International Journal of Innovative Computing, Information and Control
Volume **15**, Number **4**, August **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **1223–1241**

OVERHEAD CRANE: AN ADAPTIVE PARTICLE FILTER FOR ANTI-SWAY CLOSED-LOOP AND COLLISION DETECTION USING COMPUTER VISION

Roberto Paolo Luigi Caporali

Mathematics for Applied Physics di Roberto Caporali
P.O. Box 1255176, Roma Prima Porta, Rome 00188, Italy
robertopaolo.caporali@gmail.com

Received October 2018; revised February 2019

Abstract. *This paper presents a novel method for controlling anti-sway in closed loop and, at the same time, for detecting possible fault and collisions of an Overhead crane load, using computer vision tracking and an adaptive particle filter. The proposed work is based on an innovative statistic method relative to an adaptive particle filter. It is used to have an estimation of the camera images values waiting for the images updating arriving from the field-bus. In this way, the crane load position can be estimated and so the crane load can be controlled in close-loop. Furthermore, the innovative statistic method allows detecting collisions of the Overhead crane load through the machine vision system. The "adaptive" particle filter uses a dynamical model of the anti-sway crane load system and an alternative dynamical model that describes the crane load when a collision occurs. A Markov model is formulated where one of the models describes the crane dynamics. Moreover, all the statistic model is developed on the PLC, in this way obtaining a substantial reduction of the time latency for the camera vision. The characteristics of the proposed method are implemented and tested.*
Keywords: Adaptive particle filter, Anti-sway, Computer vision, Closed-loop control, Collision detection, Overhead crane

1. **Introduction.** This work is related to a method for obtaining two kinds of results: control in closed-loop the sway of an Overhead crane load and detect possible fault and collisions of the same crane load. That is possible using visual tracking and a statistic particle filter computed directly on the PLC to have a faster answer from the crane system control. The anti-sway crane control problem can be solved either using open-loop control system generating a trajectory of the payload able to minimize its sway, or using the feedback of the swing angle of a payload (closed-loop control). The recent work of Ramli et al. [1] allows having an exhaustive literature review of the strategies relative to the crane control. In the limits of the open-loop control system, the recent works of the author of this paper, relatively to anti-sway for Overhead and gantry cranes [2] and relatively to anti-sway for slewing cranes [3], were developed using a command smoothing method. Nevertheless, the other technique, that is the anti-sway closed-loop technique, ensures the robustness against the parameters variation and disturbances, while causing several troubles with the implementation of the sway measurement system. Therefore, the core of the anti-sway closed-loop control in industrial practice is the reliable method of measuring swing angle of a payload suspended on a rope. In recent years, many papers were published relatively to the solution of the payload sway for an Overhead crane with feedback control that is in closed-loop control. Mainly, those solutions were relative to systems where the trajectory control was automatic or semi-automatic and not with

DOI: 10.24507/ijicic.15.04.1223

the control of the human operator. Remaining to consider only the most recent works relative to closed-loop control, we examine only some of the most important among them. One of the linear control techniques that were actively applied for the crane systems is the Proportional Integral Derivative (PID) control. In this regard, the recent paper of Ahmad et al. [4] developed a method which includes the root locus technique and the paper of Jaafar et al. [5] defined a computational intelligence method, the Particle Swarm Optimization (PSO). A feedback control system for suppressing crane oscillations with on-off motors was developed by Heckman and Singhose [6]. Hamid et al. [7] investigated a solution of the closed-loop control with a neural network-based PID. A solution using a stereo-vision system and a state feedback controller was developed by Inukai and Yoshida [8]. Peng et al. [9] used a machine vision system and a combined hand-motion control. Recently, Hyla [10] defined a measurement method of the payload sway angle based on a single camera, to improve a solution with non-contact vision swinging sensor.

Besides, from the second point of view, the present work allows also to detect a fault. That is very important in automated industrial processes to minimize equipment (crane) damage. Fault detection, due to the increasing requirements of safety, reliability, and performance of control systems, is a conventional feedback control design for a sophisticated system, above all for crane systems controlled in automatic mode.

In automated crane systems, the consequences of a fault in a component or any loss of system functionality can also be catastrophic. If we define a fault as an unpermitted deviation of at least one characteristic property of a system, fault detection means to be able to determine whether faults are present in a system. The first step for realizing a fault control system is to diagnose and detect faults. Therefore, the system designers started to define Fault Diagnosis and Detection (FDD) functions for the system. In a general sense, there are two approaches to achieve FDD: an analytical approach or a heuristic approach. The analytical approach can consist of two alternative methods: signal-processing-based or model-based [11]. The signal-processing-based method considers the time domain (statistical) and frequency domain features of output signals. On the contrary, model-based analytical FDD method is a system model where usually a statistic state estimator, such as a Kalman filter, is considered.

Nevertheless, in most current industrial applications, vision-guided robots cannot support a lot of new emerging demands. The problem to solve is the visual feedback speed. In fact, due to the cost limit, industrial robot vision systems are subject to considerable latency and limited sampling rate. Going deeper, we can say that the considerable latency of the answer to the vision control is a consequence of the following fact. If camera images are used to detect possible crane collisions, it is verified that there is a very high difference between the update rates of the camera images (due to the camera hardware itself and the bandwidth available for image transfer) and the update rates of the crane's controllers. Typically, the update rates of the controllers have a range from 125 Hz to 1 kHz, while cameras have lower update rates. The images are often transferred using Ethernet or USB, but that provides no guarantees of real-time performance. The present work establishes a new way of processing camera images, computing directly on the PLC (and not in an auxiliary control station) the statistic method used for handling the images. A method for detecting a fault in a non-linear system with a fault mode parameter r_k and using a particle filter is presented in the work of Zhao and Skjetne [12]. The same method was applied for tracking and detecting a fault of a Remotely Operated Underwater Vehicle (ROV) in the paper presented by the same Zhao et al. [13]. Finally, the paper of Myhre and Egeland [14] proposed to fit the method in [12] to obtain a method for visual tracking and collision detection relatively to crane loads.

Really, in recent years a high number of jobs have been developed using the particle filter method. Mainly, we can cite, as examples of using of the particle filter method in different fields of application, the works of Cho et al. [17], Van Leeuwen et al. [18] and Xue et al. [19].

The present work defines a method to control in closed-loop the crane sway angle based on a statistic measurement method ("adaptive" particle filter) that extrapolates the position feedback of the load by the camera images. In the actual work, the model-based analytical FDD idea is implemented for extending the existing model-based methods for industrial cranes into a more general method, by using the advantages of an "adaptive" particle filter.

In a general way, this novel method proposes new algorithms and, above all, the direct computing of the vision control directly on the PLC, reducing the latency and slow sampling of visual feedback strongly. So real-time vision-guided robot control can be realized with high performance. In this paper, starting from the proposed method in [14], a novel method is developed using an adaptive particle filter to obtain two results: detecting a collision of the crane load and obtaining the statistic feedback position of the crane load during the anti-swaying movement of the Overhead crane. The adaptive particle filter is a statistical method that we use in order having an estimation of the values of the camera images during the time latency of his updating. Using this method is possible to compare the nominal dynamical model that describes the motion of the crane load controlled with an anti-sway system with an alternative dynamical model that describes the crane load under collision. A first-order Markov chain is the model used for the system mode transitions. The adaptive particle filter algorithm is developed to estimate the system states. At the end of the prediction-updating cycle, the statistic method is able to detect whether the object motion is due to the nominal model or the alternative model relative to a collision. A resampling method is developed to obtain an efficient answer of the adaptive particle filter. The particle filter estimates the load position from a sequence of camera images where a defined image on the crane load is tracked. A potential benefit of using a dynamical model based on the physical equations of motion is that the assumed noise level in the adaptive particle filter can be significantly reduced. The method is validated using simulation software.

This paper is organized as follows. The statistical model is presented in Section 2, describing in detail the used adaptive particle filter and the resampling method. In Section 3, the dynamical model of the Overhead crane with position feedback by the camera vision is described. From Lagrange equations, the dynamical equation relative to the sway angle is obtained, highlighting the closed-loop solution with the PID corrections. In the same section, using the statistic method, collision detection is considered. For this purpose, an additional fault parameter that evolves following a Markov model is calculated. In Section 4, the camera calibration is described, with the detailed computation of the intrinsic and extrinsic camera parameters. In Section 5, an implementation of the method and the most relevant results of the model simulation are presented, emphasizing the fact that closed-loop feedback is obtained changing many times the velocity set, so simulating the operator manual control. Besides, also the situation in which a fault (calculated by the adaptive particle filter method) occurs is shown and simulated. At the end, in Section 6 concluding remarks are defined.

2. **Statistical Model: Adaptive Particle Filter.**

2.1. **Reasons for the statistic model.** The present work defines a method to control in closed-loop the crane sway angle and to detect possible collisions based on a statistic

measurement method ("adaptive" particle filter). That method extrapolates the position feedback of the load by the camera images.

The reasons for which a statistical method and, consequently, a statistical model is required to describe the system in question are the following two. First, we must consider the different refresh cycle time and control cycle time to which are subject, on one side, the camera connected to the controller and, on the other side, the controller itself. That has been largely described in Section 1. Second, it is necessary to consider that the camera supplies images that typically have a resolution of 1024×768 pixels, which means about 800,000 pixels have to be transmitted to get an image. Therefore, in order to predict the temporal variation of the image with a frequency comparable to that of the controller cycle (some ms), it is necessary to statistically sample the pixels of the image set, building a representative sample of the image itself of (for example) 50,000 items. Through them, the sample set representing the state of the system (crane movement) in time evolution is described.

2.2. **Filter problem.** In order to define the problem of filtering, we introduce a general form to describe a dynamic system with nonobservable components, using a discrete succession of time instants. This general form consists of two equations: a first equation describes the evolution of the state of the system (state equation), a second equation describes the relationship between an observable variable and system state.

Bayesian filters are mathematical tools estimating the x status of a dynamic system through sensory measures. In the case of a mobile robot (as an automatically controlled crane), the dynamic system is represented by the robot-environment set, the state and the position of the robot, and the measures include sensory readings. The key idea of the Bayesian theory is to determine the posterior probability density function of the state based on all available information. This posterior distribution is typically indicated with "Belief" and is defined as:

$$Bel(x_t) \equiv p(x_t|z_{1:t}), \tag{1}$$

where the $z_{1:t}$ represent the perceptual data, what are the measurements of lasers or cameras. Furthermore, Bayesian filters assume that the environment is Markovian, that is past and future data are considered independent of each other if the current state is known. Bayesian filters calculate the probability density recursively because for many applications (as in the present work) an estimate is required at each instant when data is received from the sensors. As a consequence, using the Bayes rule and the first order Markov hypothesis, we can obtain the recursive equation:

$$Bel(x_t) = \eta p(z_t|x_t) \int p(x_t|x_{t-1}, z_{t-1}) Bel(x_{t-1}) dx_{t-1}, \tag{2}$$

where with η indicating the normalization constant:

$$\eta \equiv \{p(z_t|z_{1:t-1})\}^{-1} = \left\{ \int p(z_t|x_t) p(x_t|z_{1:t-1}) dx_t \right\}^{-1}. \tag{3}$$

Equation (2) is the recursive update equation used in Bayesian filters and, together with the initial belief, defines a recursive estimator of the state for a partially observable system. In order to implement Equation (2), we need to know two conditional distributions. The first one: $p(x_t|x_{t-1}, z_{t-1})$ is obtainable from the evolution model (or motion model) and describes how the system evolves in the time, while the second one: $p(z_t|x_t)$ is the observation model (or sensor model) and represents the density of the z measurement given the x status of the system. In the case of observation of non-linear non-Gaussian dynamic systems, there is no analytical solution in closed form. Consequently, a possible

strategy for solving the filtering problem is applying the Monte Carlo methods iteratively. These methods solve the estimation problem by providing an approximation of the probability density at a posteriori through a discrete probability distribution. An efficient way to represent the probability distribution is the particle filter, based on the concept of importance sampling. In practice, a particle filter is a Bayesian filter in which the probability distribution is calculated recursively using Monte Carlo simulations. The basic idea is to represent the probability density function a posteriori through a series of samples (or "particles") to which a weight is associated and calculate the estimate through the samples and weights themselves. This model represents an approximation of the effective density; however, by increasing the number of samples suitable for representation, the Monte Carlo technique tends to the real density and the particle filter to the optimal Bayesian filter. Each particle i is described by a pair of values $(x_i; W_i)$: the first is a vector containing the state of the particle, while the second is an index used to indicate the importance assigned to it. The posterior density probability function, $p(x_t|z_{1:t})$, is represented by the distribution of the corresponding particles and weights. The latter are chosen on the basis of the importance sampling principle. The general idea is to use for sampling a so-called function of importance (importance density). The idea is that of emphasizing, among the random variables that it receives, those that have a greater influence on the estimated parameters. Therefore, due to the complexity of the $p(x)$ distribution, in its place we often use a simpler $q(x)$ distribution, called proposal distribution or importance distribution, which approximates the behavior of the $p(x)$. In this way, the estimator variance is reduced, highlighting the samples with the highest weight. The importance of the samples is "weighing" depending on the likelihood of the values obtained with the measurements. So, we can build a discrete approximation of the posterior probability distribution:

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^{N} W_t^i \delta(x_t - x_t^i). \tag{4}$$

The weights W_t^i, relative to the i-particle considered at instant t, are scalars that are obtained from the appropriate weight renormalization. The most commonly encountered problem using the Sequential Importance Sampling (SIS) algorithm is the problem of degeneration (weight degeneracy): after some iterations, most of the particles (samples) take on weights next to zero. That is since weight variance can only increase over time. A solution to the degeneration phenomenon is to duplicate the particles (samples) with a high normalized weight, discarding those with negligible weight. That is the so-called resampling technique, of which there are various possible versions. In the present work, we use the method of "Select with Replacement". The introduction of resampling gives rise to the Sampling Importance Resampling (SIR) algorithm: at each iteration, the particles with low weight in Equation (4) are replaced by new particles selected according to a specific distribution. The resampling step changes the (4) into:

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^{N} \frac{n_t^i}{N} \delta\left(x_t - x_t^i\right), \tag{5}$$

where n_t^i defines the number of times when the sample x_t^i is selected. We also have, of course: $\sum_{i=1}^{N} n_t^i = N$. Therefore, the samples x_t^i will be approximately distributed according to the density $p(x_{0:t}|z_{1:t})$. In the SIR iterative algorithm, after the resampling phase, the next state, at time $t + 1$, must be calculated. For this purpose, the set of particles is considered as constituted by a set of new equiprobable particles (so, with weight equal to 1). In other words, if the particle "i" in the state at the time t, has weight

n_t^i, in the state $t+1$, n particles with initial weight 1 will correspond to that particle. However, resampling brings a new problem. In fact, using the SIR method, the samples that have a high weight are statistically selected many times at every step, inevitably causing that the algorithm suffers from loss of diversity (sample impoverishment). A choice, very convenient in order to minimize the loss of diversity, consists in choosing the proposal distribution $q(x)$ equal to the probability distribution in the previous step (called prior distribution):

$$q\left(x_t|x_{t-1}^i, z_t\right) = p\left(x_t|x_{t-1}^i\right). \tag{6}$$

In this way, we obtain the following recursive equation of weight update:

$$\widehat{W_t^i} = W_{t-1}^i p\left(z_t|x_t^i\right), \tag{7}$$

where the new normalized weights at time step t are given by:

$$W_t^i = \frac{\widehat{W_t^i}}{\sum_{i=1}^N \widehat{W_t^i}}. \tag{8}$$

2.3. **Adaptive particle filter.** With this variant of the particle filters, we want to limit the error introduced by this method, because it supplies an approximation of the real probability function. The goal is to determine, at each iteration of the algorithm, the number of particles in such a way that, with probability $1-\delta$, the error between the real probability distribution and the calculated approximation is less than a certain ε. To derive this, we assume that a discrete and constant piecewise distribution describes the real probability distribution. Thus, for this representation, it is possible to determine the number of samples such that the distance between the estimate of the maximum probability obtained by sampling and the real distribution does not exceed a predetermined threshold ε. The mentioned distance is measured by the idea expressed by Kullback-Leibler (known as KL-distance) and by the corresponding algorithm. The KL-distance is given by:

$$K(p,q) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)}. \tag{9}$$

This distance is never negative and is zero if and only if the two distributions are identical. We suppose that N_p particles are derived from a discrete distribution, characterized by k different bins and p indicates the probability of each bin distribution. Using N_p samples (particles) and considering the estimate $\hat{p}$ of the maximum probability of p, it is possible to prove the following sentence. If we choose

$$N_p \equiv \frac{1}{2\epsilon}\chi^2_{k-1,1-\delta}, \tag{10}$$

where χ_k^2 is referred to as k-th quantile of a general chi-square distribution, it is guaranteed that, with probability $1-\delta$, the KL-distance between the estimate $\hat{p}$ of the probability and the real distribution is less than ϵ. In order to include the result just obtained in the algorithm of the particle filters, it is not necessary to determine the entire discrete distribution, but it is sufficient to calculate the k number of bins. Although this quantity is not initially known, all samples are generated from the a priori distribution, and the value of k is estimated during sampling. Calculating the k value and using (9), we can obtain the optimum value of N_p.

3. **A Dynamical Model of the Overhead Crane with Anti-Sway and Position Feedback.** Once the statistical model used to determine the real position of the crane payload has been clarified, it is necessary to define the mathematical model through which the crane movement is described.

An industrial crane (gantry or Overhead traveling crane) includes a multi-body system with 3 independent degrees-of-freedom for positioning a pendulum suspended by spreader bar to the Trolley. Specifically, the industrial crane includes a translating load-line, having length L, and a payload attached to a moving or translating trolley. Therefore, there are three controlled motions for an industrial crane: a trolley translation, a crane translation and a vertical motion considering the variation of the length l. Since the first two movements independent, we can consider them equivalently.

In Figure 1, a geometric description of the industrial crane system as a double-pendulum is given. A rectangular body (representing the trolley of the Overhead traveling crane or

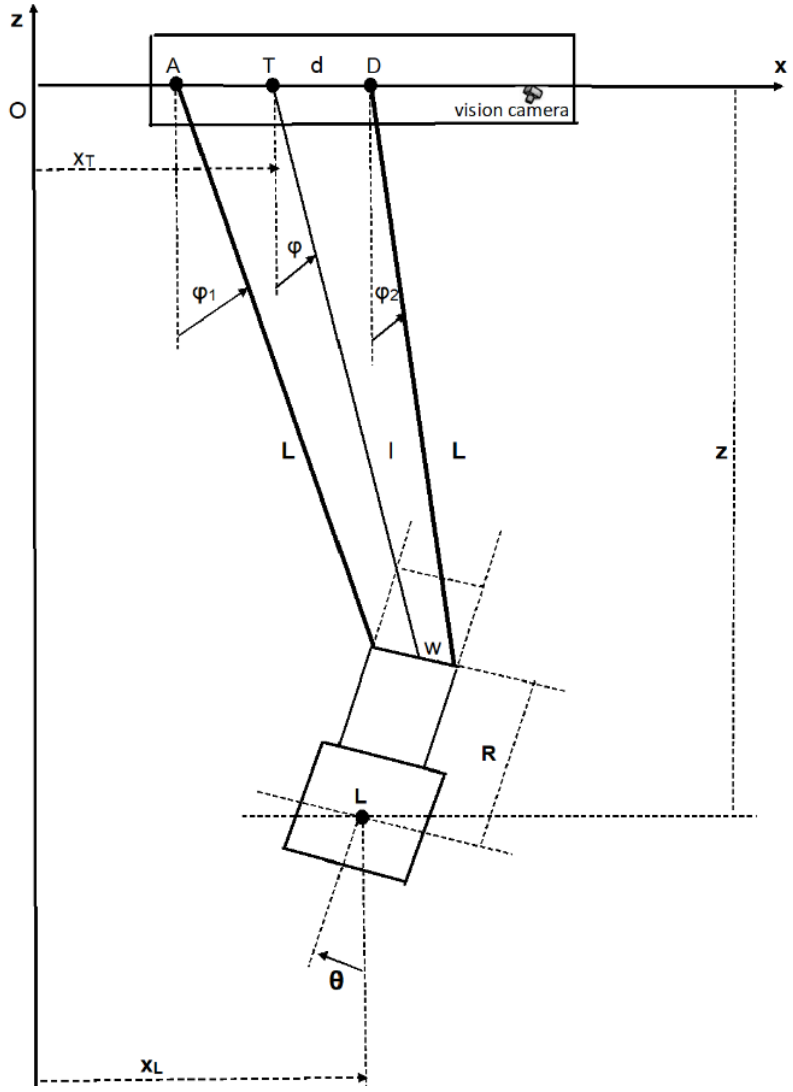


FIGURE 1. Geometric description of the industrial crane system, which can be either an Overhead traveling crane or a gantry crane. In the top is defined a camera both for the position feedback of the load and for the collision detection. The marker (a reduced 5×5 chessboard or an Aruco marker) is mounted on the top support of the payload.

the gantry crane) moves along the x axis, starting from the origin O. The points A, T and D represent, respectively, the suspension points of the lateral and central cables (of length L and l, respectively). d is the distance between the points A-T, and T-D, w is the distance between the same cables, but in correspondence of the attack point on the payload. Finally, x_T and x_L represent the horizontal position of the crane mass center T (with mass M) and the payload mass center L (with mass m), respectively. z is the height of the mass center of the payload (expressed as the distance from the x axis).

There is a correction on the cable length to consider. That is because the spreader is suspended by two inclined cables (with inclination angles φ_1 and φ_2, respectively). Therefore, there is a constraint between the real sway angle of the pendulum φ, the inclination angles φ_1 and φ_2 and the roll angle of the spreader θ, as can be seen in Figure 1. The derivation of this constraint is done in [2].

The same considerations are valid for movements of the crane along the horizontal axis y, perpendicular to x.

3.1. **Lagrange equations.** We can represent the dynamics of the generalized crane as a multi-body system with 3 independent degrees of freedom (see Figure 1):

$q_1 = \varphi$: sway angle
$q_2 = x$: horizontal position of the crane
$q_3 = l$: hoisting position of the payload.

If we establish the Lagrange function:

$$L = T + U \tag{11}$$

and applying the generalized Lagrange equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i, \tag{12}$$

we can obtain the system of equations in the generalized coordinates q_i. Focusing our interest relative to the Lagrangian coordinate φ, in the first approximation, from Lagrange equations (12) we obtain:

$$\ddot{\varphi} = \frac{1}{l}\left\{-g\sin\varphi - \ddot{x}\cos\varphi + 2kl\varphi + \left(\dot{l} - k_f\right)\dot{\varphi}\right\}. \tag{13}$$

In order to deepen the anti-sway solution in open-loop, we refer to the recent work of the author of this paper, relatively to anti-sway for Overhead and gantry cranes [2]. The solution is obtained starting from Equation (13), with an iterative method that can be represented, at any time t, in the following way:

$$\dot{\varphi}_t = \dot{\varphi}_{t-1} + \ddot{\varphi}_{t-1}\Delta t, \tag{14}$$

$$\varphi_t = \varphi_{t-1} + \dot{\varphi}_{t-1}\Delta t, \tag{15}$$

$$\ddot{x} = \left(\dot{x}_t - \dot{x}_{t-1}\right)\Delta t, \tag{16}$$

$$\dot{l} = \left(l_t - l_{t-1}\right)\Delta t, \tag{17}$$

$$\ddot{\varphi}_t = \frac{1}{l_t}\left\{-g\sin\varphi_t - \ddot{x}_t\cos\varphi_t + 2kl\varphi_t + \left(\dot{l}_t - k_f\right)\dot{\varphi}_t\right\}, \tag{18}$$

where φ_t and φ_{t-1} represent the sway angle along the x-direction respectively at the time t and at the previous time $t-1$, $\dot{\varphi}_t$ and $\dot{\varphi}_{t-1}$ represent the velocity of the sway angle along the x-direction respectively at the time t and at the previous time $t-1$, $\ddot{\varphi}_t$ and $\ddot{\varphi}_{t-1}$ represent the acceleration of the sway angle along the x-direction respectively at the time t and at the previous time $t-1$, $\ddot{x}_t$ represents the acceleration of the movement of the trolley relative to the x-direction at the time t, $\dot{x}_t$ and $\dot{x}_{t-1}$ represent the velocity

of the movement of the trolley relative to the x-direction respectively at the time t and at the previous time $t-1$, $\dot{l}_t$ represents the velocity of the movement of the hoisting relative to the z vertical direction at the time t, l_t and l_{t-1} represent the length of the cable respectively at the time t and at the previous time $t-1$, and Δt represents the time difference between the instant t and the instant $t-1$. The x-axis can represent, in an equivalent way, either the Trolley axis or the Translation axis.

3.2. **Closed-loop solution.** In order to consider the use of a closed-loop solution for the anti-sway of the payload, the effect of the feedback arriving by the camera vision has to be integrated into the solution given by Equation (18). We apply the linear control technique for crane systems defined by the Proportional Integral Derivative (PID) control. The PID controller gains can be tuned considering the different cable lengths. Specifically, we obtain the correction $\Delta\dot{x}$ of the velocity of the movement along the axis x according to the following equation:

$$\Delta\dot{x} = K_0\varphi_x + K_1\dot{\varphi}_x + k_p\left(\varphi_x - \tilde{\varphi}_x\right) + k_d\left(\dot{\varphi}_x - \tilde{\dot{\varphi}}_x\right) + k_i\int\left(\varphi_x - \tilde{\varphi}_x\right)dt. \tag{19}$$

K_0 and K_1 are the observer gains applied respectively to the sway angle and to the velocity of the sway angle along the axis x. k_p, k_d, k_i are, respectively, the proportional, derivative and integral PID gains agents on the measured error. This is defined by the difference between the calculated φ_x sway angle (or his derivative $\dot{\varphi}_t$) and the measurement $\tilde{\varphi}_t$ (or his derivative $\tilde{\dot{\varphi}}_t$) obtained by the camera vision using the adaptive particle filter. $\Delta\dot{x}$ is the correction signal that is added to the velocity set-point $\dot{x}_{set}$ along the axis x. Therefore, the reference of the velocity applied as an input to the inverter driving the motor relative to the axis x is the velocity set $\dot{x}_{ref}$ (along with the axis x), that is obtained in the following way:

$$\dot{x}_{ref} = \dot{x}_{set} + \Delta\dot{x}. \tag{20}$$

3.3. **Collision detection.** With the goal to detect a collision of the payload with possible sudden obstacles, we define two dynamical models. A nominal model Γ_1 describes the normal behavior of the swinging payload when there are no collisions in the crane movement. An alternative model Γ_2 defines the behavior of the swinging payload when a collision happens. A fundamental Markov model is formulated using both dynamical models, where a variable Ψ_t determines which of the models Γ_1 and Γ_2 that currently is determining the state dynamics. The state vector of the system is X_t, which consists of the same elements in both dynamical models. In order to identify a collision, we define an additional fault parameter ρ_t that evolves following a Markov model and determines the current system dynamical model. The fault parameter ρ_t is a random variable with values that we establish as: $P = \{0, 1\}$ and transition probability:

$$\pi_{i,j} = P(\rho_{t+1} = j | \rho_t = i). \tag{21}$$

Correspondingly, our sequence state vector X_t for the dynamic system is:

$$X_t = \begin{bmatrix} \varphi_x \\ \dot{\varphi}_x \end{bmatrix}. \tag{22}$$

The system is discretized in time as described from Equation (14) to (18), with the aim to obtain the state vector X_t for each state t. With the purpose to predict the probability distribution of the crane position after a movement, it is necessary to consider the effect of the noise on the resulting position, due to the possible external influences. Therefore,

we consider an additive noise vector $\vec{v_t}$ that is added in the X_t components of Equation (14) and of Equation (15):

$$\vec{v_t} = \begin{bmatrix} v_t \\ \dot{v}_t \end{bmatrix}, \tag{23}$$

where the components are samples obtained from the Gaussian distribution:

$$v_t \sim N(0, \sigma^2). \tag{24}$$

The same considerations must be made on the component along the axis y perpendicular to the axis x in the horizontal plane, that is in the second direction of the payload sway. Therefore, we can state that the state corresponding to the dynamical model without collision (nominal model) is described in the following way:

$$\Gamma_1 : x_t = \Psi(x_{t-1} + \alpha_1 v_t), \tag{25}$$

while the opposite model that describes a collision is given by:

$$\Gamma_2 : x_t = \Psi(x_{t-1} + \alpha_2 v_t). \tag{26}$$

We assume that the measurement vector z_t is affected by additive Gaussian noise so that the observation density $p(z_t|x_t)$ takes a Gaussian distribution.

α_1 and α_2 are 2 parameters such that $\alpha_1 \ll \alpha_2$. That is because, after a collision, the validity of the mathematical model describing the payload movement falls and so the payload position is defined by the second term in the (26) containing the Gaussian noise. On the contrary, in the states without collision, the crane movement is well described by the mathematical model, and so the Gaussian noise term in the (25) has to be less relevant.

4. **Camera Calibration.**

4.1. **Intrinsic and extrinsic camera parameters.**

With the aim to obtain the feedback of the crane load position, a sequence of the camera images is used. Therefore, we must consider the image geometry.

Practically, our goal is to define the relationship between the position of the points in an inertial frame of reference and the corresponding position in the image frame of reference. Consider a plan π, the image plane, and a point O in the space, called center or projection focus. The distance between π and O is the focal length. The line through O and perpendicular to π is the optical axis, while o, the intersection between π and the optical axis, is called the main point or image center.

As seen in Figure 2, point p is the image of P, and it is the point that lies in the line between P and O intersecting the image plane π. By focusing on the reference system along the 3 coordinate axes where the origin is O and the π plane orthogonal to the z-axis, we consider the points $P = [X, Y, Z]^T$ and $p = [x, y, z]^T$. This reference system is the reference system of the camera and is of fundamental importance in the computer vision. The basic equations of projection in the crane's reference system are:

$$x = f\frac{X}{Z}; \quad y = f\frac{Y}{Z}. \tag{27}$$

In the frame of reference of the camera, the third component of the image points is always equal to the focal length; for this, it is omitted and written that $p = [x, y]^T$. The camera images are two-dimensional arrays of pixel intensities $I(x; y)$ where x and y are pixel coordinates in the image plane. Either that reconstruct the 3D structure of a scene or that process the position of objects in space, computer vision algorithms need equations that link the coordinates of the points in the 3D space with the coordinates

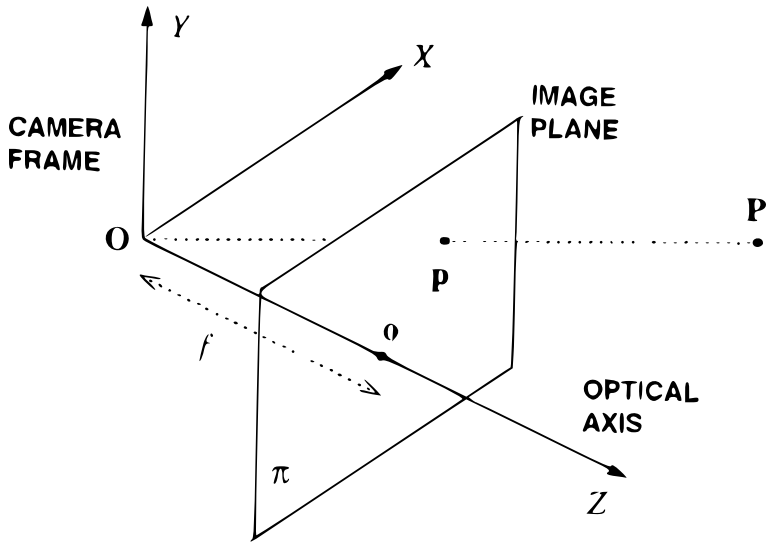


FIGURE 2. The geometry of the camera vision

of the corresponding image points. These equations are written in the crane frame of reference, but it is assumed that:

- The crane frame of reference can be located in relation to a frame of reference, known as a world reference frame (inertial frame of reference);
- The coordinates of the image points in the crane frame of reference can be obtained from the coordinates of the pixels, the only ones that can be obtained directly from the image.

That is equivalent to assuming the knowledge of some characteristics of the crane, known as intrinsic and extrinsic parameters of the crane. The extrinsic parameters are those parameters that define the position and orientation of the crane frame of reference with respect to the frame of reference in the known space. The intrinsic parameters are the parameters necessary to connect the pixel coordinates of an image point with the corresponding coordinates in the crane frame of reference. The problem of evaluating the value of these parameters is called calibration.

The extrinsic parameters of the crane, suitable for describing the transformation between the reference system united to the crane and the inertial frame of reference in the space, are:

- a 3D translation vector, T, which describes the relative positions of the origins of the two frames of reference;
- a rotation matrix 3×3, R, an orthogonal matrix that carries the corresponding axes of the two frames on each other.

Orthogonality relationships reduce the number of degrees of freedom of R to 3. The intrinsic parameters can be defined as the set of parameters necessary to characterize the optical, geometric and digital characteristics of the "vision" of the crane. They are:

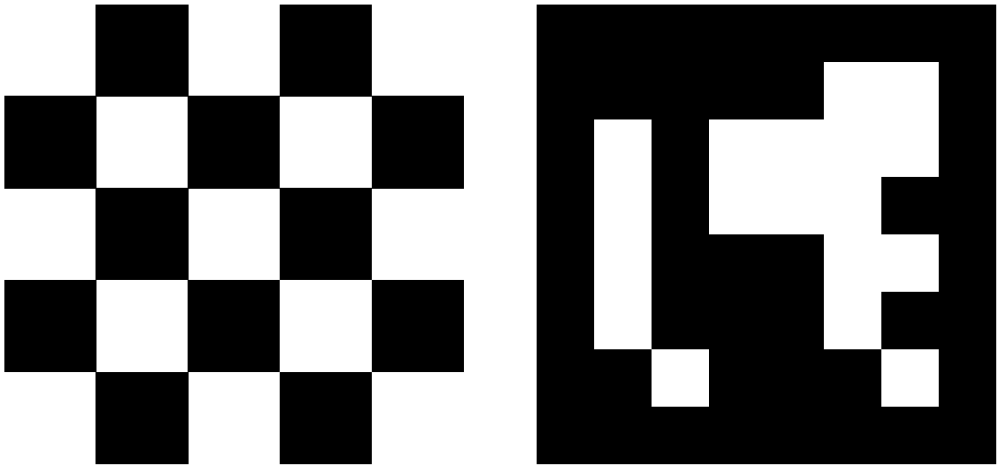
- the perspective projection, for which the only parameter is the focal length, f;
- the transformation between the coordinates of the crane frame of reference and the pixel coordinates;
- the geometric distortion introduced by the optics.

By neglecting any geometric distortion that has been introduced by the optics and assuming that the pixel matrix consists of a rectangular grid of photosensitive elements,

we have:

$$x = -(x_{im} - o_x)s_x; \quad y = -(y_{im} - o_y)s_y, \tag{28}$$

where (x_{im}, y_{im}) are the coordinates of an image point in pixel units and (x, y) are the coordinates of the same point in the crane reference system. The change of sign is because the horizontal and vertical axes of the image and the crane reference have opposite orientation. We indicate with (o_x, o_y) the coordinates in pixels of the image center and (s_x, s_y) the effective size of the pixel (in millimeters), respectively in the horizontal and vertical direction. The key idea behind the calibration is to write the projection equations that link the known coordinates of a set of some 3D points and their relative projections, and solve them for the crane parameters. With the goal to know the coordinates of some 3D point, the calibration method of the crane is linked to one or more images of a calibration model, i.e., a geometrically known 3D object, placed in the space in a known position and which generates image characteristics which can be located accurately. Figure 3(a) shows a model for calibrating the crane, a portion of a chessboard (5×5 boxes). It is also used as a detector of the payload position, placing it on the support of the payload. It is easy to know the 3D position of the vertices of each square, black and white, once the position of the two planes has been measured, and locate the vertices of the image as the intersection of the image lines, thanks to the sharp contrast and the simple geometry of the model. An alternative to the utilization of the chessboard, an Aruco marker (Garrido-Jurado et al. [15]) can be used to locate the crane load in the camera images (Figure 3(b)). Using functions from the OpenCV library (Bradski and Kaehler [16]) the pose of the Aruco marker can be found.



(a) A portion of a chessboard (5×5 boxes) is used as a model for calibrating the camera vision. It is used also as a detector of the payload position, placing it on the top support of the payload.

(b) Aruco marker used as an alternative to the chessboard for the detection of the payload position

FIGURE 3. Images of the calibration model

4.2. **Calculation of the camera parameters.** Consider the 3D point, P, defined by its coordinates $[X_w, Y_w, Z_w]^T$ in the inertial reference system. Let $[X_c, Y_c, Z_c]^T$ be the coordinates of P in the crane reference system (with $Z_c > 0$ if P is visible). The position and orientation of the crane reference system are not known, i.e., are not known the 3×3 rotation R matrix and the 3D T translation vector such that:

$$\begin{bmatrix} X^C \\ Y^C \\ Z^C \end{bmatrix} = R \begin{bmatrix} X^W \\ Y^W \\ Z^W \end{bmatrix} + T. \tag{29}$$

Assuming that the radial distortion can be neglected, we can write the image of $\left[X_c, Y_c, Z_c\right]^T$ in the reference system of the image as:

$$x_{im} = -\frac{f}{s_x}\frac{X^C}{Z_C} + o_x, \tag{30}$$

$$y_{im} = -\frac{f}{s_y}\frac{Y^C}{Z_C} + o_y. \tag{31}$$

For simplicity, we can omit the "im" subscript indicating the coordinates of the pixels of the image and write (x, y) instead of (x_{im}, y_{im}). These coordinates depend on 5 intrinsic parameters: f, s_x, s_y, o_x, o_y, and the 5 parameters are not independent. However, if we assume that $f_x = \frac{f}{s_x}$ and $\alpha = \frac{s_y}{s_x}$, we can consider a new set of four intrinsic parameters f_x, α, o_x, o_y all independent of each other. The parameter α, generally called aspect ratio, indicates the mathematical relationship between the width and height of a pixel image. From Equations (29), (30), (31), considering the 9 components of the matrix R $(R_{11}, R_{12}, R_{13}, R_{21}, R_{22}, R_{23}, R_{31}, R_{32}, R_{33})$ and the 3 components of the vector T (T_x, T_y, T_z), we obtain:

$$x - o_x = -f_x \frac{R_{11}X^W + R_{12}Y^W + R_{13}Z^W + T_x}{R_{31}X^W + R_{32}Y^W + R_{33}Z^W + T_z}, \tag{32}$$

$$y - o_y = -f_y \frac{R_{21}X^W + R_{22}Y^W + R_{23}Z^W + T_y}{R_{31}X^W + R_{32}Y^W + R_{33}Z^W + T_z}. \tag{33}$$

These equations directly connect the coordinates of the space $\left[X^W, Y^W, Z^W\right]^T$ with the coordinates (x, y) of the corresponding image point. Both vectors are measurable using the known calibration model. Thus, given a sufficient number of points on the calibration model, we can try to solve Equations (32) and (33) for the unknown parameters. Let us assume that the center of the image is the origin of its reference system and that its coordinates are known. Thus, we can consider the translated coordinates $(x, y) = (x - o_x, y - o_y)$. If we assume that the radial distortion is negligible, we must evaluate f_x, α, R and T from the image points (x_i, y_i) where $i = 1, \ldots, N$, projections of N known points of space $\left[X_i^W, Y_i^W, Z_i^W\right]^T$ in the relative reference system. Since Equations (32) and (33) have the same denominator, from each corresponding pair of points $\left[X_i^W, Y_i^W, Z_i^W\right]$, (x_i, y_i) we can write an equation in the form:

$$x_i f_y \left(R_{21}X^W + R_{22}Y^W + R_{23}Z^W + T_y\right) = y_i f_x \left(R_{11}X^W + R_{12}Y^W + R_{13}Z^W + T_x\right). \tag{34}$$

Being $\alpha = \frac{s_y}{s_x}$, we have a system with 8 unknown variables corresponding to a vector $\vec{V}$ $(V_1 = \alpha R_{11}, V_2 = \alpha R_{12}, V_3 = \alpha R_{13}, V_4 = R_{21}, V_5 = R_{22}, V_6 = R_{23}, V_7 = T_y, V_8 = \alpha T_x)$. By writing Equation (34) for N times, we obtain a linear homogeneous system of N equations, where the coefficient matrix A is $N \times 8$:

$$AV = 0. \tag{35}$$

It is possible to prove that if $N \geq 7$ and the points are not coplanar, the matrix A has rank 7 and the system has a non-trivial solution that can be determined by an SVD (Singular Value Decomposition) factorization of A. T_z, the third component of the translation vector, and f_x, the focal distance in horizontal pixel units, can be obtained

from the least squares solution thanks to a system of equations such as Equations (32) and (33), written for N points.

5. **Implementations and Results.** In the previous sections, a statistic method for defining the position of the payload and detecting possible collisions is described. Besides, the mathematical description of the crane movement is obtained. After that, in this section, the results of these computations are validated using simulations, showing some of the most relevant situations.

A fundamental characteristic of this method is that we integrated the full control device into a PLC. Not only the iterative method able to define the crane trajectories in order achieving the anti-sway effect but also, the whole calculation of the statistic particle filter including the reading of the camera's data is obtained in the PLC. The PLC tasks in which the two different processing are elaborated have different cycle times: the task where the crane trajectory is calculated has a cycle time of 30 ms. Instead, the task where the camera's data are red and elaborated using the statistic method has a cycle time of 600 ms. Nevertheless, that allows to do not have the very high retard due to the communication via bus (for example, Ethernet) between the intermediate camera's data processor (typically a supervisor PC) and the PLC where the crane speed profile is defined. In our model, the camera, connected through an Ethernet bus, sends the data directly to the PLC, which are processed, without intermediate processing. Typically, the camera delivers images at 1024768 pixels with resolution at 35 Hz.

The goal of the simulation was to observe collision detection during tracking of the moving crane load and, contemporary, to control the sway load in closed-loop by the feedback position arriving from the camera. A drive controller generates the speed profiles of the movements relative to the trolley (Trolley movement) and relative to the crane (Translation movement) and it supplies the information of the speed profile to the drive able to control the corresponding motor. The estimator module was realized inside to a PLC. With the goal to verify the theoretical results of the investigation, the whole system of governing equation was simulated in Codesys V3.5 SP7. The speed reference is the target value to which the speed must reach, defined from the crane operator or the automatic control. Usually, it is defined in Hz, as a consequence of the way in which the electric motors are defined. To the speed in Hz on the fast shaft (that is on the motor) a velocity corresponds on the slow shaft (that is on the wheels moving on the rails of the trolley), depending on the reduction gearing from the motor to the wheel. Typically, the max speed of a trolley can be from 0.4 m/s to also more 2 m/s.

Given the initial configuration of the crane, it is convenient to use as $p(x_0|0)$ a Gaussian function, centered at the point where it is positioned, and having a variance dependent on the error from which it is affected. Even if it is assumed that there are no measurement errors (really, very unlikely), it is not appropriate to generate all the initial particles at the same point in the state space. One of the conditions that allow proper functioning of the adaptive particle filter algorithm, is the maintenance of diversity.

The N_p samples (particles) initially used, considering that the camera images have 1024768 pixels, are 5000 (after that, the particles number is calculated with the method described in 2.3). The α_1 and α_2 parameters in Equations (25) and (26) are: $\alpha_1 = 0.0001$; $\alpha_2 = 0.1$.

In all the figures, the Ramp Set is the value of the time that would be necessary for the linear motor ramp to reach the speed reference. Based on the speed reference, the estimator module computes the real speed profile in order to have the anti-sway effect. The speed profile is more extended of the linear Ramp Set. The cable length has a significant influence on the speed profile because the higher is the height (and so the cable length),

the higher is the time of the speed profile. The used values, in all the figures, are: speed reference = 35 Hz, Ramp Set = 2.5 s, cable length = 15 m, cycle time of the PLC = 30 ms. Cycle time of the elaboration for the load position values from camera = 600 ms. These values are typically used in the movement control of an overhead crane.

In Figure 4, a graph relative to the Trolley movement is represented. Specifically, in Figure 4A, the velocity profile of the Trolley calculated to obtain the anti-sway effect is shown. Besides, in Figure 4B, the correction on the velocity profile due to the feedback of the load position arriving by the camera vision is well visible. An indented profile is represented in Figure 4B, corresponding to the velocity correction produced by each feedback of the load position displayed in Figure 4D. Besides, in Figure 4C, we see the

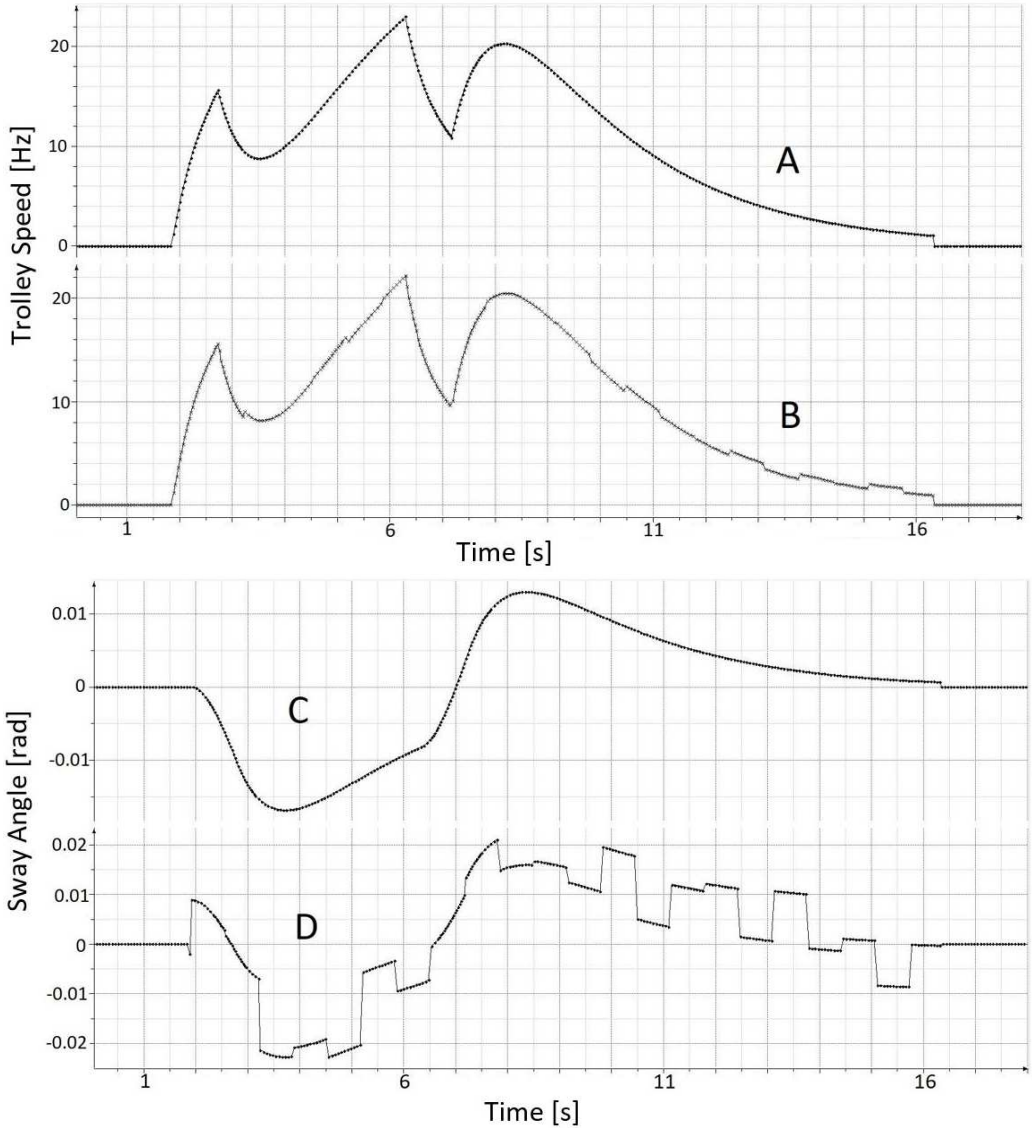


FIGURE 4. Graphs relative to the Trolley movement. Graph A: velocity profile of the Trolley without feedback of the load position. Graph B: velocity profile of the Trolley with the corrections due to the feedback of the load position with the camera vision. Graph C: profile of the ideal tangential sway angle, without feedback of the load position. Graph D: profile of the tangential sway angle with the correction due to the feedback of the load position using the camera vision.

profile of the ideal tangential sway angle, without feedback of the load position. In correspondence, in Figure 4D the profile of the tangential sway angle is represented, with the following corrections due to the feedback of the load position using the camera vision.

Instead, in Figure 5, the profile of the velocity and the corresponding profile of the sway angle are shown when the acquisition by the camera of a quite high value (0.09 rad) for the load position occurs in a specific instant (approximately at the 6.5 s). The sharp correction in the velocity profile Figure 5B corresponding to the high value of the swinging angle Figure 5D is clearly visible. It is important to remark that, in this case, because the sway angle is high but not very high, the statistic fault parameter ρ_t remains

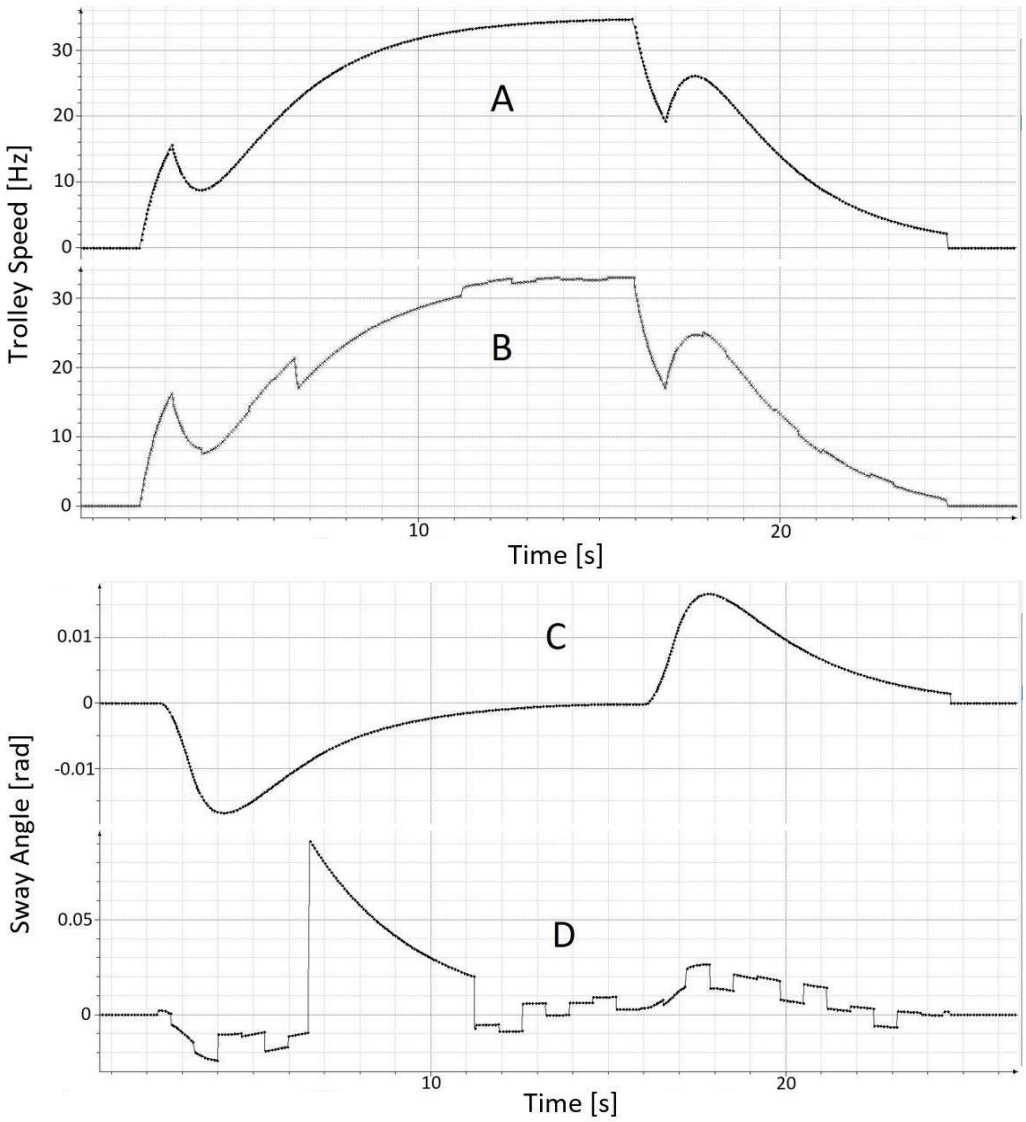


FIGURE 5. Graphs relative to the Trolley movement in correspondence with the acquisition of a quite high value (0.09 rad) for the load position by the camera. Graph A: velocity profile of the Trolley without feedback of the load position. Graph B: velocity profile of the Trolley with the corrections due to the feedback of the load position with the camera vision. Graph C: profile of the ideal tangential sway angle, without feedback of the load position. Graph D: profile of the tangential sway angle with the correction due to the feedback of the load position using the camera vision.

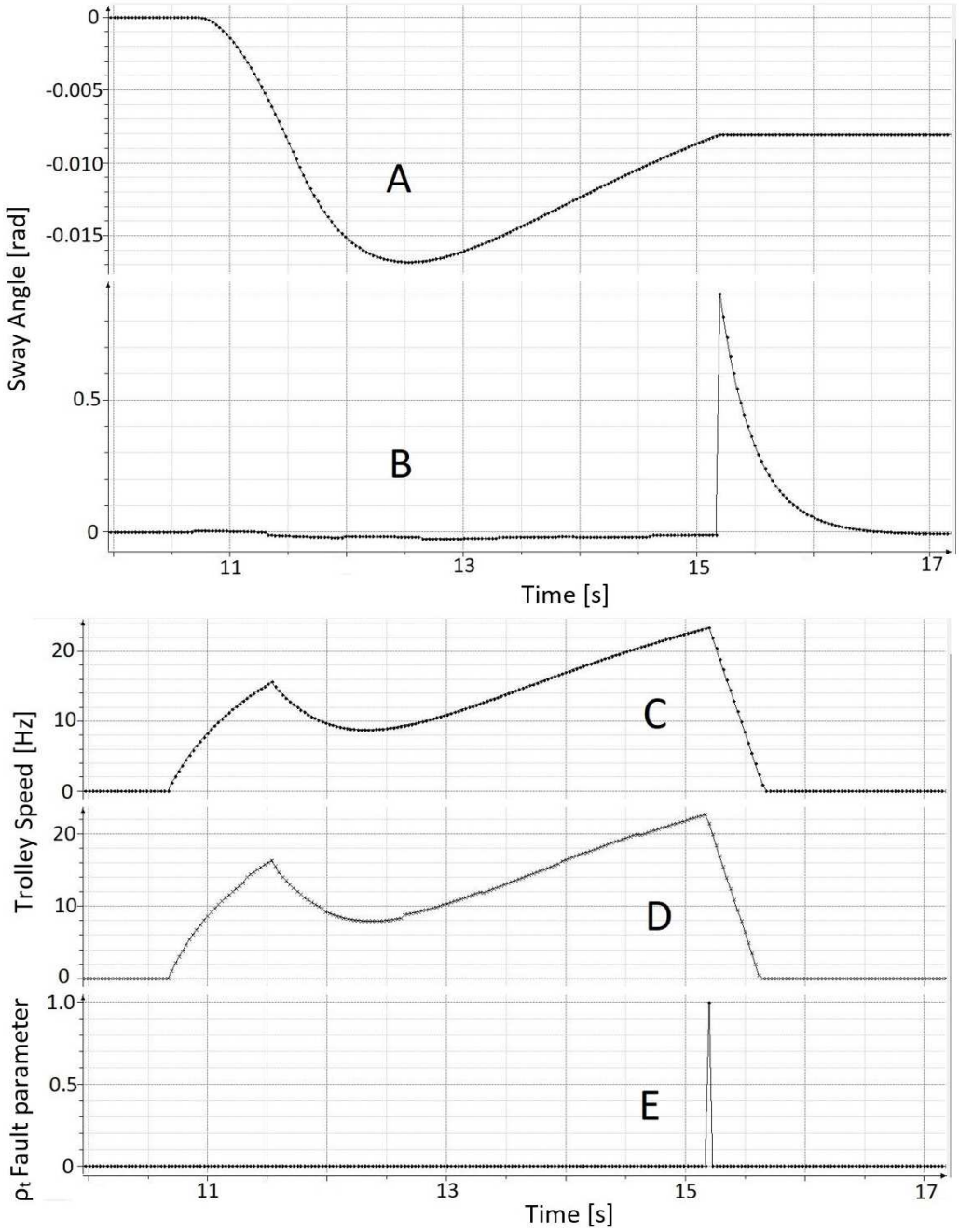


FIGURE 6. Graphs relative to the Trolley movement in correspondence with the acquisition of an exceptional very high value (0.91 rad) relative to the load position by the camera: for the statistic method, a fault occurs. Graph A: profile of the ideal tangential sway angle, without feedback of the load position. Graph B: profile of the tangential sway angle with the correction due to the feedback of the load position using the camera vision. Graph C: velocity profile of the Trolley without feedback of the load position. In the last portion of the profile, an emergency stop ramp is defined. Graph D: velocity profile of the Trolley with the corrections due to the feedback of the load position with the camera vision. In the last portion of the profile, an emergency stop ramp is defined. Graph E: fault parameter ρ_t: a random variable that assumes value = 1 when a fault occurs.

equal to 0. That means that the statistical method did not detect a collision, but only an anomalous but recoverable variation of the sway angle.

Finally, in Figure 6, the profile of the velocity, the corresponding profile of the sway angle and the fault parameter ρ_t are shown when the acquisition of a very high value (0.91 rad) for the load position by the camera occurs in a specific instant. In these graphs, one can see a usual trend of the velocity profiles and of the sway angle profiles up to the instant in which there is an extreme variation of the effective sway angle (Figure 6B), determined by the statistical method based on the camera vision. In fact, as a consequence of the statistic computation, in that instant the statistic fault parameter ρ_t takes the value equal to 1, corresponding to a collision detection. Therefore, because of this event, we can see in Figure 6C and in Figure 6D that the speed profile is stopped with an emergency ramp (high-speed ramp) with the aim to avoid damages to the crane.

6. **Conclusions.** In this paper, a novel method for controlling anti-sway in closed loop and, at the same time, for detecting possible fault and collisions of an Overhead crane load using visual tracking is presented. The method is based on an adaptive particle filter, used to have an estimation of the camera images values during the time latency of the camera images updating. In this way, the crane load position can be estimated: as a consequence, the crane load can be controlled in close-loop and detection of possible collisions of the Overhead crane load is implemented. The method uses a dynamical model of the anti-sway crane load system and an alternative dynamical model that describes the crane load when a collision occurs. A hidden Markov model allows defining the evolution of the system in the time. The calculation of the intrinsic and extrinsic camera parameters is explicated to obtain the camera calibration. Finally, the characteristics of the proposed method are implemented and tested. Future developments of this work may concern the study of the effect on the sway control of possible crane structure deformation.

Acknowledgment. The author gratefully acknowledges the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] L. Ramli, Z. Mohamed, A. Abdullahi, H. I. Jaafar and I. M. Lazim, Control strategies for crane systems: A comprehensive review, *Mechanical Systems and Signal Processing*, vol.95, pp.1-23, 2017.

[2] R. P. L. Caporali, Iterative method for controlling with a command profile the sway of a payload for gantry and overhead traveling cranes, *International Journal of Innovative Computing, Information and Control*, vol.14, no.3, pp.1095-1112, 2018.

[3] R. P. L. Caporali, Iterative method for controlling the sway of a payload on tower (slewing) cranes using a command profile approach, *International Journal of Innovative Computing, Information and Control*, vol.14, no.4, pp.1169-1187, 2018.

[4] M. A. Ahmad, M. S. Ramli, M. A. Zawawi and R. M. T. Raja Ismail, Hybrid collocated PD with non-collocated PID for sway control of a lab-scaled rotary crane, *Proc. of the 5th IEEE Conf. Ind. Electron. Appl.*, Taichung, Taiwan, pp.707-711, 2010.

[5] H. I. Jaafar, S. Y. S. Hussien and R. Gahzali, Optimal tuning of PID+PD controller by PFS for gantry crane system, *Proc. of the 10th Asian Control Conf.*, Sabah, Malaysia, pp.1-6, 2015.

[6] K. A. Heckman and W. Singhose, A feedback control system for suppressing crane oscillations with on-off motors, *International Journal of Control, Automation and Systems*, vol.5, no.3, pp.223-233, 2007.

[7] M. Hamid, M. Jamil, S. O. Gilani, S. Ikramullah, M. N. Khan, M. H. Malik and I. Ahmad, Jib system control of industrial robotic three degrees of freedom crane using a hybrid controller, *Indian Journal Sci. Technol.*, vol.9, pp.1-9, 2016.

[8] T. Inukai and Y. Yoshida, Control of a boom crane using installed stereo vision, *Proc. of the 6th International Conference on Sensing Technology (ICST)*, West Bengal, India, pp.189-194, 2012.

[9] K. C. C. Peng, W. Singhose and P. Bhaumik, Using machine vision and hand-motion control to improve crane operator performance, *IEEE Trans. Syst. Man. Cybern.*, vol.42, pp.1496-1503, 2012.
[10] P. Hyla, Single camera-based crane sway angle measurement method, *The 19th International Conference on Methods and Models in Automation and Robotics (MMAR)*, vol.978, no.1, pp.736-741, 2014.
[11] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*, Springer Science Business Media, 2006.
[12] B. Zhao and R. Skjetne, A unified framework for fault detection and diagnosis using particle filter, *Modeling, Identification and Control*, vol.35, no.4, pp.303-315, 2014.
[13] B. Zhao, R. Skjetne, M. Blanke and F. Dukan, Particle filter for fault diagnosis and robust navigation of underwater robot, *IEEE Trans. Control Systems Technology*, vol.22, no.6, pp.2399-2407, 2014.
[14] T. A. Myhre and O. Egeland, Collision detection for visual tracking of crane loads using a particle filter, *Proc. of the 42nd IEEE Conf. Ind. Electron. Soc.*, Florence, Italy, pp.865-870, 2016.
[15] S. Garrido-Jurado, R. Munoz-Salinas, F. J. Madrid-Cuevas and M. J. Marin-Jimenez, Automatic generation and detection of highly reliable fiducial markers under occlusion, *Pattern Recognition*, vol.47, no.6, pp.2280-2292, 2014.
[16] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, Incorporated, 2008.
[17] G. Cho, K. Baik and H. Kim, Study on advanced behavior analysis for towfish using particle-based simulation and real-time rendering, *ICIC Express Letters*, vol.12, no.3, pp.263-268, 2018.
[18] P. J. Van Leeuwen, H. R. Kunsch, L. Nerger, R. Potthast and S. Reich, Particle filters for applications in geosciences, *Quart. Jour. of the Meteorol. Soc.*, pp.1-32, 2017.
[19] C. Xue, G. Nie, H. Li and J. Wang, Data assimilation with an improved particle filter and its application in the TRIGRS landslide model, *Nat. Hazards Earth Syst. Sci.*, no.18, pp.2801-2807, 2018.