Visual Tracking Enhancement of Object on Circular Path based on Tuned Kalman Filter by Particle Swarm Optimization

Saad Zaghlul Saeed, PhD
Mechatronics Engineering
Dept. College of Engineering
University of Mosul,
Mosul, Iraq

Muhamad Azhar Abdilatef
Mechatronics Engineering
Dept. College of Engineering
University of Mosul,
Mosul, Iraq

Zead Mohammed Yosif
Mechatronics Engineering
Dept. College of Engineering
University of Mosul,
Mosul, Iraq

ABSTRACT

Digital image presents information in two-dimensional data, which can be used as feedback measurement for robot visual servoing control. Median filter and morphological operation are used for object detection and extraction its features. Kalman filter is applied for visual measurements that contain noises and uncertainties captured by video camera over the time. Sinusoidal Kalman filter and sinusoidal measurement model is used. The derivations of noise's process and matrices' control are presented. The Kalman filter is tuned by using PSO optimization to produce values closer to the true spatial measurements of the target. A developed PSO is proposed in which adaptive inertia weight chaotic PSO algorithm and velocity constriction factor are used for the porpuse of getting rid from the local and the adjacent optimum convergence. Simulation for tracking object on circular path are presented. Experimental result shows good performance of the proposed method for noisy measurement of the target.

Keywords

Circular path, Kalman filter, Particle swarm optimization, Robot manipulator, State space representation, Visual servoing.

1. INTRODUCTION

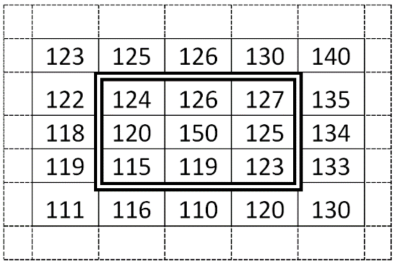
Images are a way of recording and presenting information in a visual form which are two-dimensional data. They must be converted into numerical data, referred to as digital images, to enable computer manipulation [1]. The tracking operation of objects on video sequence has many applications of real world like the robots' service, systems of surveillance, systems of public security, and processing of visual information. The tracking' target is the moving object throughout the locating process of a sequence of sequential frames. The objective of visual tracking target, is considered to be as "visual servoing", which is used for estimating the velocity and the position of an object that is spoted with the video that is recorded by the camera. The robot visual servoing refers to control of robotic systems using information from a vision system [2]. For logical term, the robot manipulator behaves as hand, and the camera as its eye. Kalman filters is now considered as a standard approximation concept for error reducing in a least squares sense and in measurements usage by different sources [3]. According to many applications, essentially the Kalman filter is a part of the development vision system in robots. The purpose of using kalman filter is to use the visual measurements those involved unwanted parts (noises) and the captured parts those are uncertainties by the video camera over the time. Using this filter can produce the values those that tend to be adjacent to the target (a self-localization robot and estimation of object) and the spatial true measurements of the robots and targets (robot self-localization and object estimation). The used Kalman filter assumes that the target dynamics could be modeled in the time that dynamics target is affected by the noise while the data of sensor is stationary with zero mean [4]. Tuning of this filter is referred to the estimation of matrix covariance. If it is not properly tuned, it leads to divergence of expected value from actual value [5]. Genetic algorithm and methods of swarm intelligence have been adopted to solve tuning of Kalman filter widely [6-8]. Unlike the genetic algorithm, the particle swarm optimization (PSO) does not have an evolution operators like crossover mutation. The advantages of using method of optimization like PSO that it does not rely explicitly on the problem's gradient to be optimized, and can process tough functions of cost with many local minima [9].

2. TARGET DETECTION

This part is very important due to the use of the obtained information in a tracking algorithm to estimate the position of the target (object). The first step is by taking a snapshot using camera to get a digital image. The digital image is consisting of pixels. The picture is in RGB (Red, Green, and Blue) form. The object detection process is required to deal with gray scale image type. In gray scale image, the image has two dimensional function, $f(x_{\text{pixel}}\ ,y_{\text{pixel}})$, where x_{pixel} and y_{pixel} are the coordinates of the image. The value of this image is represented by the amplitude of $(x_{\text{pixel}}\ ,_{\text{pixel}})$. In [10] the formula that used to convert from RGB to grayscale.

$$Gray_{image} = 0.299*R + \ 0.587*G + 0.114*B \qquad (1)$$

Many steps are applied to extract the target from the background, after converting the image to a gray scale. A median filter is applied to the image where the intensity value is assigned to the central pixel in the filter image. Median filter has many advantages of used, it doesn't reduce the brightness difference across steps because the available value is only those present in the neighborhood region, and it doesn't shift the boundaries. Applying the median filter composed three main steps: choosing suitable windows, sorting all values of pixels that surrounds neighborhood, and replacing the value of that falls at the center pixels array of the image window by the ranking result [11]. Figure 1 explains how the median filter algorithm is applied. A (3x3) neighborhood has (124, 126, 127, 120, 150, 125, 115, 119, and 123) values. These values are sorted as the following orders: 115, 119, 120, 123, 124, 125, 126, 127, and 150. The median value is 124. The last step is replacing the central pixel that equal to 150 by the median value (124).



123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Fig. 1: Application of median filter to gray scale image

A powerful operation that called morphological operation is used for the cause of image feature extraction. It is used in many real time applications like video surveillance and tracking. This operation is wildly applied to extract image component for the description and the representation of the region of the shape, it is also used to remove the unwanted regions or objects that are presented in a frame. Erosion and dilation are primitive operations that widely used. Erosion leads to reducing the size of image feature, while the dilation is the inverse of erosion that rebuilds image feature [12]. When the Erosion is followed by dilation, obtained operation is called opening, while when the dilation is followed by the erosion, the obtained operation is called closing. Image D is eroded by structuring element (E),$D \ominus E$, according to equation (2) as:

$$D \ominus E\ (l,m) = \min\{D(l+s,m+t) - E(s,t)\} \qquad (2)$$

Where (l, m) and (s,t) are the dimensions of the image, and structuring element; respectively. Image D is opened by the equation below as:

$$D \text{ o } E = (D \ominus E) \oplus E \qquad (3)$$

Erosion and opening morphological operations are used in this paper to track the moving object. After applied these operations, the information of the object (like: center, orientation, and area) are obtained in which can be used for the prediction stage. This algorithm marks its own importance as it is efficient, accurate, and less time duration.

3. KALMAN FILTER

In the year of 1960, the famous paper that named ‘’ describing a recursive solution to the discrete-data linear filtering problem’’ has been published by R. E. Kalman. From the year mentioned above, and during computing and the digitalizing revolution, Kalman filter has been the most important issue in researches and applications of tracking fields, essentially in the navigation and the autonomous tracks. According to many used aspects, kalman filter is very powerful: it can easily supports the estimations of the past dates, the present, and also the states of future, it can really do that until when the nature that is precise for the system that is modelled system is unknown [13]. Kalman filter is usually estimating a process during the use of a feedback control form: some time the filter is estimating the state of process and then obtains the feedback in the noisy measurements form. The update equations’ time are usually responsible for forward projecting (in time) the current state and error covariance estimates for obtaining a priori that estimates for the steps in next time. The equations of updated measurements are very responsible for the purpose of incorporating the new measurement into the *a priori* estimation in order to obtain the *posteriori* improved estimation. The target motion model state is defined by the following equations of the vector-matrix those shown in the following [14]:

$$x_k = A\, x_{k-1} + G\, u_{k-1} + w_k \qquad (4.a)$$

$$z_k = H\, x_k + v_k \qquad (4.b)$$

where x is a vector’s column with the system states, A is the dynamics systems matrix, u is a vector that is known, which is occasionally called a control vector, w is a white-noise tackle, that is expressed as a vector also, z is a measurement vector, H is the matrix of measurement, v is a white-noise tackle, that is expressed as a vector also. The updated time equations which are in the form of discrete are as followed:

$$\hat{x}_k^- = \Phi_k\ \hat{x}_{k-1} + G_k\ u_{k-1} \qquad (5.a)$$

$$M_k^- = \Phi_k\ M_{k-1}\Phi_k^T + Q_k \qquad (5.b)$$

The fundamental matrix here is represented by Φk, M_k is representing the covariance matrix errors in the estimated state, while Q_k is the matrix of tackle noise. The updated measurement equations are:

$$K_k = M_k^-\ H^T \left(H\, M_k^-\ H^T + R_k\right)^{-1} \qquad (6.a)$$

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H\ \hat{x}_k^-\right) \qquad (6.b)$$

$$M_k = \left(I - K_k H\right) M_k^- \qquad (6.c)$$

Where K_k represents the Kalman gain matrix, R_k is the measurement noise matrix, I is the identity matrix.

3.1 Sinusoidal Kalman Filter and Sinusoidal Measurement

The filter performance can be improved if a priori information is available. Recall that the actual signal is

$$x = \sin(\omega t) \qquad (7)$$

The preceding equation can be written as a matrix of second order differential equation in the form of state-space as:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \qquad (8)$$

From the preceding equation, the system dynamics matrix is

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \qquad (9)$$

For a time-invariant system dynamics matrix, the fundamental matrix can be derived according to

$$\Phi(t)\ = \mathcal{L}^{-1}\,[(sI - A)^{-1}] \qquad (10)$$

Using the inverse Laplace transformation to express the fundamental matrix as a function of time as:

$$\Phi(t) = \begin{bmatrix} \cos\omega t & \dfrac{\sin\omega t}{\omega} \\ -\omega\sin\omega t & \cos\omega t \end{bmatrix} \qquad (11)$$

Substituting the sampling time T_s for time, the discrete fundamental matrix is:

$$\Phi_k = \begin{bmatrix} \cos\omega T_s & \dfrac{\sin\omega T_s}{\omega} \\ -\omega\sin\omega T_s & \cos\omega T_s \end{bmatrix} \tag{12}$$

It is clear that the fundamental matrix is sinusoidal. It is very important to note that the state can be propagated forward exactly with the matrix of fundamental whenever the frequency of the sinusoid is known. The amplitude of the sinusoid does not required to be known.

3.2 Process Noise

In the previous section, Kalman filter is assumed the frequency of the sinusoid knew as constant. Thus the state equation is modified by the addition of process noise:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ u_s \end{bmatrix} \tag{13}$$

Where u_s is a source of white tackle noise that is certainly added to the second derivative. These sources of a white noise are sometime required to get the filter work later if the problem is encountered [14]. The tackle noise is already added to the derivative of the state that is the least certain, that is because of it is unknown if the sinusoid frequency is constant or not. From the preceding space-state equation, the matrix' noise continuous tackle can be written by inspection as:

$$Q = \Phi_s \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{14}$$

The discrete process noise matrix, Q_k, can be found from the continuous process-noise matrix and the fundamental matrix according to [14]:

$$Q_k = \int_0^{T_s} \Phi(\tau)\, Q\, \Phi^T(\tau)\, \mathrm{d}t \tag{15}$$

Substituting equations (11) and (14) into equation (15) to obtain:

$$Q_k = \Phi_s \begin{bmatrix} \dfrac{1}{2\,\omega^3}\left(T_s\,\omega - (\sin 2\,T_s\,\omega)/2\right) & \dfrac{1}{2\,\omega^2}\left(\sin^2 T_s\,\omega\right) \\ \dfrac{1}{2\,\omega^2}\left(\sin^2 T_s\,\omega\right) & \dfrac{1}{2\,\omega}\left(T_s\,\omega + (\sin 2\,T_s\,\omega)/2\right) \end{bmatrix} \quad 16)$$

3.3 Control Matrix

The state equation for the system is accepted as second order differential equation. For sinusoidal function, higher order derivatives are continuous and existed. Thus in this work, control matrix is also included and its magnitude will be tuned so as to improve the Kalman filter performance. If $u_{k\text{-}1}$ is assumed to be constant between sampling instants, G_k is obtained from [14] as:

$$G_k = \int_0^{T_s} \Phi(\tau)\, G\, \mathrm{d}\tau \tag{17}$$

From the equation of state-space, it can be seen that

$$G = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{18}$$

And that

$$u_{k-1} = g \tag{19}$$

Here the control action is added to the second derivative to act as constant acceleration which results in improvements for the first derivative of the signal. Substituting equations (11) and (18) into equation (17) to obtain:

$$G_k = \begin{bmatrix} (1-\cos(\omega\,T_s))/\omega^2 \\ (\sin(\omega\,T_s))/\omega \end{bmatrix} \tag{20}$$

The Kalman-filtering equation including the control matrix is:

$$\hat{x}_k = \Phi_k\,\hat{x}_{k-1} + G_k\,u_{k-1} + K_k\left(z_k - H\,\Phi_k\,\hat{x}_{k-1} - H\,G_k\,u_{k-1}\right) \tag{21}$$

Substituting the fundamental and control matrices yields a difference's matrix equation, which can be converted easily to several of difference scalar equations as shown below:

$$\hat{x}_k = \cos\omega T_s\,\hat{x}_{k-1} + \frac{\sin\omega T_s}{\omega}\,\hat{\dot{x}}_{k-1} + g(1-\cos(\omega\,T_s))/\omega^2 + K_{1k}\mathrm{Res}_\mathrm{k} \tag{22.a}$$

$$\hat{\dot{x}}_k = -\,\omega\sin\omega T_s\,\hat{x}_{k-1} + \cos\omega T_s\,\hat{\dot{x}}_{k-1} + g(\sin(\omega\,T_s))/\,\omega + K_{2k}\mathrm{Res}_\mathrm{k} \tag{22.b}$$

$$\mathrm{Res}_\mathrm{k} = x_k^* - \cos\omega T_s\,\hat{x}_{k-1} - \frac{\sin\omega T_s}{\omega}\,\hat{\dot{x}}_{k-1} - g(1-\cos(\omega\,T_s))/\omega^2 \tag{22.c}$$

4. PARTICLE SWARM OPTIMIZATION

From many years before, the modern or the nontraditional methods of optimization have been developed clearly. In 1995, the PSO is formulated by Edward and Kennedy. The thought tackle inside the algorithm is discovered by the animal's social behavior, like bird flocking, fish schooling, and ext. Each particle was moving about the cost surface with a velocity. Each particle studies the best own previous solution to the problem of optimization, and its best previous group. The optimal value must be founded by the process repeating. The PSO algorithm updates the velocity vector for each particle and then adds that velocity to the particle position or value [15]:

$$\mathbf{v}_{i,j}^{\text{new}} = w\,\mathbf{v}_{i,j}^{\text{old}} + \Gamma_1 \times \mathrm{r}_1 \times \left(\mathbf{p}_{i,j}^{\text{local best}} - \mathbf{p}_{i,j}^{\text{old}}\right) + \Gamma_2 \times \mathrm{r}_2 \times \left(\mathbf{p}_{i,j}^{\text{global best}} - \mathbf{p}_{i,j}^{\text{old}}\right) \tag{23.a}$$

$$\mathbf{p}_{i,j}^{\text{new}} = \mathbf{p}_{i,j}^{\text{old}} + \mathbf{v}_{i,j}^{\text{new}} \tag{23.b}$$

where **v** is particle velocity, *w* is inertia weight, **P** is particle position or variable, $\mathrm{r}_1$and r_2 are independent uniform random numbers, **P** $^{\text{local best}}$ is best local solution, **P** $^{\text{global best}}$ is best global solution, *i* is particle index, *j* is dimension of variable, Γ_1 is cognitive parameter, Γ_2 is social parameter.

The disadvantages of traditional PSO are the premature and local optimum convergence. PSO variants are discovered to increase its performance and improves the ability to solve a wide range of optimization problems [16]. Some modifications happened on the basic components of PSO such as inertia weight, velocity clamping, and velocity constriction [17-21]. Bansal et al. studied 15 relatively recent and popular inertia weight strategies and carried out over five basic benchmark optimization functions to compare different strategies of setting inertia weight [22]. As an overall outcome of the experiments carried out, chaotic inertia weight is the best strategy for better accuracy. Another modifications are presented which concern with the swarms itself such as swarm initiation, insertions new swarms, and mutation [23-

26]. These modifications can increase the search diversity. Jamous et al. suggested as future work to apply multiple modifications on more than one of the basic components of PSO and swarms [27]. An improved chaotic PSO algorithm is based on the inertia weight that is very adaptive (AIWCPSO) is proposed [28]. At the early stage of the evolution, the initial population and velocity are generated by using chaotic mapping. The iterative number, aggregation degree factor, and the improved evolution speed parameter are used as feedback to adjust the inertia weight adaptively.

In this paper, AIWCPSO algorithm is adapted and velocity constriction factor, λ, is introduced (equation (24)). This includes modifications to the basic components of PSO which are inertia weight and velocity constriction. Also, chaotic swarm initiation and swarm injection depending on the variance of the population’s fitness are included.

$$\mathbf{p}_{i,j}^{\text{new}} = \mathbf{p}_{i,j}^{\text{old}} + \lambda\, \mathbf{v}_{i,j}^{\text{new}} \qquad (24)$$

4.1 Novelty in AIWCPSO Algorithm

The basic novel steps of AIWCPSO algorithm are described as following:

Step 1 Chaotic initialization strategy is applied to generate initial solution to position and *N* initial velocities by cubic mapping (equation (25)). Then this initial solution is mapped to the search space range. The cubic mapping is described as following:

$$\begin{cases} p_{n+1} = 4\, p_n^3 - 3\, p_n \\ \quad -1 < p_0 < 1 \end{cases} \qquad (25)$$

Step 2 The inertia weight is updated by the equations below:

$$w_i = w_{\max} - \left(w_{\max} - w_{\min}\right) \frac{iter}{iter_{\max}}\, e^{-(\alpha\, esf_i - \beta\, adf)} \qquad (26)$$

$$esf_i = \frac{abs\,[F(pbest_i^k) - F(pbest_i^{k-1})]}{abs\; F(pbest_i^k) + abs\; F(pbest_i^{k-1})} \qquad (27)$$

$$adf = \frac{abs\,[\min\,(F_{best}^k,\; F_{avg}^k)]}{abs[\max\,(F_{best}^k,\, F_{avg}^k)]} \qquad (28)$$

Where $w_{\max}$ is represents the maximum value of the inertia weight, $w_{\min}$ is the minimum value of the inertia weight, *iter* is the current iteration number, $iter_{\max}$ is the maximum number of iteration, α and β choises are typically within the range [0, 1]. esf_i is the improved evolution speed parameter of particle *i* (*i* =1, 2,..., *N*), *adf* is the aggregation degree factor of swarm, F ($pbest_i^k$) is the particle *i* best fitness value at the k^{th} iteration, F_{best} represents the best fitness that is achieved by the particles, F_{avg} is the mean fitness of all particles in the swarm at the same iteration. Note: the superscript k is the current iteration value.

Step 3 Calculation of the variance (σ) for the population's fitness (equation (29)). If it is less than threshold value and the current iteration optimal fitness worse than the best theoretical fitness, go to chaotic disturbance.

$$\sigma^2 = \sum_{i=1}^{N} \left(\frac{F(p_i^k) - F_{avg}^k}{\max\,[\max\,[abs[F(p_i^k) - F_{avg}^k]],1]} \right)^2 \qquad (29)$$

Where F (p_i) is fitness values of particle *i*, and *N* is number of particles.

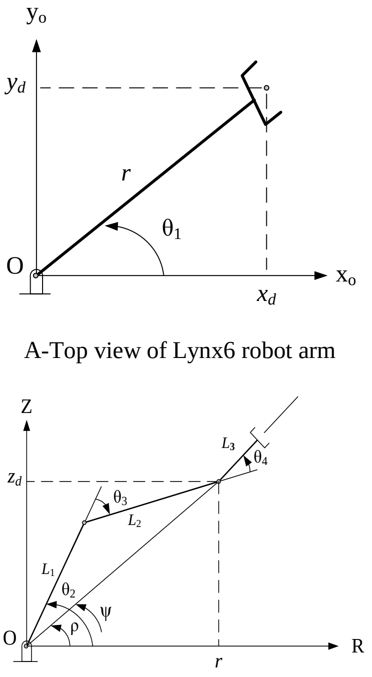
Step 4 Chaotic disturbance strategy:
Chaotic vector o_{ij} (*i* =1, 2,..., *N*; *j* =1,2,..., *J*) is generated by cubic mapping (equation (25)), where o_{0j} is (-1,1) of random numbers, and the component of this vector is loaded to the chaotic disturbance range of [$-\gamma_j$, γ_j] (*j*=1, 2,..., *J*). Then chaotic disturbance variation is $\Delta p_i = (\gamma_1\, o_{i1}, \gamma_2\, o_{i2},\ldots,\gamma_J\, o_{iJ})$. The position updated of particle after adding the chaotic disturbance variation is given by: $pb_{ij}\,(k+1) = p_{ij}\,(k) + v_{ij}\,(k) + \Delta p_{ij}$. Finally, comparison is made between the fitness values of F (pb_i (k +1)) and F (p_i (k +1)). If F (pb_i (k +1)) is better than F (p_i (k +1)), then p_i (k +1) is updated by pb_i (k +1). Note: *J* is the variable's dimension of particle *i*. For more details about AIWCPSO algorithm see [28].

5. INVERSE KINEMATICS of ROBOT ARM

The inverse kinematics problem for a robotic manipulator involves obtaining the required manipulator joints' angel for a given desired end-point position and orientation. Thus the knowing of the links' parameters and desired coordinates (x_d, y_d, z_d) allows to calculate the joints' angle by using the inverse kinematics [29]. The top view of the robot arms on the x_o, y_o plan gives a radial distance (Fig.2a). The specified radial distance, r, from the base is related to the desired target position coordinates x_d and y_d by:

$$r = \sqrt{x_d^2 + y_d^2} \qquad (30.a)$$

$$\theta_1 = A\tan 2\,(y_d\,, x_d\,) \qquad (30.b)$$



A-Top view of Lynx6 robot arm

B-geometry associated with three links

Fig. 2: Geometric representation for lynx6 robot arm.

From Fig. 2b and considering the law of cosines, it can be used to solve for the joints' angle as:

$$\cos\theta_3 = \frac{r^2 + z_d^2 - L_1^2 - L_2^2}{2L_1L_2} \qquad (31)$$

$$\rho = \text{Atan2}(z_d, r) \tag{32}$$

Applying the law of cosines to find ψ:

$$\cos\psi = \frac{r^2 + z_d^2 + L_1^2 - L_2^2}{2L_1\sqrt{r^2 + z_d^2}} \tag{33}$$

Then it is obtained:

$$\theta_2 = \rho + \psi \tag{34}$$

Finally, it can solve for θ_4 as:

$$\theta_4 = -90 - \theta_2 + \theta_3 \tag{35}$$

6. SIMULATION RESULTS

Simulations are implemented in Matlab7. The Kalman filter parameters are initialized to unity. The tuned parameters are: process noise (*Q*), noise measurement (*R*), and control factor (*g*). These parameters are tuned by PSO to minimize the root mean square error (RMSE). The tuning is achieved off-line. Then, these tuned parameters are used for on-line Kalman tracking. The parameters of PSO are set as follows: Γ_1=2.05; Γ_2=2.05; λ=0.7298 [21]; *N*= 40; $iter_{\max}$= 30; $w_{\max}$=0.45, $w_{\min}$= 0.15; *α*=0.75; *β*=0.25; *γ*=10^{-3}; threshold value= 10^{-3}; best theoretical value =10^{-6}.

In the two-dimensional x-y plane, circular motion can be described by the equations:

$$\begin{aligned} x &= +\,50\cos(\omega\, t) \\ y &= +\,50\sin(\omega\, t) \end{aligned} \tag{36}$$

A circular path with speed constant angular (ω=2π/23 rad/sec) is created in Matlab. A white noise with amplitude equals to unity is added to the measurements. These circular paths are shown in Fig. 3. Due to an unknown filter usage environment, the tackle noise is left to intrusions of engineering. Proper initialization of this, is essential as it is necessary to minimize the error obtained during tracking [30]. Figure 4 illustrates the performance of the Kalman filter with and without process noise and control matrices. The inclusion of process noise and control matrices improves the performance of sinusoidal Kalman filter for the coordinate's estimation with its derivate (Fig. 5). Tuning the filter by PSO minimizes the error to almost zero after 10 seconds as shown in Fig. 6. Thus tuning of Kalman filter is important to obtain more accurate results for object tracking. A comparison between linear and sinusoidal Kalman filters is shown in Fig.7. It is clear that sinusoidal Kalman filter with process noise and control matrices provides more smooth results in comparison to the linear Kalman filter.

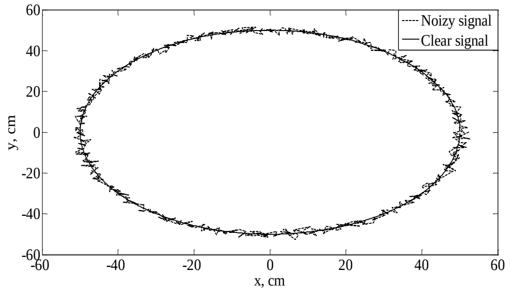
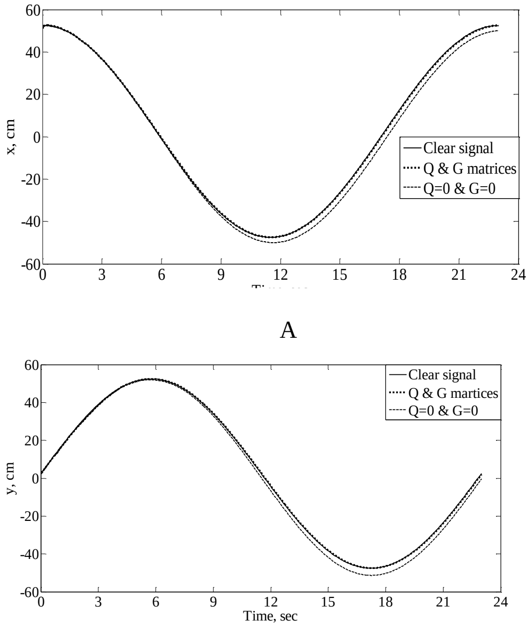


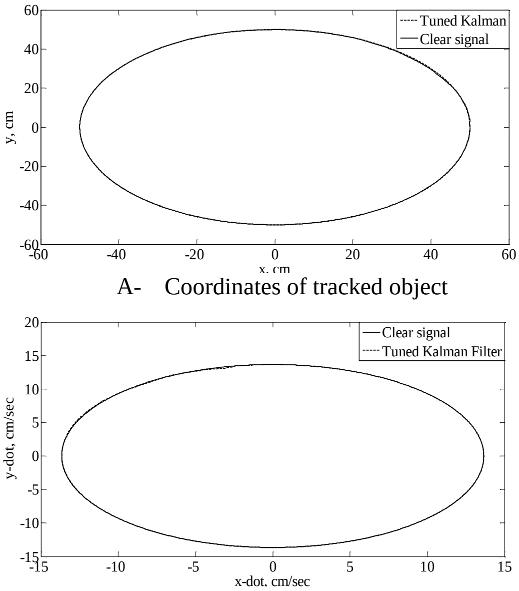
Fig. 3 Circular path with constant angular speed



A

B

Fig. 4: Performance of the Kalman filter with and without process noise and control matrices



A- Coordinates of tracked object

B- Derivative of coordinates of tracked object

C-

Fig. 5: Path obtained by sinusoidal Kalman filter with process noise and control matrices tuned by PSO

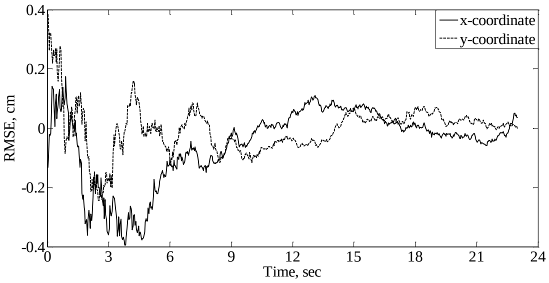


Fig. 6: Error in the coordinates after Tuning Kalman filter by PSO

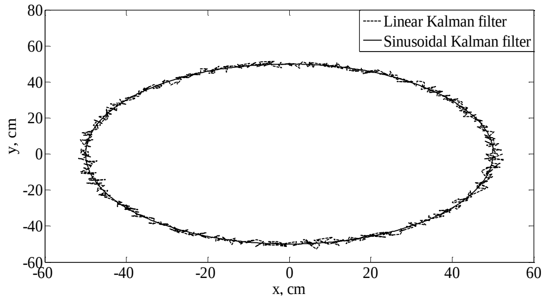


Fig. 7: Comparison between linear and sinusoidal Kalman filters

Table 1 summarizes the RMSE for the circular path in the coordinates. According to the following table, and during the use of the sinusoidal Kalman filter with the tackle noise and control the matrices, the errors of estimations are significantly reduced in both axes.

Table 1. Summary of Results for Circular Path

Results	RMSE (cm)		
	x- coordinate	y- coordinate	Radius
Noisy measurement	1.0064	0.9529	1.3859
Linear Kalman filter [30]	0.8944	0.9529	1.3069
Sinusoidal Kalman filter without *Q* and *G* matrices	2.4853	2.4933	3.5204
Sinusoidal Kalman filter with *Q* and *G* matrices	0.1342	0.0812	0.1568

During the experimental test (Fig. 8), the robot arm (Lynx6), rotary carousal, and digital camera are used. The supposed target was a dark circle, which is placed at 5 cm from the disc center of the rotary carousal. The rotary carousal rotates with constant angular speed. It consists of plastic disc, stepper motor, and electrical circuit driver. The motor's angular speed is controlled by micro-controller. It completes one revolution every 23 seconds. The focal length of the camera is obtained by calibration as (240 pixel=227 mm) for both axis. The used frame rate is 15 frames per second. Figure 9 shows the experimental results for tracking the object which is fixed on the rotary carousal. The positions of the object's tracking on rotary carousal during 5 seconds period, are presented in Fig.10. The black solid line represents an actual noisy measurement comes from using camera after image processing. The red x-mark is the center of tracked object obtained by the sinusoidal Kalman filter with process noise and control matrices.

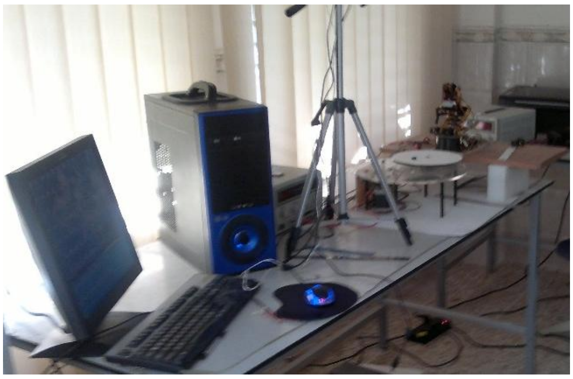
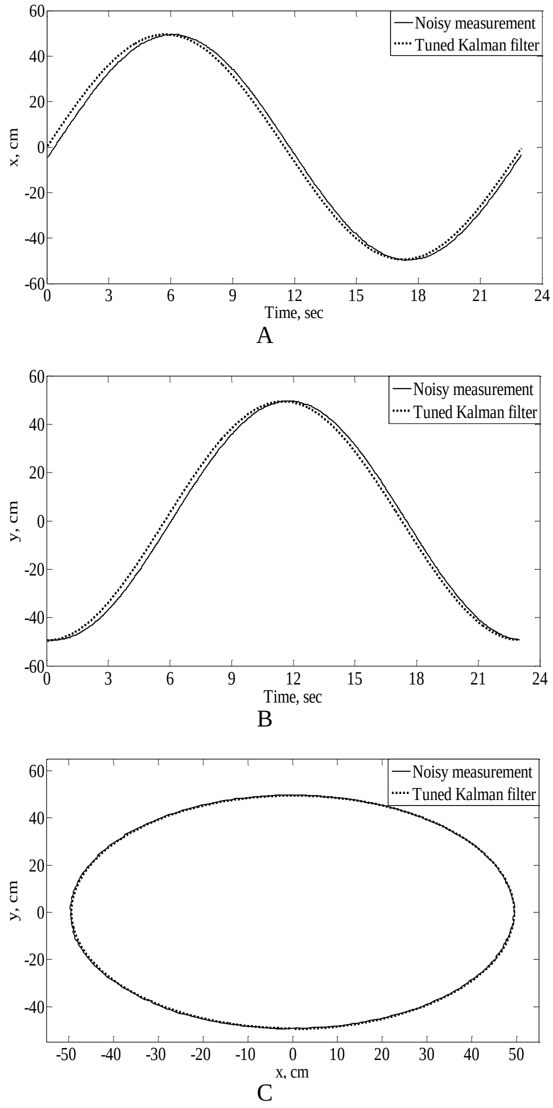


Fig. 8: Experimental Devices

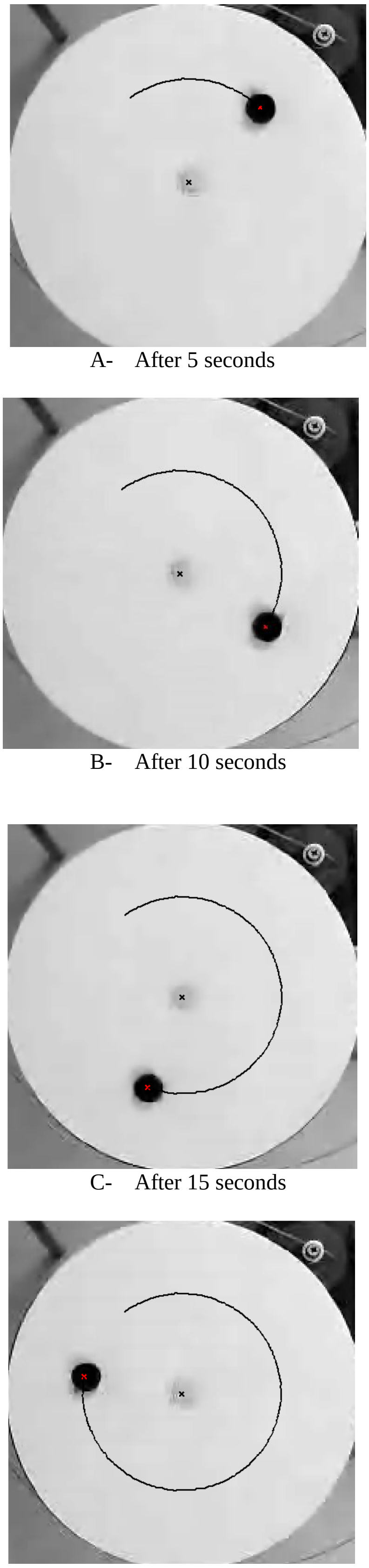


A

B

C

Fig. 9: Experimental results of tracking a target in a circular path



A- After 5 seconds

B- After 10 seconds

C- After 15 seconds

D- After 20 seconds

Fig. 10: Positions for tracking the object on rotary carousal experimentally

Table 2 summarizes the RMSE of experimental results for tracking moving object in the coordinates. The estimation error is significantly reduced in both axes

Table 2. Experimental results

Results	RMSE (cm)		
	x-coordinate	y-coordinate	Radius
Noisy measurement	2.1465	2.0587	2.9742
Tuned Sinusoidal Kalman filter	0.1713	0.2037	0.2661

7. CONCLUSION

In this paper, a real-time tracking target for circular path is presented. The proposed method includes an image processing with Kalman filter as the target estimator position. The process noise and control matrices are proved to improve the performance of Kalman filter for tracking rotating object on circular path. Experimental result shows good performance of the proposed method for noisy measurements of the target.

8. FUTURE WORK

The future work will focused on the realization of tracking more than one object, use more than one robot arm and employee this ideas in industry fields. In order to satisfy that, another algorithm using the proposed method should be in addition to tracking algorithm (Kalman filter) to achieve multiple object tracking.

9. REFERENCES

[1] Sarp Ertürk, “Digital Image Processing,” February 2003 Edition, University of Kocaeli February 2003 Edition Part Number 323604A-01 National Instruments Corporation.

[2] A. Muis and K. Ohrishi, “Eye-to-Hand Approach on Eye-in-Hand Configuration within Real-Time Visual Servoing,” IEEE/ASME Transactions on Mechatronics, Vol.10, No. 4, 2005.

[3] S. Y. Chen, “Kalman Filter for Robot Vision: A Survey,” IEEE Transactions on Industrial Electronics, Vol. 59, No. 11, November 2012.

[4] B. Torkaman and M. Farrokhi, “A Kalman–Filter Method for Real-Time Visual Tracking of a Moving Object Using Pan and Tilt Platform,” IJSER Journal, Vol. 3, No. 8, August-2012.

[5] R. K. Jatoth and Dr. T. K. Kumar, “Swarm Intelligence Based Tuning of Unscented Kalman Filter for Bearings Only Tracking,” Int. J. of Recent Trends in Engineering and Technology, Vol. 2, No. 5, Nov 2009, DOI: 01.IJRTET.02.05.335.

[6] M. A. Badamchizadeh, N. Nikdel, and M. Kouzehgar, “Optimization of data fusion method based on Kalman filter using Genetic Algorithm and Particle Swarm Optimization,” IEEE 2nd International Conference on Computer and Automation Engineering (ICCAE), Vol. 5, 2010, pp.359-363, DOI: 10.1109/ICCAE.2010.5451413

[7] R. K. Jatoth and T. K. Kumar, “Particle Swarm Optimization Based Tuning of Extended Kalman Filter for Maneuvering Target Tracking,” International Journal Of Circuits, Systems and Signal Processing, Issue 3, Volume 3, pp. 127-136, 2009.

[8] G. Lin, Z. Jing and Z. Liu, “Tuning of extended kalman filter using inproved particle swarm optimization for sensorless control of induction motor,” Journal of Computational Information Systems 10(6):2455-462 · January 2014. DOI: 10.12733/jcis9762

[9] M. E. H. Pedersen and A. J. Chipperfield, “Simplifying Particle Swarm Optimization,” Applied Soft Computing, vol.10, no.2, March 2010, pp.618-628.

[10] John C. Russ, “The image processing handbook,” CRC Press, 6th edition. April 7, 2011. pp. 885

[11] R. C. Gonzalez, “Digital Image Processing,” 3rd edition, Prentice-Hall, Inc., 2007.

[12] R. C. Marsal, “Morphological And Statistical Analysis Of Biomaterials with Applications in Tissue Engineering by Means of Microscopy Image Processing,” IEEE Latin America Transactions, vol. 9, no. 3, 2011, pp. 399-407.

[13] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” University of North Carolina at Chapel Hill Chapel Hill, July 24, 2006.

[14] P. Zarchan and H. Musoff, “Fundamentals of Kalman Filtering: A Practical Approach,” Third Edition 2009.

[15] S. S. Rao, “Engineering Optimization Theory and Practice,” Fourth Edition, John Wiley & Sons, Inc., 2009, pp. 708.

[16] S. D. Chavan and N. P. Adgokar, “An Overview on Particle Swarm Optimization: Basic Concepts and Modified Variants,” IJSR Journal, Vol. 4, No.5, 2015

[17] A. Alfi, “Particle Swarm Optimization Algorithm with Dynamic Inertia Weight for Online Parameter Identification Applied to Lorenz Chaotic System,” ICIC International journal, Vol.8, No.2, February 2012.

[18] A. Djoewahir, T. Kanya, and M. Shenglin, “A Modified Particle Swarm Optimization with Nonlinear Decreasing Inertia Weight Based PID Controller for Ultrasonic Motor,” International Journal of Innovation and Technology, Vol.3, No.3, June 2012.

[19] R. Malwiya and V. Rai, “Optimal Speed Controlling of Induction Motor Using New PSO,” Journal IJATER, Vol.5, Nol.2, March 2015.

[20] N. M.A. Ibrahim, H. E.M. Atti, H. E.M. Talaat, and A. H. K. Alaboudy, “Modified Particle Swarm Optimization Based Proportional-Derivative Power System Stabilizer", I.J. Intelligent Systems and Applications, Vol. 3, 2015, pp62-76.

[21] D. Tian, “A Review of Convergence Analysis of Particle Swarm Optimization,” International Journal of Grid and Distributed Computing", Vol.6, No.6, 2013.

[22] J.C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, “Inertia Weight Strategies in Particle Swarm Optimization,” IEEE 3rd World Congress on Nature and Biologically Inspired Computing, Nature and Biologically Inspired Computing, Salamanca 19-21 Oct. 2011, pp.633 – 640, DOI: 10.1109/NaBIC.2011.6089659.

[23] C. H. Yang, S. W. Tsai, L. Y. Chuang, and C. H. Yang, “A Modified Particle Swarm Optimization for Global Optimization,” International Journal of Advancements in Computing Technology, Vol.3, No. 7, August 2011.

[24] S. M. Elsayed, R. A. Sarker, and E. M. Montes, “Particle Swarm Optimizer for constrained optimization,” Evolutionary Computation (CEC), 2013 IEEE Congress on, Year: 2013, Pages: 2703 - 2711, DOI: 10.1109/CEC.2013.6557896 , IEEE Conference Publications.

[25] C. Ratanavilisagul and B. Kruatrachue, “A modified particle swarm optimization with mutation and reposition,” ICIC International journal, Vol.10, No.6, December 2014.

[26] D. Tian, “Particle Swarm Optimization with Chaotic Maps and Gaussian Mutation for Function Optimization,” International Journal of Grid and Distributed Computing", Vol.8, No.4, 2015, pp.123-134.

[27] R. A. Jamous, A. A. Tharwat, E. EL.Seidy, and B. I. Bayoum, “Modifications of Particle Swarm Optimization Techniques and Its Application on Stock Market: A Survey,” IJACSA Journal, Vol.6, No. 3, 2015, pp.99–108.

[28] J. W. Li, Y. M. Cheng, and K. Z. Chen, “Chaotic particle swarm optimization algorithm based on adaptive inertia weight,” Control and Decision Conference (2014 CCDC), The 26th Chinese. Year: 2014, Pages: 1310 - 1315, DOI: 10.1109/CCDC.2014.6852369, IEEE Conference Publications.

[29] J. J. Craige, “Introduction to Robotics: Mechanics and Control,” Prentice-Hell, Inc., 2005, pp.180.

[30] N. Ramakoti, A. Vinay, and R. K. Jatoth, “Particle Swarm Optimization Aided Kalman Filter for Object Tracking,” IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies. Date of Conference: 28-29 Dec. 2009, Page(s): 531 – 533, Trivandrum, Kerala, DOI: 10.1109/ACT.2009.135.