An Anticipatory Control for a Flexible Manufacturing System Based on the Perception of Mobile Units Using WSNs

J.A. Silva-Faundez, C. Duran-Faundez, P. Melin, C. Aguilera

Joaquin Alder Silva-Faundez, Cristian Duran-Faundez*,
Pedro Melin, Cristhian Aguilera
Departamento de Ingeniería Eléctrica y Electrónica,
Universidad del Bío-Bío
Avda. Collao 1202, Casilla 5-C, Concepción, Chile
E-mail: joaquin.silvaf@ingenieros.com, {crduran, pemelin, cristhia}@ubiobio.cl
*Corresponding author: cduran@ubiobio.cl

Abstract: In this paper, we design and evaluate a control system which, by using as input RSSI measures, allows anticipatory movements of robotic arms decreasing idle times at the CIMUBB Laboratory. Classical Log-Normal model, which relates the strength of a signal received by a node with the distance at which the sender of the signal is, was adopted. The hidden state of the system is determined by the Extended Kalman filter which allows us to estimate the distance and the speed of pallets moving over a closed-loop conveyor belt. From these estimates, remaining time in which the pallet will get to a stopping point near the robot is determined. This information is finally processed by a controller to determine the instant at which the robot must operate and handle the pallet. Both, a Proportional-Integral and a Fuzzy controller, were implemented and evaluated. Results show the feasibility of using wireless signals to accomplish the described goal, with some practical restrictions.

Keywords: Anticipatory control, intelligent manufacturing, wireless sensor networks, computer applications, localization.

1 Introduction

In the last few years, Wireless Sensor Networks (WSNs) [1] have attracted the industrial sector because of their many advantages such as low power consumption, high flexibility and scalability. Indeed, many application cases of WSNs, including factory and building automation, inventory management, efficient irrigation, cattle tracking and many others, nowadays exist [2–6]. Moreover, latest works on new approaches, such as the "active products"and the "ambient intelligence", have considered WSNs as a supporting technology because they represent an efficient way to embed autonomous intelligence, sensing and communication capabilities on multiple objects (see, for example, works in [7–9]). On the other hand, localization and positioning of people, machines, vehicles, and other entities moving in automated environments are highly desired capabilities in modern industries and services. Indeed, these capabilities may allow significant improvements in process control, monitoring, tasks distribution, resources optimization, safety, and others. In certain application cases, these can be achieved thanks to very accurate location systems, such as the Global Positioning System (GPS) [10] and Ubisense [11]. In the matter of WSNs in localization applications, these are interesting for many cases where, given the number of units to identify, limitations of the technologies, and other factors, accurate solutions such as GPS or UWB-based indoor location technologies represent too important energy, economic or structural costs, or when we want to take advantage of the already installed wireless sensing equipment. Here, in order to provide distance estimations, many works have applied the Received Signal Strength Indicator (RSSI), because it is inherently provided by most of the currently applied transceivers in wireless sensor nodes.

In [12] we reported our first experiments on the 1-Dimensional (1-D) positioning of pallets moving through a closed loop conveyor belt at the CIMUBB Laboratory [13]. We aimed to minimize the idle time of a robot arm by integrating information about the position of the pallets approaching its Flexible Manufacturing Cell (FMC). To do so, pallets carried WSNs devices periodically sending messages to a receiver node near the robotic arm. RSSI traces were captured and analyzed concluding that, even if this indicator presents high instabilities in closed environments, this metric could be useful in our case study.

In this paper, we design and evaluate a control system which, by using RSSI measures as input, allows anticipated movements of robotic arms decreasing idle times at the CIMUBB Laboratory, continuing works in [12]. In the system, we adopted the classical Log-Normal model which relates the strength of a signal received by a node with the distance at which the sender of the signal is. The hidden state of the system is determined by the Extended Kalman filter which allows us to estimate the distance and the speed of the objects of interest (the pallets). From these estimates, remaining time in which the pallet will get to the stopping position near the robot is determined. This information is finally processed by a controller to determine the instant at which the robot must operate and manipulate the pallet. Both, a Proportional-Integral (PI) controller and a fuzzy controller, were implemented and evaluated. Results show the feasibility of using wireless signals to estimate distances in an indoor real-world manufacturing environment, such as the CIMUBB Laboratory, considering some restrictions.

The remainder of this article is organized as follows: In Section 2 we present the CIMUBB Laboratory, describing its physical configuration. In Section 3 we design the anticipatory control system, describing the developed models, filtering, and controllers to be evaluated. In Section 4 we describe our test implementation, exposing performed tests and results in Section 5. Finally, in Section 6 we conclude and give some future directions.

2   Case study: CIMUBB Laboratory

CIMUBB laboratory is a small scale fully automated Flexible Manufacturing System (FMS) consisting of three flexible manufacturing cells. All of the flexible cells are integrated by means of a closed loop conveyor belt through which several parts travel on pallets. The transportation between the various cells and the conveyor belt is made by preprogrammed robotic arms. Due to the way this system is implemented, most of the time, the robots do not have any information about the position of the pallets to be handled. Indeed, only when a pallet passes over a stop station (where magnetic sensors are located), the system knows the position of a particular pallet. This produces a long-term significant idle time between the time the pallet is identified and stopped until the moment the robot handles it (this is where this work acts). The implementation adopted in this work is similar to the one in [12]. Pallets carrying WSNs devices periodically send messages to a receiver node near the target FMC stop station. RSSI traces are collected and processed by a control system. The system sends orders to the robotic arm, trying to match the time of the pallet's arrival with the end of a robot movement reaching an approaching position that allows it to grasp the pallet with minimal waiting time.

Obviously, this is not the industrial solution. The simplicity of this system should enable us to meet the same goal by simply placing sensors before the stop station. Our current research subjects aim to provide positioning systems and anticipatory control (among others) in more complex production environments. This work is a study that will allow us to determine (as has been done in other work environment) how usable the RSSI is as a metric to estimate distance and speed of moving objects in real-world indoor environments, initially, in a simple cyclic system such as the CIMUBB laboratory.

Regarding the problem of using an RSSI-based location, our previous studies have resulted

in the characterization of electromagnetic (EM) waves allowing to observe the manner in which signals behave in this system. Now, it is necessary to characterize the behavior of the object of interest (i.e., the pallets) at which we want to estimate distances. The behavior of a pallet is given by the mathematical model of its position regarding time. Therefore, it is necessary to know the technical details of the conveyor system. The conveyor is a closed loop conveyor system made of metal. It is supported by pillars that rise the system 1 meter above the building's floor. Circulating pallets moving on the conveyor are dragged by a belt consisting of two plastic chains running in parallel and driven by an AC motor. This rotates at a constant speed giving the pallets a forwarding speed of 0.19[m/s] over straight lines, which decreases to 0.18[m/s] at the corners. activated via a switch which makes it to operate independently from the FMS. The location of the flexible cells, together with the dimensions of the conveyor belt, are shown in Figure 1(a). In this work, we focused on the Machining FMC which is shown in Figure 1(b).
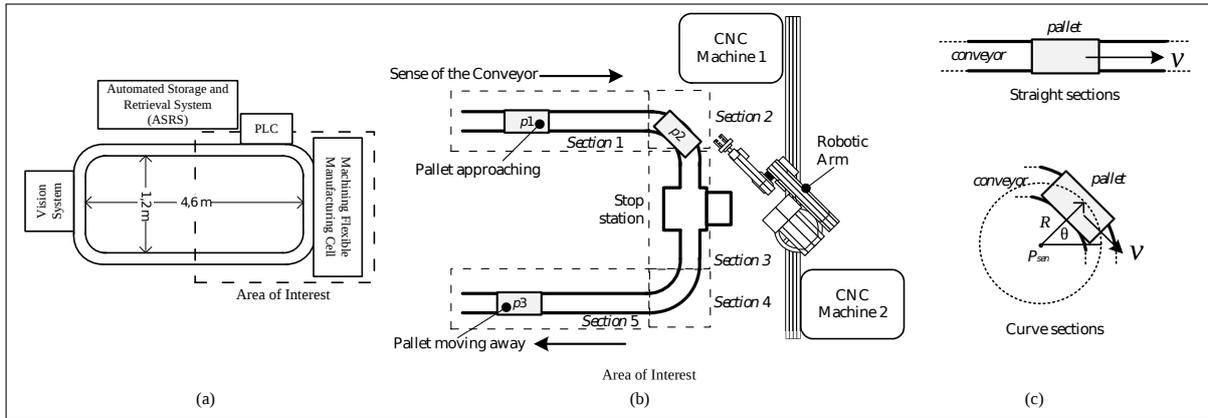

Figure 1: Case study.

Now, knowing the conveyor's geometric structure it is possible to model the path of the pallets within the area of interest. The pallets path, over the area of interest, can be divided into five sections, as shown in Figure 1(b): three of them, in which pallets perform straight movements (sections 1, 3 and 5), and two for curved movements (sections 2 and 4). Thus, the position of a pallet over time can be defined as:

$$
\overrightarrow{r}_k =
\begin{cases}
\overrightarrow{r}_{k-1} + \overrightarrow{V}.\Delta T & \text{during straight} \\
& \text{movements} \\
R.\left[\left(\cos\left(-\omega.\Delta T + \theta\right) + P_{cenx}\right).\hat{i} + \left(\sin\left(-\omega.\Delta T + \theta\right) + P_{cenx}\right).\hat{j}\right] & \text{during curved} \\
& \text{movements}
\end{cases}
$$

(1)

were $\theta = \arctan\left(\frac{P_{k-1_y} - P_{ceny}}{P_{k-1_x} - P_{cenx}}\right)$ and $\omega = \frac{|\overrightarrow{V}|}{R}$. For both of cases, $\overrightarrow{r}$ and $\overrightarrow{V}$ represent the position and velocity vectors of the pallet, respectively. The unit vectors i and j, take the direction of the Cartesian reference system xy. ΔT is an increment in time. Thus, the equations give the new position of the pallet at a determined time increment from its current position (r_{k-1}). Particularly, during curved movements, ω is the angular velocity of the pallet calculated from its rapidity and the radius of the curve (R). The pallet's position is reflected in the angle that has about the center of rotation P_{cen} (see Figure 1(c)).

We established a reference system in the plane xy, with its origin at the bottom left of the conveyor (end of Section 5). From this reference coordinate, various reference points are fixed in order to mark the beginning and the end of each section of the path. Along with these references,

the location of the data collector node is set. This point will serve as a reference point for distance estimation. This node is located 24 centimeters from the stop station.

3   System design

The proposed "*Anticipatory Control System*"(ACS) will be in charge of detecting the approaching of a pallet to the manufacturing cell and estimate the time at which the robot must act in anticipation of its arrival. The system estimates the proximity from available RSSI information supplied by the collector node (a node located near the controlled robotic arm). These values are calculated by the node upon receipt of datagrams sent by the wireless nodes located on the pallets. These act as beacons regularly indicating their presence. The intensity with which the collector gets beacons' signals can be related to the distance at which pallets are through a mathematical model describing their behavior. However, these signals are not free from interferences. Indeed, various factors affect the radio-frequency (RF) waves such as the composition of the surroundings (walls, objects, etc.), the orientation of the transmitters, signals bouncing, etc. From experiments, generated disturbances were observed and analyzed, observing that these phenomena do not completely affect the tendency of the received signal regarding the studied models. Then, it is necessary to develop the estimation system using a mathematical model that approximates the EM waves path loss and an algorithm allowing noise filtering or the estimation considering such alterations. For this task the Extended Kalman filter (EKF) is used as a tool.

Along with estimating the proximity of a pallet, the system handles the operation of the SCORBOT-ER 9 robot manipulator and the control of the FMS's conveyor belt. In addition, we implemented an interface that allows a user to interact with the system. Figure 2 illustrates a general view of the system, exposing the different components and interacting devices.
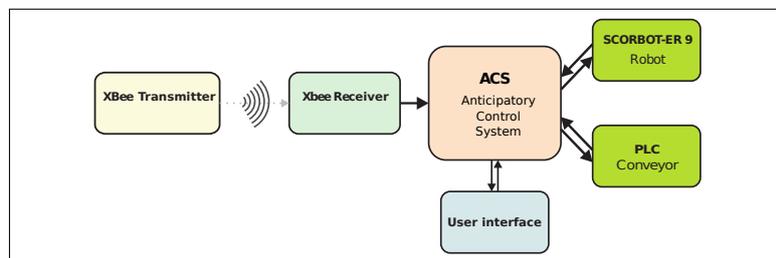

Figure 2: ACS general scheme.

3.1   System model

One of the key aspects on the development of the ACS is to obtain a mathematical model for path loss of the wireless signal, phenomenon related to the distance between the sender and the receiver nodes. Such a system can be modeled considering it as a "gray-box system"because a part of the dynamics of the RF signals is known, existing different prediction models. However, many variables and interactions with the environment are unknown. One of the models used to describe the propagation loss is the "Log-Normal Shadowing", whose equation is given by:

$$PL(d)[dB] = \overline{PL}(d_0)[dB] + 10.\gamma.\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma[dB] \tag{2}$$

were X_σ represents noise and interference as a zero-mean Gaussian random variable with standard deviation σ. In order to find the value of power received by the receiver node, we must substract

this attenuation to the transmission power of the signal Pt, so we have:

$$PL_{rcvd}(d)[dBm] = Pt[dBm] - PL(d)[dB] = -10.\gamma.\log_{10}\left(\frac{d}{d_0}\right) + \left(P_t - \overline{Pl}(d_0)\right) - X_\sigma \quad (3)$$

A relationship between the received strength and the distance between the nodes can be obtained by the propagation model . Along with this variable, it is also necessary to know the speed with which the pallet approaches, thus we must find a relationship between the pallet's speed and its distance to the stop station. An alternative is the derivation of the speed value from distance estimations, however we should consider that this value represents only a measure of the distance change rate and not the actual speed of the pallet. Despite this, both of the values are related, therefore it can be used as an indicator of how quick the pallet moves.

By derivation we obtain the instantaneous speed with which the pallet approaches: $\frac{\delta d}{\delta t} = \nu$, in this work, considered as a constant. This expression is discretized, obtaining the distance for instant $tk + 1$:

$$d(tk + 1) = \nu(tk)T + d(tk) \quad (4)$$

Next, and in order to identify a discrete model parameters, a linear difference equation can by used as $y(t) + a_1 y(t-1) + \ldots + a_n y(t-n) = b_1 u(t-1) + \ldots + b_m u(t-m)$, where $u(t)$ and $y(t)$ are the input and output, respectively, and a_n and b_m are the system's parameters. From the last expression it is possible to determine the value of the next output solving the difference equation for $y(t)$ and predict the system's output from the system's observed measurements and its parameters [14]. Considering a system of N equations and n unknowns, the estimated output can be written as [14]:

$$\hat{Y}(N) = M(N).\hat{\theta} \quad (5)$$

where the variables have, respectively, the structures: $Y(N) = [y(1) \quad y(2) \quad \ldots \quad y(N)]^T$, $\hat{Y}(N) = [\hat{y}(1) \quad \hat{y}(2) \quad \ldots \quad \hat{y}(N)]^T$, and $M(N) = [m(1) \quad m(2) \quad \ldots \quad m(N)]^T$. In order to estimate the parameters of Equation 5, Least Squares method is used. The predictions made using estimated parameters will have an associated error defined as $E = Y(N) - \hat{Y}(N)$. Then, the Least Squares method reduces this error by minimizing a functional which is a sum of squared errors, i.e., $\min\left(J(\hat{\theta}) = E^T(N).E(N)\right)$, with $J(\hat{\theta}) = \sum_{K=1}^{N} e^2(k)$. In order to minimize the error, we must have $\frac{\partial J(\hat{\theta})}{\partial \hat{\theta}} = 0$. Developing for θ from previous equations, we obtain the following expression for obtaining parameters which minimizes the prediction error that can be used to estimate parameters of the propagation loss model :

$$\hat{\theta} = \left[M(N)^T.M(N)\right]^{-1}.M(N)^T.Y(N) \quad (6)$$

Let us consider the propagation model in Equation 3. This can be developed to adopt the structure of Equation 5, by doing:

$$PL_{rcvd}(d) = -10.\gamma.G(tk) + W \quad (7)$$

where $G(tk) = \log_{10}\left(\frac{d(tk)}{d}\right)$ and $W = \left(Pt - \overline{Pl}(d_0)\right)$. For modeling purposes, the random variable X_σ was considered as zero, obtaining a log-normal model. Finally, expressing Equation 7 in matrix form we have:

$$PL_{rcvd}(d) = \begin{bmatrix} G(tk) & 1 \end{bmatrix}.\begin{bmatrix} -10.\gamma \\ W \end{bmatrix} \quad (8)$$

In this way, the variables are: (1) The system output $Y(N) : PL_{rcvd}(d)$, (2) the measurements vector $M(N) : \begin{bmatrix} G(tk) & 1 \end{bmatrix}$, and (3) the parameters vector $\theta : \begin{bmatrix} -10.\gamma \\ W \end{bmatrix}$

In order to generate useful data for estimations, various experiments were performed. In these, the pallets circulate in the conveyor while the robot takes one of the predetermined positions used in previous experiments. These experiments are performed with each of the three transmitting nodes adopted in this work (3 variants of the Digi's XBee family, explained in Section 4). As an example, Figure 3 illustrates the parameters estimation for a XBee wire antenna transmitter. Table 1 presents estimated parameters for each of the used nodes.
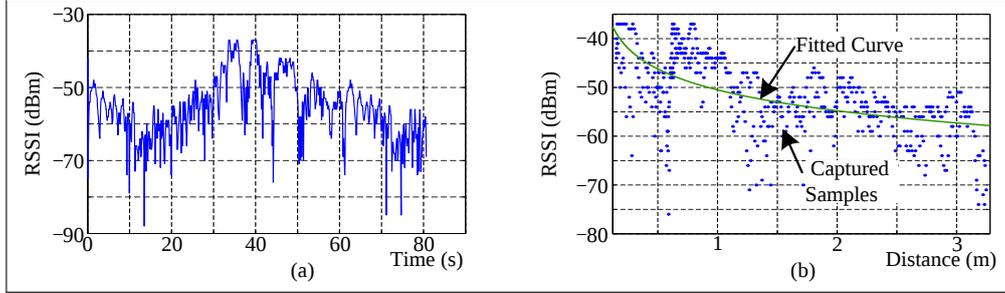

Figure 3: Parameter estimation for a XBee wire antenna transmitter case.

Table 1: Parameters of transmitter nodes.

Transmitter node	Pt	$\overline{Pl}(1.5[m])$	γ
XBee chip antenna	0[dBm]	59.156[dBm]	1.142
XBee wire antenna	0[dBm]	53.0017[dBm]	1.438
XBee Pro wire antenna	16[dBm]	51.376	1.065

The devices sampling time was set to 54[ms], however, the time observed during measurements is not constant. This has a mean deviation of 82[ms] with a standard deviation of 34[ms]. Therefore, the sampling time used in the model will be estimated during the algorithm execution, by the time difference between a RSSI measurement and the previous one. To summarize, the model of the system is given by the following equations:

$$d(tk + 1) \quad = \quad v(tk)T + d(tk) \tag{9}$$

$$v(tk + 1) \quad = \quad v(tk) \tag{10}$$

$$PL_{rcvd}(d) \quad = \quad -10.\gamma.\log_{10}\left(\frac{d(tk)}{d_0}\right) + \left(Pt - \overline{Pl}(d_0)\right) \tag{11}$$

3.2 Signal filtering

For this work the Extended Kalman Filter (EKF) is used. EKF is a variant of the traditional Kalman filter that addresses the problem of estimating states in nonlinear systems. Let us consider the nonlinear model of the process in a discrete-time state variable representation, i.e., representing the unknown state of the system as $x_k = f(x_{x-1}, u_{k-1}, v_{k-1})$, with known input u_k, and the observable output $z_k = h(x_k, n_k)$. v_k and n_k are random variables representing the process and measurement noises, respectively. It is assumed that the distribution densities are Gaussian and both v_k and n_k are random variables independent to each other, so we can define $p(x_{k-1}|z_{1_{k-1}}) = N(X_{k-1|k-1}, P_{k-1|k-1})$, $p(v_{k-1}) = N(O, Q)$, and $p(n_k) = N(O, R)$. However, being a non-linear model, distribution densities $p(x_k|z_{1:k-1})$ and $p(x_k|z_{1:k})$ for next instant k may not be Gaussian, therefore, they are approximated to a Gaussian density. At the densities

and distribution of v_{k-1} and n_k, Q and R are the covariance matrices of process disturbance and measurement disturbance covariance, respectively [15]. So, the filter operation may be divided into two stages:

1. **Prediction.** This stage generates *a priori* prediction of the state $x_{k|k-1} = f(x_{k-1}, u_k, 0)$ and the error covariance $P_{k|k-1} = F_{k|-1}^x . P_{k-1} . (F_{k-1}^x)^T + F_{k-1}^v . Q_{k-1} . (F_{k-1}^v)^T$, considering all the information available at that time.

2. **Correction.** At this stage, previous estimates are corrected, then generating *a posteriori* estimates of $x_{k|k}$ and $P_{k|k}$. For this, we calculate the Kalman gain as $K_k = P_{k|k-1} . (H_k^x)^T . \left[H_k^x . P_{k|k-1} . (H_k^x)^T + H_k^\eta . R_k . (H_k^\eta)^T \right]^{-1}$. This factor minimizes the error covariance of the new state estimation. So, new estimates are calculated as $x_{k|k} = x_{k|k-1} + K_k . (z_k - h(x_{k|k-1}, 0))$ and $P_{k|k} = (I - K_k . H_k^x) . Pk|k - 1$.

Matrices F_{k-1}^x, F_{k-1}^v, H_k^x and H_k^η are obtained as Jacobians of the system's function and of the output function, i.e., $F_{k-1}^x = \frac{\partial f}{\partial x}(x_{k-1|k-1}, u_{k-1}, 0)$, $F_{k-1}^v = \frac{\partial f}{\partial v}(x_{k-1|k-1}, u_{k-1}, 0)$, $H_k^x = \frac{\partial h}{\partial x}(x_{k|k-1}, 0)$, and $H_k^\eta = \frac{\partial h}{\partial \eta}(x_{k|k-1}, 0)$.

The EKF is a recursive algorithm. Therefore, the prediction and correction stages are executed in cycles while the algorithm is running. For more details about an EKF recursive implementation see [16]. On the application of the filter to the developed system it is necessary to obtain matrices F_{k-1}^x, F_{k-1}^v, H_k^x and H_k^η, with respect to the exposed model in Equations 9, 10 and 11. These are defined as:

$$F_{k-1}^x = \begin{vmatrix} \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial d} \\ \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial d} \end{vmatrix} = \begin{vmatrix} \frac{\partial d(tk+1)}{\partial v(tk)} & \frac{\partial d(tk+1)}{\partial d(tk)} \\ \frac{\partial v(tk+1)}{\partial v(tk)} & \frac{\partial v(tk+1)}{\partial d(tk)} \end{vmatrix} = \begin{vmatrix} T & 1 \\ 1 & 0 \end{vmatrix} \quad ; \quad F_{k-1}^v = \frac{\partial f}{\partial v} = \mathbf{I} \quad ;$$

$$H_k^x = \begin{vmatrix} \frac{\partial h}{\partial v} & \frac{\partial h}{\partial d} \end{vmatrix} = \begin{vmatrix} \frac{\partial PL_{rcvd}(tk)}{\partial v(tk)} & \frac{\partial PL_{rcvd}(tk)}{\partial d(tk)} \end{vmatrix} = \begin{vmatrix} 0 & -\frac{10.\gamma}{d(tk).\ln(10)} \end{vmatrix} \quad ; \quad H_k^\eta = \frac{\partial h}{\partial \eta} = \mathbf{I} \quad (12)$$

To initialize the algorithm it is necessary to designate initial values for the estimated states matrix and the error covariance matrix. In case of $x_{k|k-1}$, it is considered that during the system startup the pallet is far from the stop station and outside the area of interest. The pallet's initial speed is considered as zero. For the case of the $P_{k|k-1}$ matrix, it is initialized to a small value different but close to zero, because if the covariance is zero the filter will keep the last estimated value without innovating.

Matrices of process perturbation covariance and measurement are assumed to be constant although in practice this does not happen like that. Values of matrix Q related to the process, are generally more difficult to obtain, but it is presumed that the variance values of the process are low. Matrix R related to measurements can be predicted from the output observable values. From experimentation, a standard deviation of the measurements less than 0.5[dBm] for the case of the chip and wire antenna XBee nodes and less than 2.5[dBm] in the case of XBee Pro nodes, is observed. However, these values, measured in a quasi-stationary regime (i.e. without movements of the node-of-interest and by controlling as many environmental variables as possible), increase in dynamic conditions reaching a deviation of 5[dBm]. Due to this instability in measurements, an intermediate value for the deviation of 2[dBm] is considered. In this way, variances arbitrarily selected for the process variables and measurements are $\sigma_d^2 = \sigma_v^2 = 0.002$ and $\sigma_{PL_{rcvd}}^2$. The disturbance covariance matrices are $Q = \begin{vmatrix} 0.002 & 0 \\ 0 & 0.002 \end{vmatrix}$ and $R = 4$.

3.3 Estimating time of arrival

EKF, together with the discrete system model, allows the estimation of the hidden states of the system. For our case, they will be the variables of distance (between the pallet and the stop station) and the rate at which this changes. Estimating the time at which the manipulator robot must act only with distance information may be appropriate under the studied environment which considers a constant movement speed. However, alterations on this parameter would trigger an erroneous reaction of the system. Therefore, it is necessary to determine the instant in which the manipulator operates by considering both the distance and speed of the pallet estimations.

Estimates of distance and speed can be used together to determine the time a pallet will take to get to the target stop station. We name this estimate *time-of-arrival*.

The preceding variable modeling is made from both experimental data and data obtained through the pallet's path simulation.

By Matlab's surface fitting toolbox we obtain the surface that best fits the available data. This is a 2-order polynomial defined as:

$$T_{arrival}(tk) = p_{00} + p_{10}.d(tk) + p_{01}.v(tk) + p_{20}.d^2(tk) + p_{11}.d(tk).v(tk) + p_{02}.v^2(tk) \qquad (13)$$

where $T_{arrival}$ is the time-of-arrival and d and v are the pallet's distance and speed respectively. Surface parameters p_{xx} are obtained for each of the wireless modules used in the pallets.

3.4 System controller

The instant at which the manipulator robot must act is determined by the time of arrival of the target pallet. The operation of the robot must be done in the instant in which the time the robot employs in executing the approaching routine equals the time of the pallet's arrival. To achieve the above consideration two controllers were implemented.

The first control is an open-loop PI. This was chosen due to the lack of feedback on the system. Its implementation is performed from the discretization of the PI controller given by $y(tk) = K_c\left(e(tk) - e(tk-1)\right) + \frac{K_c}{T_i}T_s e(tk) + y(tk-1)$, where $y(k)$ is the output and $e(k)$ is the calculated error between the controller input and a given reference for a time k, and K_p and K_i are the proportional and integral gains, respectively. The implementation is recursively performed using as a reference the time of the robot operation (T_{OP_Robot}) and the time of arrival of the pallet as input ($T_{Arrival}$). Thus the controller was modeled as follows:

$$e(tk) = T_{OP\_Robot} - T_{Arrival} \qquad ; \qquad a(k) = K_p.e(tk) \qquad ; \qquad b(k) = K_i.e(k) \qquad (14)$$

$$y(k) \;=\; a(k) - a(k-1) + b(k) + y(k-1) \qquad (15)$$

$$a(k-1) \;=\; a(k) \qquad (16)$$

$$y(k-1) \;=\; y(k) \qquad (17)$$

During algorithm's operation, Equation 15 represents an update of the controller output. Equations 16 and 17 are updates of the variables.

The second implemented control was a Fuzzy logic controller that is based on the fuzzy sets theory developed by Lotfi A. Zadeh [17]. This kind of logic intends to represent human logical thinking allowing a system to work with information that is not accurate (e.g., input sentence "the pallet is very close"). The truth degree of a variable x to a "linguistic variable"(e.g., "The pallet is far") is determined by a membership function that takes truth values in the 0 to 1 range. The linguistic variable or label is associated with a fuzzy set which can be defined as: $A = \{x, u_A(x)|x \in Ut\}$, were $u_A(x)$ is the membership function of x and U is the universe of discourse or the exact values that can take the variable [18]. The membership function can

adopt many geometries, such as triangular, trapezoidal, etc. The components of the controller are shown in Figure 4(a), where the *Fuzzification* stage converts crisp inputs or real-world values to a membership degree of a linguistic variable. Later, this variable is evaluated by a set of rules of type: **IF** *{set of conditions to meet}* **Then** *{set of consequences}*. Finally, the obtained fuzzy result must undergo a reverse process or *Defuzzification* in order to obtain a crisp output or an output with useful values for the process, again, using a membership function for the output values.
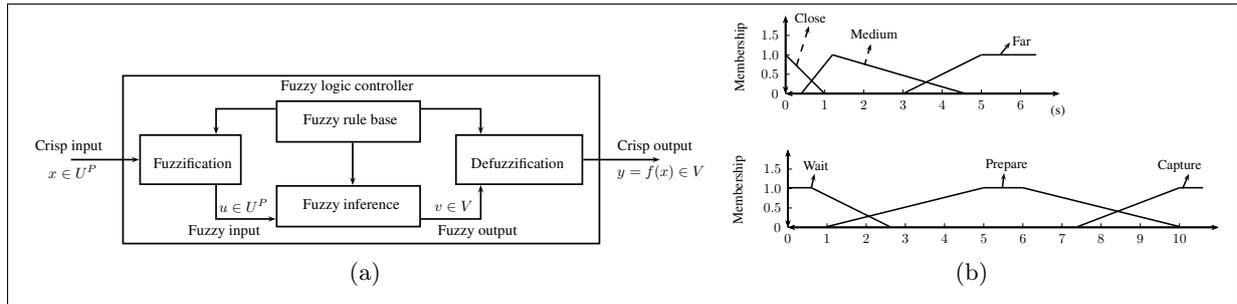

Figure 4: (a) Components of a fuzzy logic controller and (b) the implemented fuzzification (top) and defuzzification (bottom) membership functions.

As in the PI controller, for the fuzzy controller design only one input was considered. This input is the tracking error $e(tk)$ described above. An input range for this variable from 0 to 10 [sec] is established. Besides this, three linguistic variables are defined within the established universe of discourse, "*Far*", "*Medium*" and "*Close*". On the other hand, for the Deffuzification stage, three types of linguistic variables are defined: "*Wait*", "*Prepare*" and "*Capture*". The set universe of discourse corresponds to the range 0-12. Figure 4(b) shows the implemented membership functions.

For the fuzzy inference stage the following set of rules are implemented: (1) **If** $e(tk)$ **is** "*Far*" Then Output **is** "*Wait*", (2) **If** $e(tk)$ **is** "*Medium*" Then Output **is** "*Prepare*", and (3) **If** $e(tk)$ **is** "*Close*" Then Output **is** "*Capture*".

Different controller parameters have been selected from our experience operating the system. The adjustment of membership functions was performed by various assays in the system.

4   Implementation

Pallets moving on the conveyor belt transport wireless beacon nodes periodically sending data packets to a collector node located near the stop station. The collector is directly connected to a Personal Computer (PC) executing the control software. Three types of wireless nodes corresponding to the Digi's XBee family [19] were used in this work. They are: a XBee Chip Antenna, a XBee Wire Antenna, and a XBee Pro Wire Antenna, all from the product's 1 Series. These components are based on the IEEE 802.15.4, and are mainly differentiated by hardware features, such as type of antenna and supported power outputs. In our experiments, the three different XBee modules were tested as beacon nodes. A XBee Chip Antenna module was used as the collector. Nodes main parameters are detailed in Table 2. The rest of the parameters are set to be default parameters recommended by the manufacturer.

Both the robotic arm and the conveyor system are controlled by the ACS software via serial ports. These elements only execute predetermined tasks. The robot has its own operating system. This manages its resources and provides a programming interface for which the user has a set of available functions and operations to work with the robot. Three routines were

Table 2: XBee modules configuration.

	Collector node	Beacon nodes
Device	XBee Chip Antenna	XBee Wire/Chip Antenna - XBee Pro
IEEE 802.15.4 Role	Coordinator	End device
Interface speed	57600[bps]	
Sampling rate	-	54[ms]
Operation mode	API	
Power output	0[dbm]	0[dbm]-16[dbm]
Power source	USB	Batteries

programmed in its controller:

(1) **Standby:** places the robot in one of four positions previously recorded.

(2) **Approaching target:** moves the robot from any position to a position near the point of capture of the pallet (over the stop station).

(3) **Target capture:** allows the robot handling the template carrying the target pallet.

The conveyor system is controlled by a PLC with a resident program. This executes the pallet control services at the stop stations (stop, release, or identify a pallet over a stop station). The services are provided via serial communication software to the ACS. This must extract the relevant information from the data frames sent by the PLC and to generate the necessary commands to perform the required actions.

On the PC, the core of the ACS is a Matlab-based software implementing the filtering and control routines, and serial i/o and user interfacing. The ACS software aims to estimate the proximity of the objects of interest (the pallets) and to determine the instant in which the system must generate the necessary commands to anticipate the arrival of the pallet, using as input only RSSI traces. The software components are summarized in Figure 5a.

The operations sequence cyclically executes the following steps: (1) devices reading, (2) estimation algorithm execution, and (3) device status update. This sequence is repeated until achieving the target pallet capture. The operating cycle of the program is initiated by a user requirement. The software starts the cycle by reading the RSSI value and the transmitting node id, from the assigned serial port (these data are delivered by a C++-based program who actually reads and decodes packets from the XBee collector). Then, the data are filtered before being processed by the estimation algorithm. Once the arriving time is estimated, its value is entered to the system controller. If the estimated time does not equal the reference time, the current system's output state remains and the cycle is repeated. On the contrary, the controller gives the order to start the operations of approaching and capturing of the target pallet.

To summarize, Figure 5b illustrates the implemented algorithm stages.

5   Tests and Results

At the begging of the system's integration stage, parameters are adjusted in order to achieve a proper system's operation. System calibration is performed by contrasting experimentally obtained information and simulations of system's models. For obtaining real data from the system, we registered RSSI values for each node considering different positions of the manipulator robot (as discussed in previous section). With these data we simulate the behavior of the estimation and control algorithm under adjusted system parameters. These parameters are: Reference power $PL(d_0)$, Path loss exponent γ, Standard deviation of the process disturbance σ_{pp}, and Standard
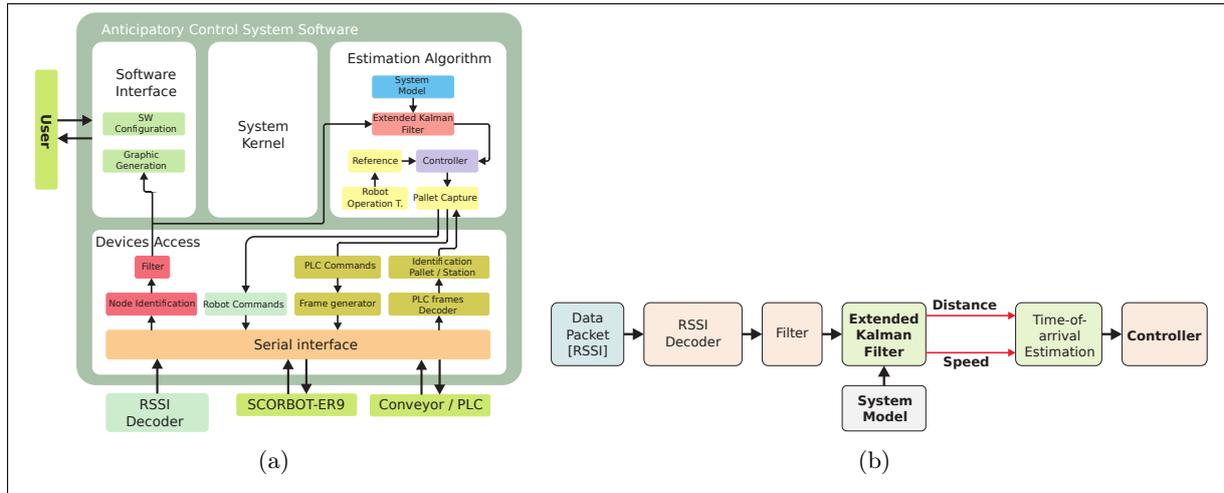
Figure 5: ACS (a) software components and (b) general scheme.

deviation of measurement disturbance σ_{pm}. System parameters calibration was performed by the Greedy algorithm [20]. Table 3 shows the new parameters obtained.

Table 3: Adjusted system parameters.

Node	Pt	$\overline{Pl}(1.5[m])$	γ	σ_{pp}	σ_{pm}
XBee chip antenna	0[dBm]	59.490[dBm]	1.949	0.0034	4.090
XBee wire antenna	0[dBm]	53.838[dBm]	3.186	0.0025	4.818
XBee Pro wire antenna	16[dBm]	52.160[dBm]	2.720	0.0023	6.0

After calibration, we performed an execution test in order to verify the estimation algorithm parameters for each beacon node. During this test, pallets are not stopped at the stop stations. Figure 6(a) shows registered RSSI values for each node. Figure 6(b) and 6(c) show estimations of time-of-arrival and distance for each node.
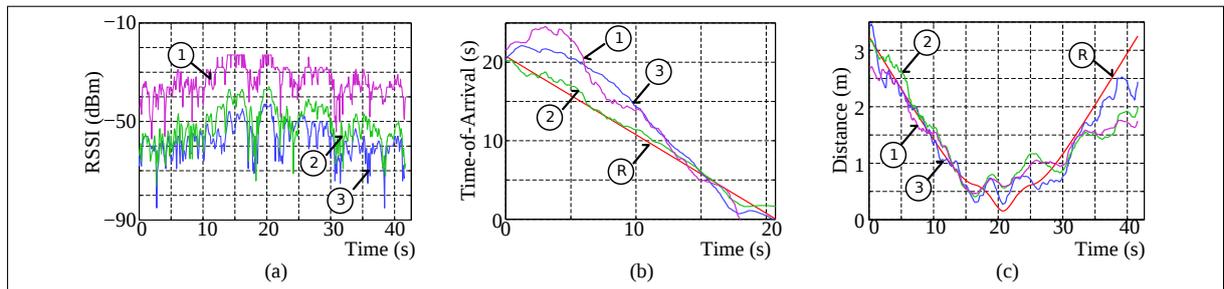

Figure 6: Estimation with calibrated parameters for ① XBee A. Chip, ② XBee A. Wire and ③ XBee Pro A. Wire. In (a), input captured data. In (b) and (c) show Ⓡ real arriving time and distance, respectively, and estimations for the three nodes ① XBee A. Chip, ② XBee A. Wire and ③ XBee Pro A. Wire.

We observe that the best distance estimate is performed on the first half of the workspace represented by the first 21 seconds in Figure 6(c). In Here, the pallet approaches the stop station reaching the shortest distance between the transmitter and the collector nodes (approximately at second 21). Then, the pallet moves away.

Now, in order to evaluate the performance of the ACS, several tests under different operating conditions were conducted. The objectives of these tests are: (1) to compare system performance for different transmitting nodes, (2) to compare performance of the system under different initializations of the error covariance matrix, (3) to measure overall system performance, and (4) to observe the effect of sampling rate variations in the system estimates. To accomplish these objectives, we performed both an individual and a global performance tests, and a test to observe the degradation of the estimation due to sampling rate variations. In each test estimates of distance, time-of-arrival and time of capture of the pallet are recorded. This last value is defined as the time between the arrival of the pallet to the stop station and the time the robot takes to get the catching position. A good result for the capture time is a near zero (seconds) value. The performance metrics of the system are: the Distance Absolute Error Estimation, the Time-of-Arrival Absolute Error Estimation, and the Absolute Capture Time Absolute Error.

In the individual performance test, we compare ACS performance considering the use of different beacon nodes. Firstly, individual beacons are located on a template, then we let them move over the conveyor. Individual tests are performed with the three types of XBee nodes and different initial positions of the robotic arm. In each case, ACS executes a full cycle allowing the robot to capture the template. Along with this, we want to observe the effect of the initialization of the error covariance matrix (P). For this, we executed two versions of the estimation algorithm: (Version A) the original version of the algorithm, initializing the error covariance matrix to a near zero value, modifying its value as tests succeed, and (Version B) a modified algorithm, where the error covariance matrix is initialized with a default value. This value is obtained after successive runs of the original algorithm. During each execution of the experiment the value of the array is initialized to this value. Average results for these tests are shown in Figure 7. In the global
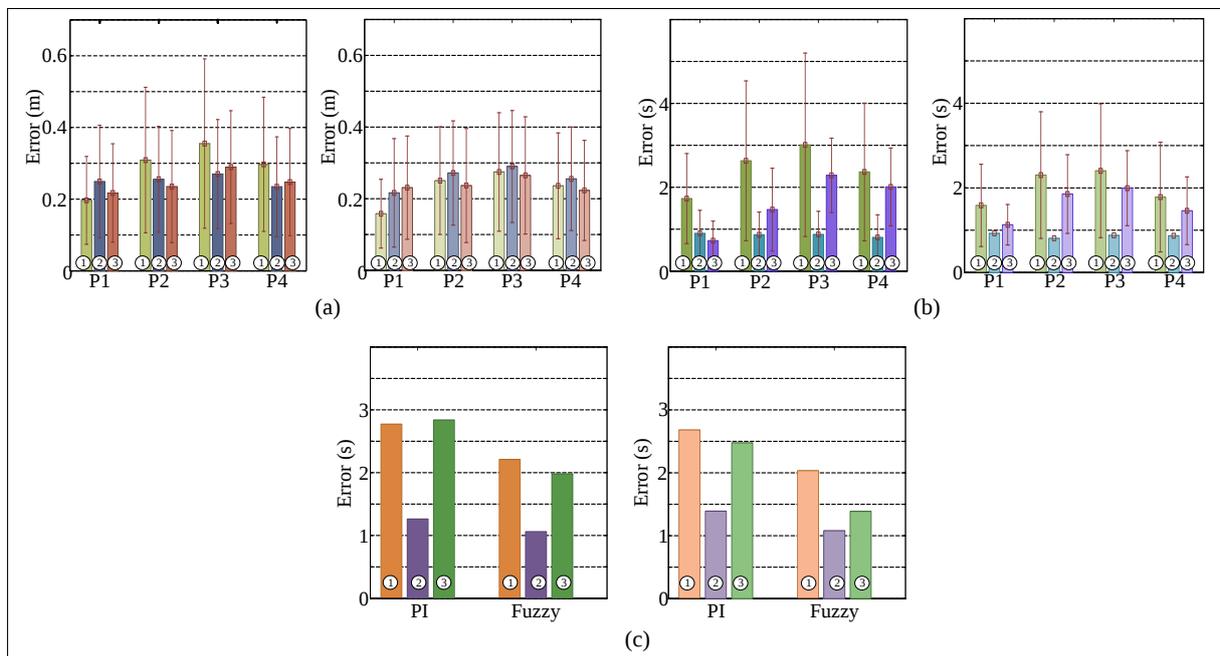

Figure 7: Average results for the individual performance tests with the tree wireless nodes ① XBee Antenna Chip, ② XBee Antenna Wire and ③ XBee Pro Antenna Wire. In (a), Absolute Error of the Estimated Distance with both versions of the filter A (left) and B (right). In (a), Absolute Error of the Estimated Time-of-Arrival with both versions of the filter A (left) and B (right). In (c), Absolute Error of the Capturing Time with both versions of the filter A (left) and B (right).

performance test, we observe the performance of the system under normal operating conditions.

This involves capturing any of the three available pallets on the system from a random position as the starting point for the robotic arm. For this test, we use the three adopted types of nodes as beacons transmitting simultaneously. We compare the performance of each node individually with the performance of the system to a random selection of the target node. The selection of the initial position of the robot is performed at random for each test. As in the previous test, we observe the effect of the P matrix initialization. For each configuration of the experiment 10 tests are performed in a row. Average results for these tests are shown in Figure 8. Finally, we study
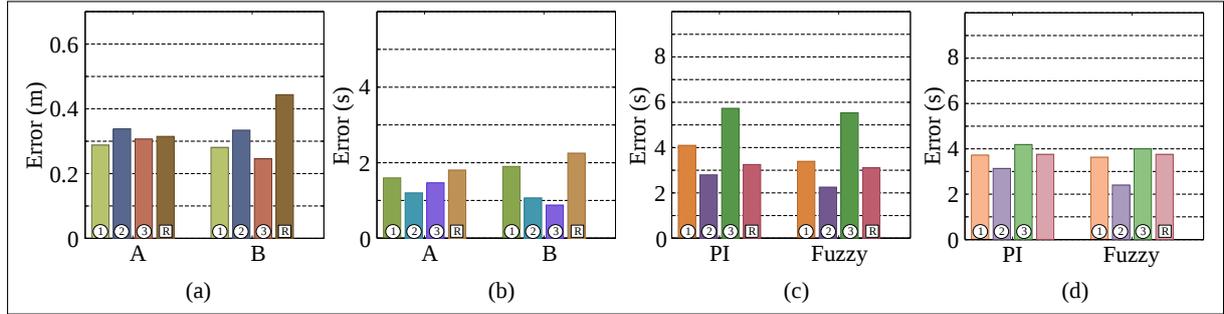

Figure 8: Average results for the global performance tests for the three adopted nodes ① XBee Antenna Chip, ② XBee Antenna Wire and ③ XBee Pro Antenna Wire, and a ⌷R⌷ random selection for the target node. In (a) and (b), Absolute error of the estimated distance and of the estimated time-of-arrival for both of the filter versions A and B. In (c) and (d), the absolute error of the capturing time for both filter version A and B, respectively.

the degradation of the estimation produced by reductions on the sampling rate. This test is performed by using the same Beacon node (a XBee Chip Antenna) while the manipulator robot adopts a same initial position for each of the experiments. The experiments consider increases 100[ms] in the given sampling rate from an initial value of 100[ms] and a final test value of 1100[ms]. Average results for these tests are shown in Figure 9(a), 9(b), and 9(c).
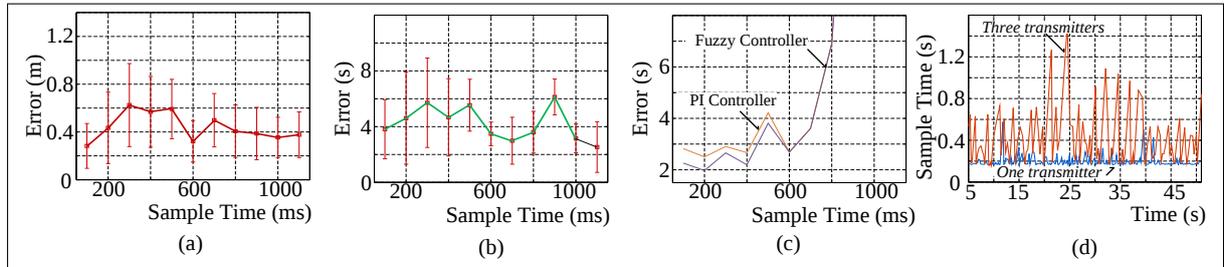

Figure 9: In (a) and (b), the absolute error of the estimated distance and time-of-arrival, respectively. In (c), the absolute capture time shows the absolute error of the capturing time with respect to the sampling time. In (d) the effect on the sampling time of increasing the number of transmitting nodes.

From the results, we can observe a system performance decline when including a larger number of simultaneously operating transmitter nodes (see Figures 7(a) and 8(a)). Note that the operation mode of the transmitting nodes causes dispute over the transmission medium. The information received from these nodes is at random, without an order in the delivery. This causes instability and decrease in the RSSI sample rate, resulting in an increased error in the estimation of distance which grows in a 18.3 %. Figure 9(d) illustrates this effect. The figure shows the sampling rate for XBee Chip Antenna nodes operating independently and the rate obtained in a scenario with 3 nodes transmitting simultaneously.

Regarding the performance of the system, this is maintained within acceptable parameters by using only one node Beacon, observing errors in distance estimation and in the time-of-arrival of

$25[cm]$ and $1.5[s]$ around, respectively. This allows the system to meet the design requirements by obtaining a capture time of less than $3[s]$ ($2[s]$ average).

The last performed test allows us to determine the maximum sampling rate supported by the system. This is $600[ms]$ for the case of an individual node. In case of using a larger number of nodes this rate should allow increasing the number of nodes without significantly affecting the system performance. However, a way to synchronize the senders to the collector node is necessary. A sampling time greater than the determined one can cause the system not to properly determine the area of operation of the controller. This affects the pallet capturing algorithm, causing it not to stop in a timely manner so the pallet flows without stopping at the corresponding stop station and must wait for a new approach to perform the capture. This is observed in Figure 9(c) in which higher values to the determined one produces excessive time increases in pallet capture. Moreover, results show that there is a considerable improvement in the obtained results when initializing the error covariance matrix to a preset value (Version B of the algorithm). In general, we observe a decrease in errors estimates close to 8%. In the case of capture time we observe an improvement of 10% by using the version B of the algorithm. Regarding the performance of the controllers employed in the system, better results are observed when using the fuzzy controller in all of the implemented tests. The fuzzy controller has reduced capture times compared to PI controller by 27% in the individual performance test and 10% in the overall performance test.

Finally, we observe the performance of the system considering the three employed nodes. From these, the best results, considering the best estimate of time-of-arrival and time of capture of the pallet, are obtained by applying the XBee Wire Antenna module. This is observed in Figures 7(b) and 7(c).

6   Conclusions and Future Works

This paper presents the development and implementation of a proposed control scheme which is able to trigger an anticipatory control action. For the above, RSSI measurement from a wireless sensor mounted in a moving pallet are used along with an EFK in order to estimate the speed and distance of the moving pallet relative to a stop station.

The control scheme is proposed for short distance in order to avoid many factors which affect the signals behavior (bouncing, multipath, etc.). Specifically for a 2.25 m radius case the control scheme is able to estimate the time of arrival of the pallet traveling at 0.19 m/s and to generate the necessary control action (for the case presented the movements of a robotic arm) before it reaches the stop point. Due to the lack of a feedback loop for the robotic arm used, a PI and Fuzzy controller are implemented, obtaining a better response by the fuzzy strategy with a 54[ms] sampling time.

Due to RSSI instabilities it is necessary to avoid significant changes in the environment in order to estimate correctly speed and position of the pallet. Future work will focus in the RSSI instabilities compensation.

Acknowledgment

Bibliography

[1] I.F. Akyildiz,W. Su, Y. Sankarasubramaniam, E. Cayirci (2002); Wireless Sensor Networks: A Survey, *Computer Networks*, 38(4), 393-422.

[2] K.S. Low,W.N.N. Win, M.J. Er (2005); Wireless Sensor Networks for Industrial Environments, Computational Intelligence for Modelling, Control and Automation, *2005 and Int. Conf. on Intelligent Agents, Web Technologies and Internet Commerce*, 2: 271-276.

[3] V.C. Gungor, G.P. Hancke (2011); Industrial Wireless Sensor Networks, In *Industrial Electronics Handbook*, 2nd ed., CRC Press and IEEE Press.

[4] L. Krishnamurthy et al. (2005); Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the North Sea, in *3rd Int. Conf. on Embedded Networked Sensor Systems*, 64-75.

[5] J. McCulloch, P. McCarthy, S.M. Guru, W. Peng, D. Hugo, A. Terhorst (2008); Wireless sensor network deployment for water use efficiency in irrigation, in *REALWSN'08: Proc. of the workshop on Real-world wireless sensor networks*, ACM, 46-50.

[6] Z. Butler, P. Corke, R. Peterson, D. Rus (2006); From robots to animals: Virtual fences for controlling cattle, *International Journal of Robotics Research*, 25(5-6): 485-508.

[7] A. Zouinkhi, A. Ltifi, M. Ben Gayed, N. Abdelkrim, E. Bajic, E. Rondeau (2010); Simulation of active products cooperation for active security management, *8th Conf. Int. de Modélisation et Simulation (MOSIM'10)*, Hammamet, Tunisie.

[8] D. J. Cook, J. C. Augusto, V. R. Jakkula (2009); Review: Ambient intelligence: Technologies, applications, and opportunities, *Pervasive Mob. Comput.*, DOI=10.1016/j.pmcj.2009.04.001, 5(4): 277-298.

[9] D. Dobre, G. Morel, D. Gouyon (2010); Improving human-system digital interaction for industrial system control: some systems engineering issues, In: *10th IFAC Workshop on Intelligent Manufacturing Systems, IMS'10, Lisbone : Portugal*, 1-7.

[10] S. E. Taylor, T. P. McDonald, M. W. Veal, T. E. Grift (2001); Using GPS to evaluate productivity and performance of forest machine systems, *First Int. Precision Forestry Symposium*, 1-20.

[11] Ubisense. `http://www.ubisense.net/`

[12] C. Duran-Faundez, C. Aguilera-Carrasco, A. S. Norambuena, (2010); Experimenting with RSSI for the perception of moving units in intelligent flexible manufacturing systems, *2010 IEEE Int. Conf. on Industrial Technology, ICIT2010*, 1400-1405.

[13] Laboratorio de Sistemas Automatizados de Producción CIMUBB, Universidad del Bío-Bío, `http://www.ubiobio.cl/cimubb/`

[14] L. Ljung *System Identification Theory for the User*. Prentice Hall PTR. 1999.

[15] G. Welch, G. Bichop (2006); *An introduction to the Kalman Filter. Reference Material*, Univ. of North Carolina at Chapel Hill, Dept. of Computer Science, 1-16.

[16] R. Grover Brown, P.Y. C. Hwang (1997); *Introduction to random signals and applied Kalman Filtering with Matlab: exercises and solution*, 3rd ed.. USA: John Wiley & Sons.

[17] L.A. Zadeh (1965); Fuzzy sets, *Information and Control*, 8 (3): 338-353.

[18] A. M. Ibrahim (1997); *Introduction to Applied Fuzzy Electronics*, Prentice Hall.

[19] Digi International `http://www.digi.com/`

[20] T. H. Cormen, C. E. Leiserson, R.L. Rivest, C. Stein (2009); *Introduction to Algorithms*, (3a. Ed.) Massachusetts: Massachusetts Institute of Technology.