p(\bar{c})}{f(V_{ccl},\mu_c,\sigma_c)\,p(c) + f(V_{ccl},\mu_{\bar{c}},\sigma_{\bar{c}})\,p(\bar{c})} \tag{2}$$

In these equations, $p(c)$ is the probability that a reading is a collar, $p(\bar{c})$ is the probability that the reading is not a collar, $f(V_{ccl}, \mu_c, \sigma_c)$ is the probability distribution function (p.d.f.) of the collar reading, and $f(V_{ccl}, \mu_{\bar{c}}, \sigma_{\bar{c}})$ is the p.d.f. of the noise reading. Both p.d.f. are assumed as Gaussian.



(a) (b) (c)

Figure 4. Evaluation of the identification system (abscissas: depth): (a) Original CCL reading (ordinate: CCL reading) (b) Real position of the collars (ordinate: CCL reading in V) (c) Output probability (ordinate: collar probability).

To illustrate this, the CCL reading presented in Figure 4a will be used. The real position of the collars is presented in Figure 4b, as points. They are shown at $\pm 2.0V$ and overlapped to the CCL graph just for convenience. At this CCL reading, collar readings presented a mean value of $\mu_c = 3.7V$ (absolute value) and a standard deviation of $\sigma_c = 0.90$V. Noise readings presented a mean value of $\mu_{\bar{c}} = 0.004V$ and a standard deviation of $\sigma_{\bar{c}} = 0.31V$.

Figure 4c presents the probability that there is a collar at each position, ranging from 0 to 1, given the readings presented in Figure 4b. It presented 100% probability of correctly identifying a collar and 2.4% probability of indicating a false positive.

In this first evaluation, the real means and standard deviations were used. But these values are usually not available, since data is collected during the operation. For this reason, instead of using the real means and standard deviations, at the real algorithm, a first guess of those values was used initially. After that, results were calculated again using exponentially-weighted moving average and exponentially-weighted moving variance, with (Finch, 2009):

$$\mu_n = \alpha\, \mu_{n-1} + (1-\alpha)\, x_n \tag{3}$$

$$\sigma_n^2 = (1-\alpha)\, (\sigma_{n-1}^2 + \alpha\, (x_n - \mu_{n-1})^2) \tag{4}$$

In these equations, μ_n and σ_n^2 represent the mean and the variance at iteration n, and α represents the weighting factor.

Using $\mu_c = 1.3$, $\sigma_c = 0.20$, $\mu_{\bar{c}} = 0.$ and $\sigma_{\bar{c}} = 0.20$ as first guess and $\alpha = 0.05$, it was obtained again 100% probability of correctly identifying a collar, and 2.8% probability of indicating a false positive. Results were similar with other values as first guess, being able to correctly identify the collars with good precision. CCL readings of Figure 3 were also tried with the same initial guess. The system presented 89% / 0.83% probability for graph (a), 99% / 3.6% for graph (b), and 90% / 1.5% for graph (c).

An important remark is that in a first approach, Bayes' Rule wasn't used. Instead, the probability $p(c|V_{ccl})$ was calculated using only probability theory, as $\frac{f(V_{ccl},\mu_{\bar{c}},\sigma_{\bar{c}})}{f(V_{ccl},\mu_c,\sigma_c)+f(V_{ccl},\mu_{\bar{c}},\sigma_{\bar{c}})}$. As expected, probability of false positive decreased to about one tenth as Bayes was used. In the graph from Figure 4a, for example, it decreased from 32% to 2.8%. In the graphs from Figure 3, reductions went from 26% to 0.83% for graph (a), from 17% to 3.6% for graph (b), and from 39% to 1.5% for graph (c).

3.2 Sensor Fusion

When a single sensor presents significant reading errors, it may be convenient to use sensor fusion, which is the use of the signal of a sensor to correct the other. In the present project, odometers may slip due to oil films between the wheels and the pipe. On the other hand, casing collars are not always easily identifiable. The filtered CCL, proposed in Section 3.1, presents good results in most situations, but may present problems in some of them. Furthermore, collars have typical 10m spacing, compromising the positioning precision. Based on this, a sensor fusion between an odometer and a CCL was chosen for the magnetic reading to correct slippage and the odometer reading to help avoid wrong collar identification and allow position estimation between collars. To do so, some methods of probabilistic robotics are evaluated, presented in Section 4.

4. PROBABILISTIC ROBOTICS

Probabilistic Robotics is the approach that deals with uncertainty in robot perception and action. The main idea is to represent uncertainty explicitly, using probability theory. Instead of using a simple "best guess," probabilistic algorithms present the probability that the robot is at a certain place.

In this section, a short introduction to some filtering techniques is presented, starting with a general Bayes Filter, Gaussian Filters, and some Nonparametric Filters. To illustrate them, the case of a robot moving upwards in a production string is used. Furthermore, for each described technique, there are some trials in order to evaluate them as possible filtering techniques for the proposed robot.

The robot is simulated for its motion through four well sections, presented in Figure 3 and Figure 4a. For each motion, two performance indexes are used. The first is the greatest distance of the most likely point, i.e. the distance of the most probable position according to the filter and the real position. The second was the uncertainty, measured by the distance from the 5^{th} percentile to the 95^{th} percentile. The results presented in Figure 7, Figure 9 and Figure 11 illustrate the results during the first 100 seconds. In these simulations, a variable slippage with mean 0.10 and standard deviation 0.05 was modeled.

4.1 Bayes Filters

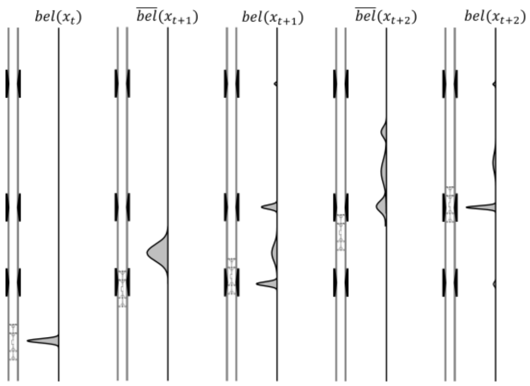
Bayes Filters are based on *Bayes' Rule*, which relates conditionals of the type $p(x|y)$ to $p(y|x)$, when $p(x) \neq 0$. For the discrete case:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{x'} p(y|x')p(x')} \tag{5}$$

Bayes filters follow the algorithm presented in Figure 5a and illustrated in Figure 5b, where $bel(x_t)$ is the belief that the robot is at a certain set of possible positions x_t at a moment t. These belief distributions represent the probability of the robot being at some place, given its internal knowledge about the environment. Furthermore, u_t represents the control system actuation; z_t are the sensor readings, and η is a normalization factor. For each point, based on the previous belief $bel(x_{t-1})$ and on the control commands u_t, a first guess of the new beliefs $\overline{bel}(x_t)$ is performed. After that, based on this new guess $\overline{bel}(x_t)$ and on the measurements, the belief is updated, leading to a new corrected guess $bel(x_t)$.

Algorithm BayesFilter$(bel(x_{t-1}), u_t, z_t)$
for all x_t do
$\overline{bel}(x_t) =$
$\int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx$
$bel(x_t) = \eta p(z_t|x_t)\overline{bel}(x_t)$
endfor
return $bel(x_t)$

(a)



(b)

Figure 5. (a) General algorithm for Bayes filtering (b) Illustrative application of Bayes filtering.

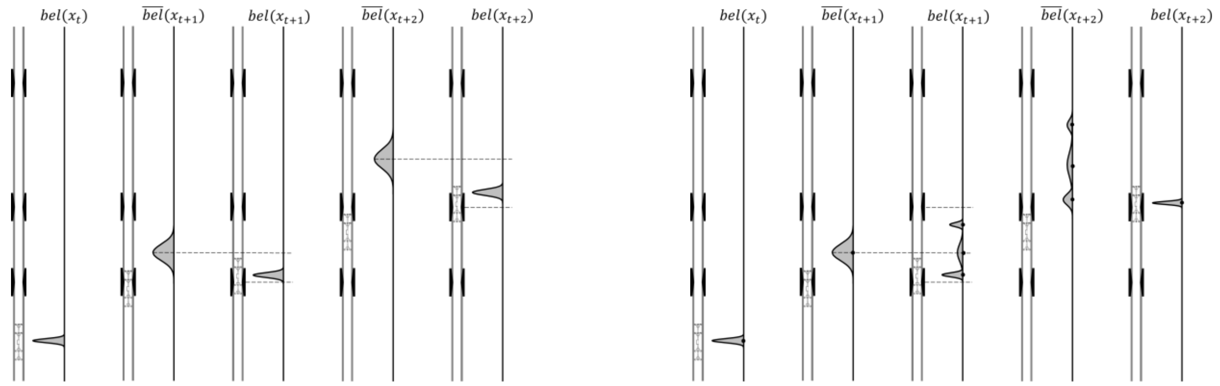
The use of a Bayes Filter without any additional assumption is impracticable, as it would require large computational memory and processing capacity. So, assumptions and simplifications are usually made to keep the problem tractable, as discussed in Sections 4.2 and 4.3.

4.2 Gaussian Filters

Gaussian Filters share the basic idea that beliefs are represented by multivariate normal distributions. Their probability distribution functions have the following form, where μ is the mean vector and Σ is the covariance matrix (Thrun, 2006):

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}}.\exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\} \tag{6}$$

A common implementation of Gaussian Filters is the Kalman Filter. To illustrate it, Figure 6a will be used. In this figure, the robot has an initial belief of its position. As it moves upward, the mean and the covariance matrix are updated to generate the estimated belief $\overline{bel}(x_{t+1})$. After that, the sensor readings are used to update the belief distribution $bel(x_{t+1})$. In all cases, the belief distribution is represented by a Gaussian.



(a) (b)

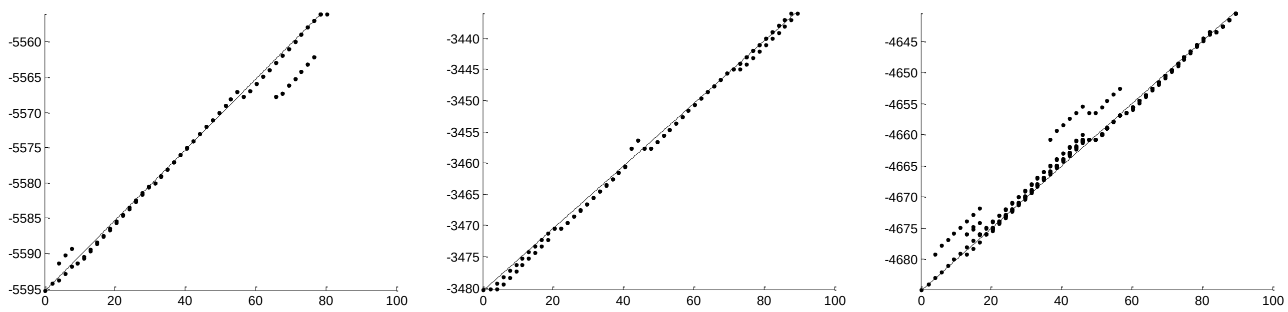
Figure 6. Illustrative application of (a) Kalman Filter Algorithm and (b) MHT Algorithm.

Kalman Filters are very used for tracking, i.e. to follow a vehicle when the initial position is known. It is used for aircraft and spacecraft control, as well as for vessels' dynamic positioning. Despite this widespread use, they present some limitations, particularly when dealing with non-unique hypotheses. This is a problem for the proposed robot self-localization system, since collars are not unique elements. In other words, when the robot detects a collar, it doesn't know *a priori* which collar this is. There are other similar implementations of pure Gaussian filters, such as Extended Kalman Filters (EKF) and Information Filters (IF), which don't allow following more than one hypothesis.

In order to develop a Gaussian-derived filter that could allow more than one hypothesis, it was used the Multi-Hypothesis Tracking (MHT) algorithm. The MHT represents the posterior by a weighted sum of the Gaussians.

$$p(x) = \frac{1}{\sum_l \psi_{t,l}} \sum_l \psi_{t,l} \det\left(2\pi\Sigma_{t,l}\right)^{-\frac{1}{2}} . \exp\left\{-\frac{1}{2}\left(x_t - \mu_{t,l}\right)^T \Sigma_{t,l}^{-1}\left(x_t - \mu_{t,l}\right)\right\} \tag{7}$$

The procedure is similar to Kalman filter and is illustrated in Figure 7, where the hypotheses are marked as points. The difference is that, instead of following a single point, MHT may follow more points, each of them with an associated probability. A limit can be imposed in order to avoid the need to follow a large number of points. For this purpose, a common application is following just points with probability greater than a certain value.



(a) (b) (c)

Figure 7. Results for Multi-Hypothesis tracking algorithm (abscissa: time; ordinate: depth).

The MHT filter is applied to the wells with the CCL readings presented in Figure 3. Results are presented in Figure 7, where the continuous line presents the real position of the robot while the points present the MHT estimate. The dispersion is found to be 25, 16, 22 and 16 respectively for the readings of Figure 3 a, b, c and Figure 4a. The greatest error was 5.5, 18, 7.4 and 8.2, respectively.

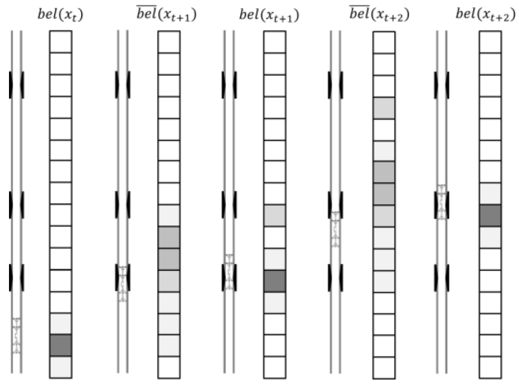
4.3 Non Parametric Filters

An alternative to Gaussian techniques is the use of nonparametric filters. There are several implementations of this type of filters. Some of them use a decomposition of the state space. Others approximate the state space by random samples drawn from the posterior distribution. Two techniques are implemented: Histogram Filter, a method of the first type, and Particle Filter, of the second one.

Histogram filters are the application of Discrete Bayes Filter to continuous state space, as an approximation. The general algorithm of discrete Bayes filter is presented in Figure 8a and illustrated in Figure 8b, where the domain is divided into parts, each one with a certain probability that the robot is there. In this figure, a darker color means that there is a higher probability that the robot is there. As the robot moves, there is a change in the belief distribution to $\overline{bel}(x_{t+1})$. As it obtains new sensor readings, the distribution $bel(x_t)$ is achieved.

Algorithm DiscreteBayesFilter $(\{p_{k,t-1}\}, u_t, z_t)$
for all k do
$\bar{p}_{k,t} = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i) p_{i,t-1}$
$p_{k,t} = \eta . p(z_t | X_t = x_t) . \bar{p}_{k,t}$
endfor
return $\{p_{k,t}\}$

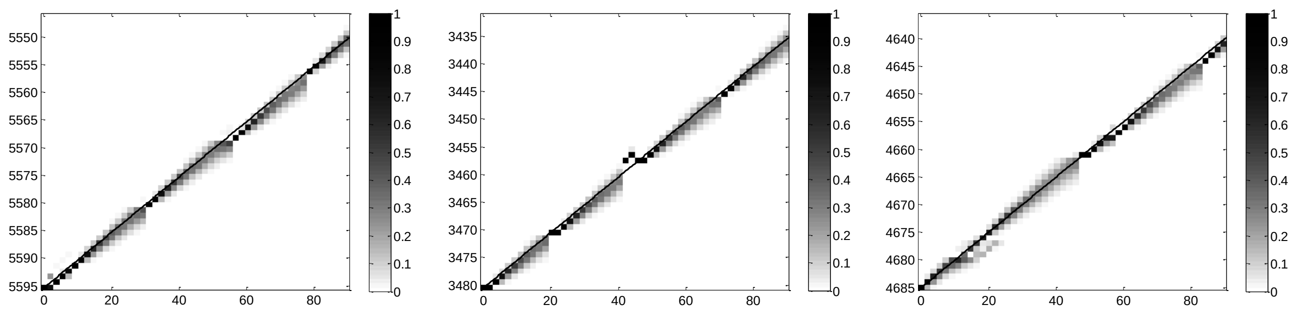
(a)



(b)

Figure 8. (a) Discrete Bayes Filter Algorithm (b) Illustrative application of Discrete Bayes Filter Algorithm.

The Histogram Filter is applied to the wells with the CCL readings presented in Figure 3. Results are presented in Figure 9, where the continuous line presents the real position of the robot and the points present the probability that the robot is at a certain position. The dispersion is 4, 9, 6 and 5 respectively for the readings of Figure 3 a, b, c and Figure 4a. The greatest error is 0.9, 13, 1.4 and 4.6, respectively.



(a) (b) (c)

Figure 9. Results using Histogram Filter for each CCL reading (abscissa: time; ordinate: depth).

The Particle Filter algorithm is presented in Figure 10a and illustrated in Figure 10b. It consists on generating a large number of particles. At the estimation step, a set of particle positions $\bar{\chi}_t$ is randomly generated based on the position of the previous set χ_{t-1} and on the control commands u_t. For each particle, a weight $w_t^{[m]}$ is calculated, based on the sensors reading z_t. This weight represents the probability that the robot is at that position, given the reading z_t.

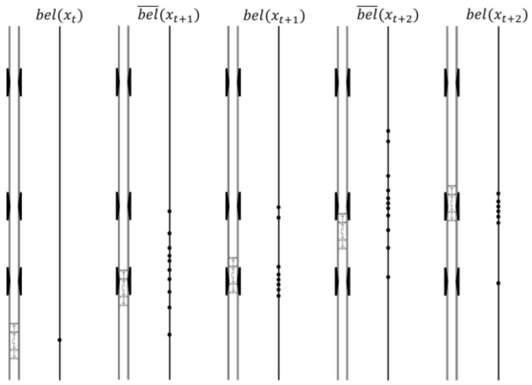
This filter is also applied to the wells with the CCL readings presented in Figure 3. Results are presented in Figure 11, where each point represents a particle. The dispersion is 22, 20, 23 and 22 respectively for the readings of Figure 3 a, b, c and Figure 4a. The greatest error is 9.1, 37, 3.5 and 6.6, respectively.

5. CONCLUSION

In this work, several filters were evaluated for a self-localization application inside oil wells. Based on the simulation results using field data, the Histogram Filter presented lower dispersion and better precision than Kalman and Particle Filters. Furthermore, the Histogram Filter algorithm is simpler and has lower computational cost. It is not normally used for 2D and 3D problems since it consumes too much computer memory, but for a 1D problem with about 4000 points it doesn't represent a significant restriction. For these reasons, the Histogram Filter has been chosen for this application.

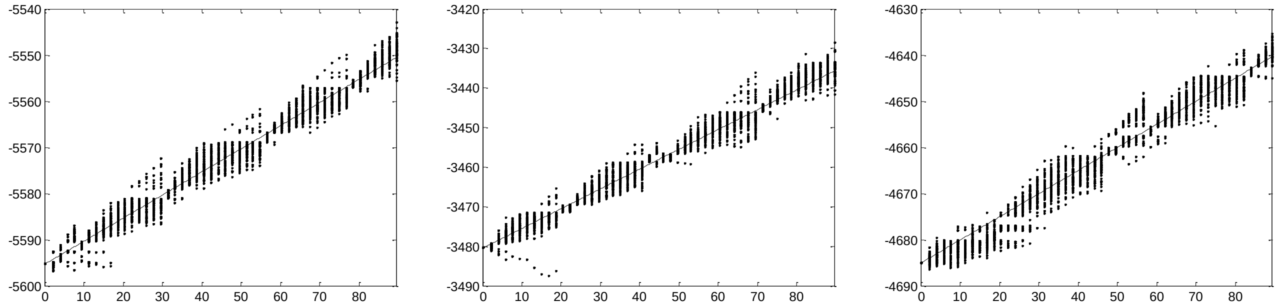
Algorithm ParticleFilter (χ_{t-1}, u_t, z_t)
  $\bar{\chi}_t = \chi_t = \emptyset$
  for $m = 1$ to M do
    sample $x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})$
    $w_t^{[m]} = p(z_t | x_t^{[m]})$
    $\bar{\chi}_t = \bar{\chi}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
  endfor
  for $m = 1$ to M do
    draw i with probability $\alpha\ w_t^{[i]}$
    add $x_t^{[i]}$ to χ_t
  endfor
  return χ_t

(a)



(b)

Figure 10. (a) Particle Filter Algorithm (b) Illustrative application of Particle Filter.



(a) (b) (c)

Figure 11. Results using Particle Filter for each CCL reading (abscissa: time; ordinate: depth).

6. REFERENCES

Bellarby, J. (2012). *Well Completion Design.* Amsterdam.

Finch, T. (2009). *Incremental calculation of weighted mean and variance.* Cambridge: University of Cambridge Computing Service.

Kakogawa, A. M. (2010). Mobility of an In-pipe Robot with Screw Drive Mechanism inside Curved Pipes. *Proceedings of the 2010 IEEE International Conference on Robotics and Biomimetics* (pp. 1530-1535). Tianjin, China: IEEE.

Qi, H. Y. (2010). Wireless tracking and locating system for in-pipe robot. *Sensors and Actuators*.

Qi, H. Z. (2009). Tracing and localization system for pipeline robot. *Mechatronics*.

Reis, N. R. (2000). Gabarito Interno Robótico com Incidência Normal ao Oleoduto. Rio de Janeiro.

Roh, S. C. (Fevereiro de 2005). Differential-Drive In-Pipe Robot for Moving Inside Urban Gas Pipelines. *IEEE TRANSACTIONS ON ROBOTICS.* Seoul, Korea: IEEE.

Roslin, N. S. (2012). A Review: Hybrid Locomotion of In-pipe Inspection Robot. *2012 International Symposium on Robotics and Intelligent Sensors* (pp. 1456 – 1462). Elsevier.

Thomas, J. E. (2004). *Fundamentos de Engenharia de Petróleo.* Rio de Janeiro: Interciência.

Thrun, S. B. (2006). *Probabilistic Robotics.* Massachusettsk: MIT Press.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.