

A Robust Incremental Algorithm for Predicting the Motion of Rigid Body in a Time-Varying Environment

Ashraf Elnagar

Department of Computer Science, University of Sharjah, Sharjah, UAE

Email: ashraf@sharjah.ac.ae

Received January 30, 2012; revised April 8, 2012; accepted May 5, 2012

ABSTRACT

A configuration point consists of the position and orientation of a rigid body which are fully described by the position of the frame's origin and the orientation of its axes, relative to the reference frame. We describe an algorithm to robustly predict futuristic configurations of a moving target in a time-varying environment. We use the Kalman filter for tracking and motion prediction purposes because it is a very effective and useful estimator. It implements a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance. The target motion is unconstrained. The proposed algorithm may be viewed as a seed for a range of applications, one of which is robot motion planning in a time-changing environment. A significant feature of the proposed algorithm (when compared to similar ones) is its ability to embark the prediction process from the first time step; no need to wait for few time steps as in the autoregressive-based systems. Simulation results supports our claims and demonstrate the superiority of the proposed model.

Keywords: Time-Varying Environments; Kalman Filtering; Rigid-Body Motion Prediction

1. Introduction

The importance of designing and developing robots is capable of performing a variety of tasks becoming of great interest to a large community. For example, autonomous robots to help and cooperate in unsafe environments in order to clean up hazardous wastes or to carry/handle radioactive materials. For true autonomy in such tasks, a capability that would enable each moving robot to react adaptively to its surrounding environment is needed while carrying out a certain task. For instance, when a robot navigates between two configurations, it should recognize the presence of static and moving obstacles and constantly update its knowledge of the environment. The situation is similar to that of a person crossing a street.

Despite of the different advances in the field of robot motion planning, there are still a number of complex problems which require more study and investigation. Uncertainty is an important factor that should be considered when addressing this problem. Uncertainty is a result of partial knowledge about the environment, in which a robot moves and noisy data captured by sensors. This problem is more evident in time-varying (dynamic) environments.

Although extensive research was reported on the problem of motion planning in static environments (e.g., see

[1,2] for a survey), few studies tackled the problem in time-varying environments, for example [3-10]. All of these works assume complete knowledge about the environment and a full control of the motion of obstacles.

Few studies dealt with the problem of estimating (or predicting) future positions of moving objects. These studies have used different techniques such as autoregressive models [11-13], collision cones [14], neural networks [15], fuzzy control systems [16], and potential fields [17]. Such estimation is central for a robot moving in a time-varying environment and avoiding obstacles while deciding about its next configuration.

The problem of motion planning may be subdivided into three interrelated phases: sensor integration and data fusion; scene interpretation and map building; and trajectory planning. Each of these phases consists of several sub-problems. One of which is the prediction problem that deals with predicting future positions and orientations of moving obstacles. This information is required for trajectory planning of the robot in order to avoid any possible future collisions. In the case of humans, the prediction procedure is usually characterized by a high performance and rarely misses its objective. This may be because of the accurate decisions we make based on the data collected through our biological sensors and what we predict about over a period of time.

In this work, we address the problem of predicting

next configurations for moving objects in a time-varying environment. We propose an algorithm which predicts future positions and orientations of freely moving obstacles using a Kalman filter. To make our analysis practical and more realistic, we do not assume any control over the trajectories of moving obstacles or the robot. We assume that previous and current positions and orientations are available from sensory devices. One advantage of this model when compared to others is the fact the prediction process starts from the first time step without any delay. The Kalman filter is a mathematical model that implement a predictor-corrector that is optimal in the sense that it minimizes the estimated error covariance assuming some presumed conditions are met. It has been the subject of extensive research and application. This is likely due to the relative simplicity and robustness of the filter itself. It apparently works well for many applications in spite of the absence of the conditions necessary for optimality. The Kalman filter has been used extensively for tracking in interactive computer graphics [18]. It has also been used for mo static and dynamic registration in computer graphics [19], and it is used for multi-sensor fusion in tracking systems [20].

The paper is organized as follows. Section 2 describes the prediction model which includes the mathematical equations for predicting positions and orientations of a moving object. The complete algorithm is discussed in Section 3. Simulation results are demonstrated in Section 4 and concluding remarks are made in Section 5.

2. Kalman Filter

In this section we develop the prediction model in order to decide about future configurations (a configuration = position + orientation) of moving objects. The following set of equations constitute the Kalman filter for the model used [21]:

$$\begin{aligned}\hat{\mathbf{x}}(k) &= A\hat{\mathbf{x}}(k-1) + K(k)[\hat{\mathbf{y}}(k) - CA\hat{\mathbf{x}}(k-1)] \\ K(k) &= AP_1(k)C^T [CP_1(k)C^T + R(K)]^{-1} \\ P_1(k) &= AP(k-1)A^T + Q(k-1) \\ P(k) &= P_1(k) - K(k)C(k)P_1(k)\end{aligned}$$

where $\hat{\mathbf{x}}(k)$ is the estimator, $\hat{\mathbf{y}}(k)$ is the noisy measurements, $K(k)$ is the filter gain, C is the observation matrix, $P_1(k)$ is the predicted covariance matrix and $P(k)$ is the error covariance matrix. The model symbols will be explained later in this section.

2.1. Translational Motion

It is assumed the robot is equipped with a set of sensors in order to collect data about its environment. Such data is vital for safe navigation. The data is collected in time

steps where each one represents a short period of time $\Delta(T)$. This enables the robot to learn about any moving obstacles in its visibility field at discrete points in the time-space. For now we are interested in the translational motion. Formally, let the position, of a translating obstacle, O_i , be x_i and velocity \dot{x}_i . Using vector notation, it is:

$$x_i(t) = \begin{bmatrix} x_i(t) \\ \dot{x}_i(t) \end{bmatrix} \quad (1)$$

If the sampling steps are small enough then it is fair to assume that the acceleration of a translating obstacle is constant or slowly changing. That is,

$$\ddot{x}(t) = c_i \quad (2)$$

where c_i is a constant value. Equation (2) may be represented (using (1)) in state-space form as

$$\dot{\mathbf{x}}(t) = F\mathbf{x}(t) + Gu(t) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_i(t) \\ \dot{x}_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [c] \quad (3)$$

In order to apply the Kalman filter, the difference equation is required:

$$\mathbf{x}(kT) = A\mathbf{x}[(k-1)T] + Bu(t) \quad (4)$$

where

$$\begin{aligned}A &= e^{FT} = I + FT + \frac{(FT)^2}{2!} + \dots \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} T + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}^2 \frac{T^2}{2} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\end{aligned} \quad (5)$$

and

$$B = \left(\int_0^T e^{F\tau} d\tau \right) G = F^{-1} (e^{FT} - I) G = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \quad (6)$$

T is the time period, $u(t)$ is the forcing factor, F is the state matrix (2×2 , where the dimensions are determined by the number of state variables), G is the excitation matrix (2×1 , where the dimensions are determined by the number of state and forcing variables, respectively.). Notice that A , B , F , G , and I are matrices; I is the unitary matrix; A and B are computed once. Assuming $T = 1$, we obtain:

$$\begin{aligned}\mathbf{x}(k) &= A\mathbf{x}[(k-1)] + Bu(t) \\ \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_i(k-1) \\ \dot{x}_i(k-1) \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix} [c] \\ &= \begin{bmatrix} x_i(k-1) + \dot{x}_i(k-1) + \frac{1}{2}c \\ \dot{x}_i(k-1) + c \end{bmatrix}\end{aligned} \quad (7)$$

which is the difference equation representing the moving

robot with constant acceleration. This system is observed by measuring the positions of the robot. However, positions are affected by some independent random disturbance $v(k)$, so that the observation equation can be expressed as

$$y(k) = Cx(k) + v(k) = [1 \ 0]x(k) + v(k) \quad (8)$$

with the random disturbance (noise) variance given by $R(k) = \sigma_v^2 = 1$. Moreover, we start with an initial value of the state vector, $\hat{x}(0)$, and the assumed errors at time $k = 0$, $P(0)$. To compute the state estimates, we start with the following covariance equation:

$$P_1(k) = AP(k-1)A^T + Q(k-1) \quad (9)$$

with input noise $Q(0) = 0$ for $k = 1$, we proceed as follows:

$$P_1(1) = AP(0)A^T \quad (10)$$

then using the following equation (filter gain):

$$K(k) = P_1(k)C^T [CP_1(k)C^T + R(k)]^{-1} \quad (11)$$

Using (10), C , and assuming $R(k) = 1$, we obtain:

$$K(k) = \frac{1}{P_{11} + 1} \begin{bmatrix} P_{11} \\ P_{21} \end{bmatrix} \quad (12)$$

Next, we calculate the prediction term (estimator)

$$\hat{x}(k) = \hat{x}'(k) + K(k)[\hat{y}(k) - CA\hat{x}'(k)] \quad (13)$$

where

$$\hat{x}'(k) = A\hat{x}'(k-1) + Bu \quad (14)$$

All the components required to compute the estimates (or filtering equation) are already determined. The errors are computed by the error covariance matrix:

$$P(k) = P_1(k) - K(k)C(k)P_1(k) \quad (15)$$

As a consequence of evaluating (12) in (15), the resultant diagonal terms $P_{11}(k) = P_{22}(k)$ are the ones of interest because they represent the mean-square errors of position and velocity. Since velocity affects position, we notice that good estimates of position are obtained only after obtaining good estimates for velocity. **Figures 1 and 2** show actual versus predicted trajectories of a translating object along X-axis and Y-axis, respectively. **Figures 3 and 4** depict the errors in position and speed, respectively. **Figure 5** depicts the actual (o) versus predicted (*) trajectories of the moving object in 2D.

2.2. Rotational Motion

In general, a moving object undergoes a combination of translation and rotational motion. Therefore, a model for predicting rotational motion is necessary to complete the prediction model. Without loss of generality, we

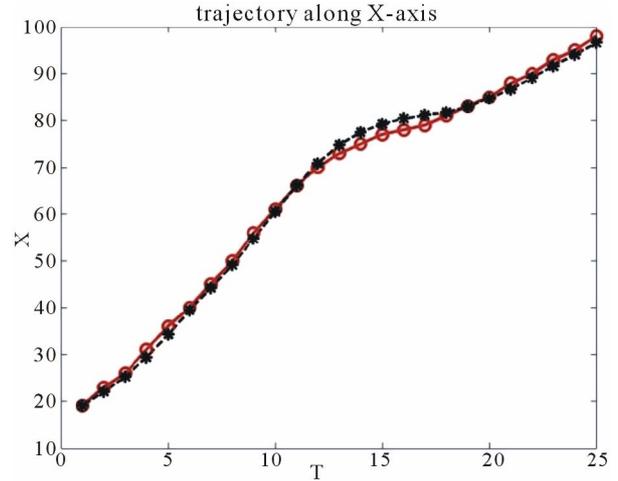


Figure 1. Actual vs. predicted trajectories of a translating object (point) along the X-axis.

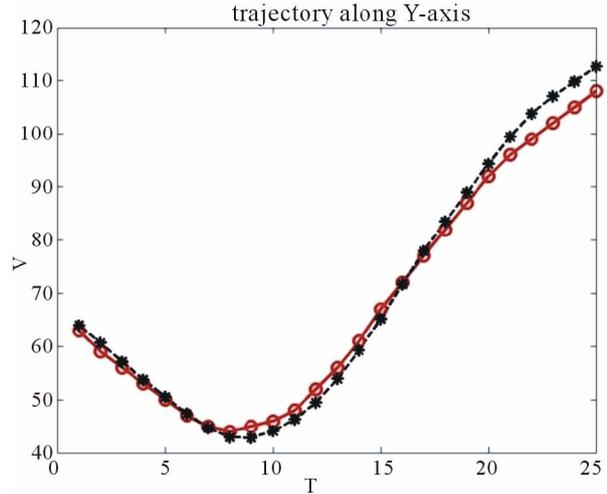


Figure 2. Actual vs. predicted trajectories of a translating object (point) along the Y-axis.

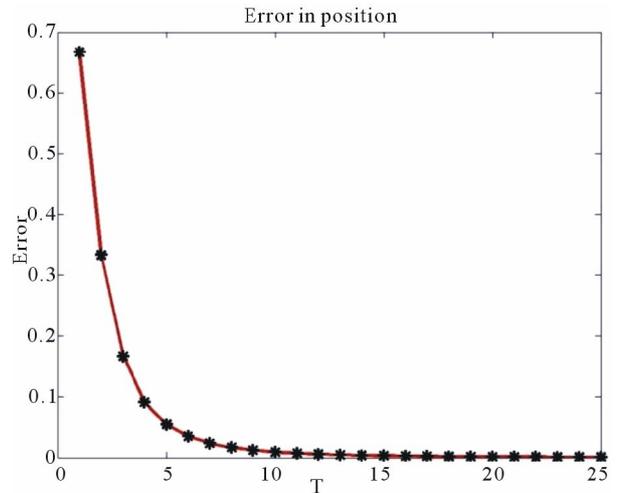


Figure 3. The error in position.

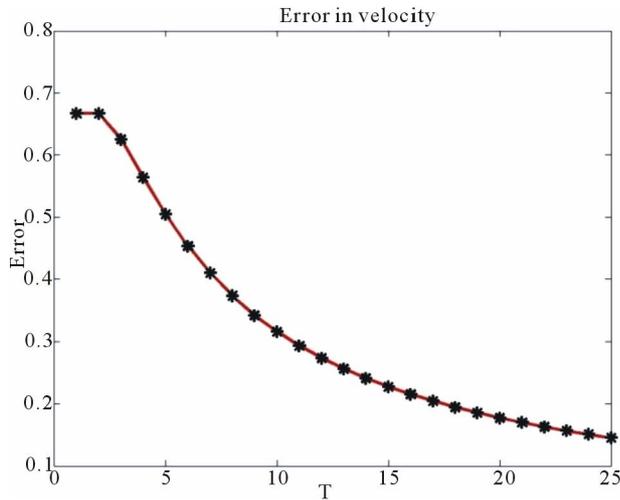


Figure 4. The error in velocity.

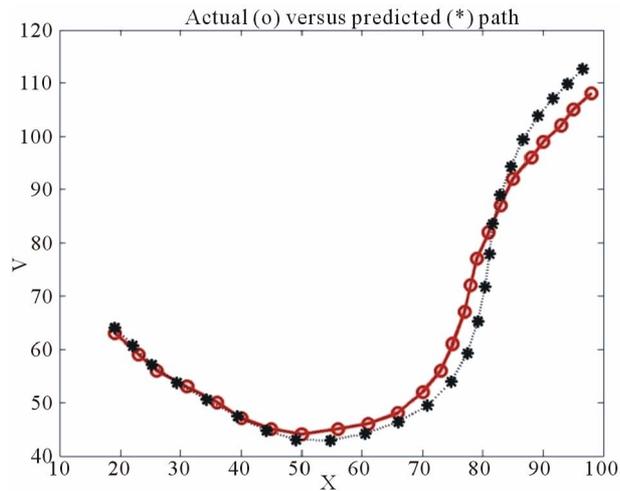


Figure 5. Actual (o) versus predicted (*) trajectories.

represent a given moving object with its center of mass and some other reference (feature) points that be in the right place on the object. For example, a line segment in space is defined by its center of mass and its end-points. We only predict the trajectory of the center of mass and then relate the computations to the other reference points. Reference points are used to show the orientation of a given object. The mathematical analysis for developing the rotational-prediction model is analogous to the one of the translational case. Formally, let the orientation, of a rotating obstacle, O_i , be θ_i and angular velocity $\dot{\theta}_i$. Using vector notation, it is:

$$\theta_i(t) = \begin{bmatrix} \theta_i(t) \\ \dot{\theta}_i(t) \end{bmatrix} \quad (16)$$

If the sampling steps, of sensing the environment, are small enough then it is fair to assume that the angular acceleration of a rotating obstacle is constant (d_i) or

slowly changing. That is,

$$\ddot{\theta}_i = d_i \quad (17)$$

The analysis follows exactly as for the translational case. The resulting model can be used to predict orientation around X and Y axes.

3. Prediction Algorithm

Table 1 describes the main steps in order to generate all predicted configurations in 2D.

We integrate both prediction models introduced so far. To illustrate this procedure, we use an example. Suppose a line segment, represented by three points $a = (a_1, a_{cm}, a_3)$ is moving freely in a 3-D environment. We only predict the trajectory of center-of-mass point (a_{cm}) that is regarded as a feature point. In view of the fact that we deal with rigid bodies, the prediction findings are applied to the other two end-points. The expected orientation of the line segment at each new predicted position is also computed. Using both the expected position and orientation, a point ${}^N A$, that belongs to the line segment at the current frame of center of mass reference (N) is mapped to its corresponding position in the global frame of reference (W). Formally,

$$\begin{bmatrix} {}^W A \\ 1 \end{bmatrix} = {}^W_N T \begin{bmatrix} {}^N A \\ 1 \end{bmatrix} \quad (18)$$

where ${}^W_N T$ is a 4×4 homogeneous transformation matrix defined as:

Table 1. Algorithm 1.

```

Initialize  $P(0)$ ,  $x_1(0)$  and  $x_2(0)$ 
For i : 1 to No_of_obstacles do
    For t=1 to N do
        if t < 4 then
            a=c
        else
            a=x(t-1)-2x(t-2)+x(t-3)
             $x_1(t)=x_1(t-1)+x_2(t-1)+0.5*a$ 
             $x_2(t)=x_2(t-1)+a$ 
             $P_{11}=P_{11}+P_{22}+2P_{21}$ 
             $P_{21}=P_{21}+P_{22}$ 
             $K(1)=\frac{P_{11}}{1+P_{11}}$ 
             $K(2)=\frac{P_{21}}{1+P_{11}}$ 
             $Z=Y(t)-x_1(t)$ 
             $x_1(t)=x_1(t)+K(1)*Z$ 
             $x_2(t)=x_2(t)+K(2)*Z$ 
             $P_{22}=P_{22}-K(2)P_{21}$ 
             $P_{21}=P_{21}-K(2)P_{11}$ 
             $P_{11}=(1-K(1))P_{11}$ 
            Save predicted values and errors
        Next t
    Next i
    
```

$${}^w_N T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The 3×3 sub-matrix ${}^w_N R = [r_{11} \dots r_{33}]$ represents the rotation matrix relating the current frame of reference (N) to the global one (W), and $[d_x, d_y, d_z]$ denotes the displacement vector from the origin of N with respect to frame W . We use the roll, pitch, yaw angles representation¹ to describe the orientation of N in W . **Table 2** summarizes the steps required to predict the $(n + 1)^{th}$ future position and orientation of a free moving object in space based on its first n positions and orientations:

4. Simulation Results

We presume a 2D environment in which an object is freely moving. Based on its past positions, configurations are estimated using the proposed model. The prediction is carried out over 25 sampling steps. Data points are arbitrary chosen.

In **Figures 1-4**, we predict the positions and orientations of the moving object that is specified with a center of mass and 3 feature points (vertices). The object translates freely in 2D. **Figures 1** and **2** show both the actual sensed and the predicted trajectories of the object translational motion in 2-D along X and Y axes, respectively. The errors between actual and predicted values in position and velocity are depicted in **Figures 3** and **4**, respectively. Note that the position error drops after the first observation is processed whereas the velocity error drops after the second observation. The mean square errors are 1.12 and 2.01 distance-units along X and Y axes, respectively. The predicted path is quite close to the actual trajectory as can be seen in **Figure 5**.

Table 2. Algorithm 2.

Input previous positions and orientations.
For $i : 1$ to $No_of_obstacles$ do
For each feature point $\in O$, do
Using Algorithm 1,
1. predict next configuration: position (x_p, y_p) and
orientation (θ_x, θ_y) .
2. apply transformation and obtain the resulting
configuration.
Next
Next i

$${}^w_N R_{XYZ} = [\theta_x, \theta_y, \theta_z]$$

$$= \begin{bmatrix} c\theta_x c\theta_y & c\theta_x s\theta_y c\theta_z - s\theta_x c\theta_z & c\theta_x s\theta_y s\theta_z + s\theta_x s\theta_z \\ s\theta_x c\theta_y & s\theta_x s\theta_y c\theta_z + c\theta_x c\theta_z & s\theta_x s\theta_y s\theta_z - c\theta_x s\theta_z \\ -s\theta_y & c\theta_y s\theta_z & c\theta_y c\theta_z \end{bmatrix}$$

where $c\theta$ is shorthand for $\cos\theta$ and $s\theta$ for $\sin\theta$, etc.

The effect of rotational motion of the same object is depicted in **Figure 6**. The mean square error is 0.369 radians (**Figure 7**). **Figure 8** shows the actual and predicted trajectories for both translation and rotational motion. The effect of rotation may be traced by tracking the symbols (*) and (+), which indicates the orientation (angle around X-axis). The details of this figure are already explained in the previous figures.

5. Conclusion

We have described a robust algorithm to predict futuristic configurations of a freely moving target in a time-varying environment. We employed the Kalman filter for tracking and motion prediction purposes for the reason that it is an effective and useful predictor-corrector estimator. It is optimal in the sense that it minimizes the es-

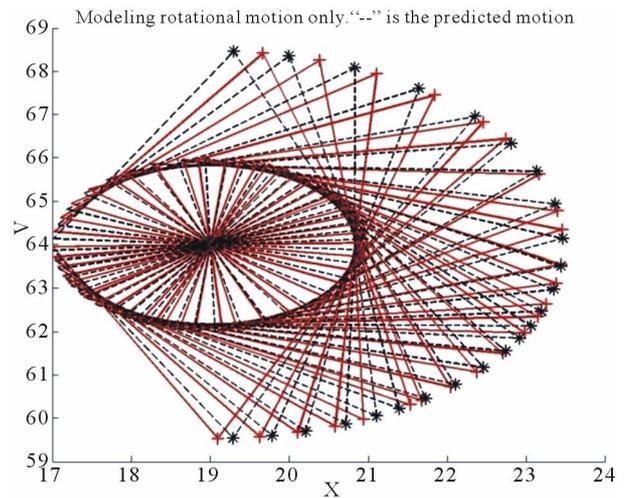


Figure 6. Predicting rotational motion only for an object in 2D; orientation is indicated by the (+: actual) and (*: predicted) symbols.

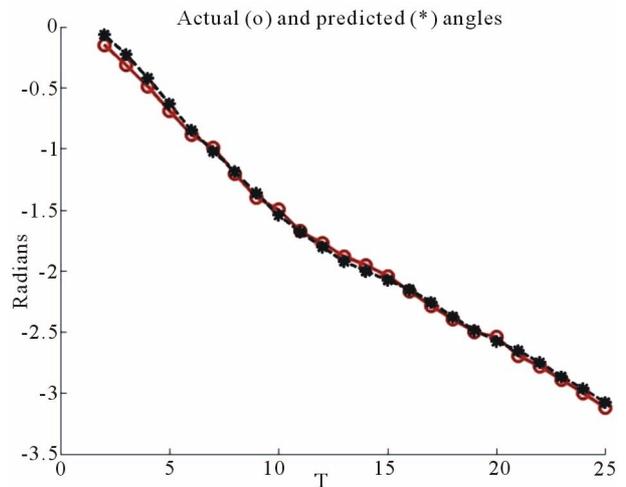


Figure 7. Actual (o) and predicted (*) rotational motion.

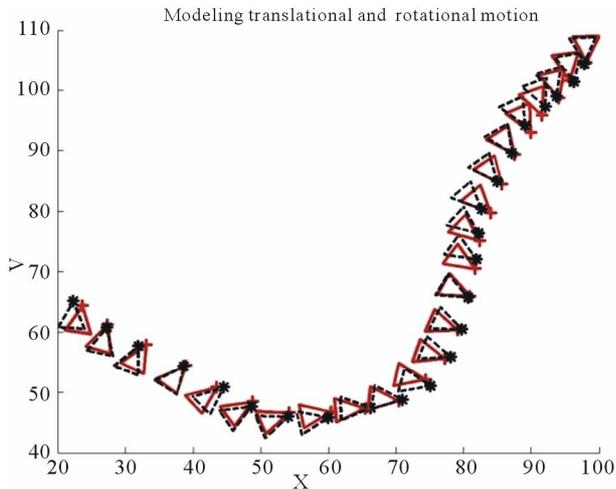


Figure 8. Actual (solid) and predicted (dashes) configurations of a moving object in 2D based on all previous figures.

timated error covariance. The target motion is unconstrained. The ability of the proposed algorithm to embark on the prediction process from the first time step rather than waiting for few time steps is a noteworthy quality; no need to wait for few time steps as in the autoregressive-based systems. Simulation results confirmed the validity of the proposed model and demonstrated its the superiority. This research may be viewed as a seed point for further research in robot motion planning in a dynamic environment.

REFERENCES

- [1] A. Basu and A. Elnagar, "Safety Optimizing Strategies for Local Path Planning in Dynamic Environments," *International Journal on Robotics Automation*, Vol. 10, 1995, pp. 130-142.
- [2] R. Azuma and G. Bishop, "Improving Static and Dynamic Registration in an Optical See-Through HMD," *SIGGRAPH 94 Conference Proceedings*, ACM Press, Addison-Wesley, Orlando, 1994, pp. 197-204.
- [3] S. Bozic, "Digital and Kalman Filtering," Edward Arnold Publishers, London, 1994.
- [4] A. Chakravarthy and D. Ghose, "Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach," *IEEE Transactions on SMC-Part A: Systems and Humans*, Vol. 28, No. 5, 1998, pp. 562-574. [doi:10.1109/3468.709600](https://doi.org/10.1109/3468.709600)
- [5] C. Chang and K. Song, "Environment Prediction for a Mobile Robot in a Dynamic Environment," *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 6, 1997, pp. 862-872. [doi:10.1109/70.650165](https://doi.org/10.1109/70.650165)
- [6] A. Elnagar, "Prediction of Future Configurations of a Moving Target in a Time-Varying Environment Using an Autoregressive Model," *Journal of Intelligent Control and Automation*, Vol. 2, No. 4, 2011, pp. 284 -292. [doi:10.4236/ica.2011.24033](https://doi.org/10.4236/ica.2011.24033)
- [7] A. Elnagar and K. Gupta, "Motion Prediction of Moving Objects Based on Autoregressive Model," *IEEE Transactions on SMC-Part I: Systems and Humans*, Vol. 28, No. 6, 1998, pp. 803-810. [doi:10.1109/3468.725351](https://doi.org/10.1109/3468.725351)
- [8] P. Fiorini and Z. Shiller, "Time Optimal Trajectory Planning in Dynamic Environments," *IEEE International Conference on Robotics and Automation*, Vol. 2, 1996, pp. 1553-1558.
- [9] E. Foxlin, M. Harrington and G. Pfeifer, "Constellation™: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications," In: M. F. Cohen, Ed., *Computer Graphics (SIGGRAPH 98 Conference Proceedings)*, Orlando, ACM Press, Addison-Wesley, 1998, pp. 371-378.
- [10] T. Fraichard and C. Laugier, "Path-Velocity Decomposition Revisited and Applied to Dynamic Trajectory Planning," *Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, 2-6 May 1993, pp. 40-45.
- [11] K. Fujimura and H. Samet, "Motion Planning in a Dynamic Environment," *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, 14-19 May 1989, pp. 324-330.
- [12] Y. Hwang and N. Ahuja, "Gross Motion Planning—A Survey," *ACM Computing Surveys*, Vol. 24, No. 3, 1992, pp. 219-291. [doi:10.1145/136035.136037](https://doi.org/10.1145/136035.136037)
- [13] K. Intersense, "IS-900," 2000. <http://www.isense.com>
- [14] K. Kant and S. W. Zucker, "Towards Efficient Trajectory Planning: The Path-Velocity Decomposition," *The International Journal of Robotics Research*, Vol. 5, No. 3, 1986, pp. 72-89. [doi:10.1177/027836498600500304](https://doi.org/10.1177/027836498600500304)
- [15] N. Kehtarnavaz and N. Griswold, "Establishing Collision Zones for Obstacles Moving with Uncertainty," *Computer Vision, Graphics and Image Processing*, Vol. 49, No. 1, 1990, pp. 95-103. [doi:10.1016/0734-189X\(90\)90165-R](https://doi.org/10.1016/0734-189X(90)90165-R)
- [16] J.-C. Latombe, "Robot Motion Planning," Kluwer Academic Publishers, London, 1991.
- [17] K. Pratihari, D. Deb and A. Ghosh, "A Genetic-Fuzzy Approach for Mobile Robot Navigation among Moving Obstacles," *International Journal of Approximate Reasoning*, Vol. 20, No. 2, 1999, pp. 145-172. [doi:10.1016/S0888-613X\(98\)10026-9](https://doi.org/10.1016/S0888-613X(98)10026-9)
- [18] R. Spense and S. Hutchinson, "An Integrated Architecture for Robot Motion Planning and Control in the Presence of Moving Obstacles with Unknown Trajectories," *IEEE Transactions on SMC*, Vol. 25, No. 1, 1995, pp. 100-110.
- [19] T. Tsubouchi, K. Hiraoka, T. Naniwa and S. Arimoto, "A Mobile Robot Navigation Scheme for an Environment with Multiple Moving Obstacles," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots, Raleigh*, 7-10 July 1992, pp. 1791-1798.
- [20] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller and D. Colucci, "High-Performance Wide-Area Optical Tracking—The HiBall Tracking System," *Presence: Teleoperators and Virtual Environments*, Vol. 10, No. 1, 2001.