IMTC 2003 – Instrumentation and Measurement
Technology Conference
Vail, CO, USA, 20-22 May 2003

Improving Predictive Control of a Mobile Robot: Application of Image Processing and Kalman Filtering

C. H. Messom[1], G. Sen Gupta[2,3], S. Demidenko[3] and Lim Yuen Siong[2]
[1]II&MS, Massey University, Albany, New Zealand
[2]School of EEE, Singapore Polytechnic, 500 Dover Road, Singapore
[3]IIS&T, Massey University, Palmerston North, New Zealand
Email: C.H.Messom@massey.ac.nz, G.SenGupta@massey.ac.nz
S.Demidenko@massey.ac.nz, LimYS@sp.edu.sg

Abstract ***– This paper discusses a control algorithm for the interception of a mobile target. The application domain is robot soccer in which the target is a ball while the interceptor is a wheeled robot. A case study of a wheeled robot approaching a target in a given direction is presented in detail. A shoot function has been developed, using the proposed algorithm, which calculates the robot wheel velocities to position the robot behind the ball. In order to improve the efficiency of the control, a scaling factor has been introduced which can be optimized so that the robot travels a shorter distance to intercept the target. Since the target is constantly moving, it is imperative that the future position of the target be predicted and used in the control algorithm to calculate the wheel velocities. Kalman filtering has proven to be an effective tool for predicting the target position at the point of interception, several frames ahead and thus improve the accuracy of interception.***

I. INTRODUCTION

In the last decade robot soccer has been extensively used as a test bed for adaptive control of dynamic systems in a multi-agent collaborative environment [1]. Because of the dynamics and high complexity of the robot soccer system as well as manoeuvrability and speed of its robots, the accurate path planning and prediction of moving targets have gained special importance. Several techniques have been proposed for path planning including the use of genetic algorithms [2] and fuzzy logic [3]. Coupled to the path-planning problem is obstacle avoidance in a dynamic environment. The path plans must be dynamically updated to reflect the changes in the environment, which means that they have to be created in real-time [4].

Robots have to exhibit basic actions like positioning at a designated location, moving to the ball or blocking an opponent, turning to a desired angle, circling around the ball and shooting the ball into opponent's goal. Among other factors, the strategy and path planning in a robot soccer game are dependent on the ball position [5].

The robot's main sensor system is vision, which captures and processes the image at 30 to 60 frames per second [6]. The odd and even scan fields of the interlaced image are processed separately. For this reason a stationary object like the ball is reported at different locations in each frame due to different quantization errors. The errors are compounded by the variation of light intensity from one frame to another. These quantization errors are inherent in the system and have a significant impact on the shooting accuracy even when the ball is stationary. However, the tests carried out on a moving ball are more significant because the interception accuracy suffers when the ball is moving. This is due to the fact that control actions are initiated based on the current 'static' state of objects whereas action must be taken based on predicted future positions.

Robotic interception of moving objects in industrial settings has gained much attention in the literature in recent years. Among the most efficient approaches is APPE (Active Prediction, Planning and Execution) system for robotic interception of moving objects [7]. The key feature of the system is the ability of a robot to perform a task autonomously without complete *a priori* information. The APPE system has been tested for an object moving at a maximum velocity of 45mm/s.

Kalman filtering [8, 9] has proved very useful in autonomous and assisted navigation and guidance systems, radar tracking of moving objects, etc. The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) solution of the least-squares method. The filter is very powerful in several aspects: it supports estimates of past, present, and even future states, and it can do so even when the precise nature of the modelled system is unknown. Kalman filtering is also a computationally efficient algorithm, which generates an optimal estimate from a sequence of noisy observations.

This paper discuses an implementation of a robotic interception (a shoot function in robot soccer) based on image capture/processing combined with the successful use of Kalman filtering aiming at substantial improvement in shooting accuracy both for a stationary and moving object (the ball). Several studies have implemented real-time image processing [10, 11] and robot localisation [12] in the robot soccer domain but have not achieved the target interception success rate of the system presented in this paper. The proposed technique has been developed and tested for a fast moving target of up to 1.5 m/s. The system successfully predicts the trajectory of the target through the robot's workspace, plans the robot motion to reach the target at a given angle and executes that plan. The system is robust in an environment with substantial noise introduced from variation in ambient light, shadows and reflections.

II. EXPERIMENTAL SETUP

The experimental setup is shown in Fig. 1. Among other items it includes a Pulnix 7EX NTSC camera and a FlashBus MV Pro image capture card [13]. The camera is mounted above the field as a global vision system that is available to all the robots. The robots are equipped with color tags to aid the vision system in their identification.

The image is captured at a resolution of 320 x 480 at a sampling rate of 30Hz. The odd and even fields of the interlaced image are processed separately. Hence the effective image resolution is 320 x 240 is delivered at a sampling rate of 60Hz. Processing the odd and even fields separately introduces quantization effects that must be compensated by the system. The captured image is processed using a Pentium II PC (450MHz, 128MB RAM). The results of the vision processing are passed to the Control Algorithms Unit for implementation of *position*, *angle* and *shoot* control functions.

III. CONTROL ALGORITHM FOR OBJECT INTERCEPTION

The intercept (shoot) function is implemented by providing the robot with a target position inline with the ball and goal that progressively approaches the ball. This is the position **P** (see Fig. 2), a point behind the stationary object (ball) inline with the target. The velocities of the left and the right wheel, V_L and V_R of the robot that move it to the position **P** and finally orient it to face the target are calculated by the intercept (shoot) algorithm.

The angles shown in Fig. 2 are:

ϕ_1 - *TargetAngle*
(negative value in the case under consideration)
ϕ_2 - *BallAngle* (positive value)
ϕ_2-ϕ_1 - *BallTargetAngle*
ϕ_3 - *DesiredDirectionAngle*
θ_e - *AngleArror*, and
θ_r - *RobotAngle*

The robot's desired direction angle ϕ_3 is

$$\phi_3 = \phi_2 + (\phi_2 - \phi_1) = 2\phi_2 - \phi_1 \tag{1}$$

The angle error θ_e between the required direction and the current robot direction is

$$\theta_e = \phi_3 - \theta_r \tag{2}$$

The wheel velocities are calculated using the following pair of the equations:

$$V_L = V_c - K_a\theta_e, \text{ and } V_R = V_c + K_a\theta_e, \tag{3}$$

K_a is the constant proportional gain. When the robot is in line with the target the angle error θ_e is equal to zero, the robot moves straight towards the target with a constant velocity V_c.

The above control algorithm provides considerable good stationary target interception – ball shooting. However its accuracy is not high. Since the image is processed in two parts – the odd and even scan fields separately – the position of the stationary ball shifts, by as much as 0.5cm (approximately 1 pixel), as shown in Fig. 3. This results in incorrect calculation of wheel velocities.

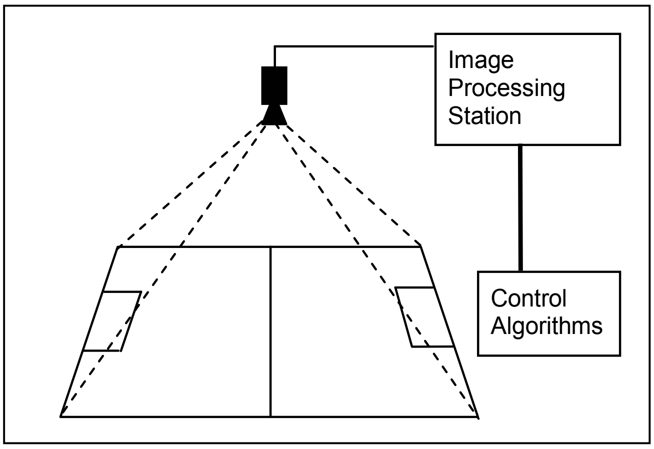

Fig. 1 Experimental setup

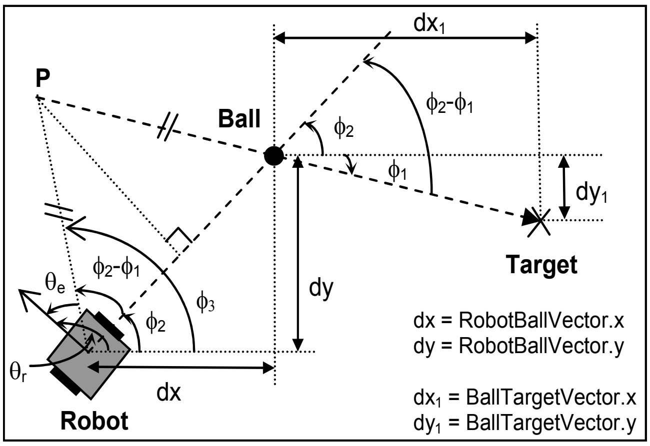

Fig. 2. Control over a stationary object interception

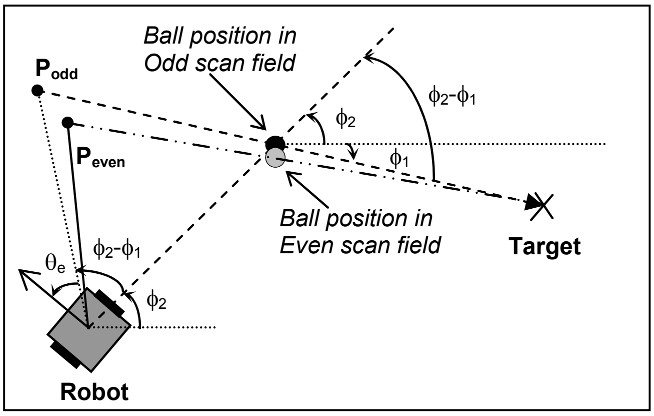

Fig. 3. Object position in odd and even scan fields

The problem is more severe when the ball is close to the target. This is due to the fact that a small change in the ball position alters the *Targetangle* by a large amount. From one field to the next, the difference in angle error $\delta\theta_e$ is quite large resulting in wider variations of V_L and V_R as the robot takes the shot. As such the trajectory followed by the robot is not smooth.

As a solution to the problem, a Kalman Filtering based approach is proposed, resulting in reduction of variation in calculated position. The filter can be described by the following equations:

$$\bar{\dot{x}}_n = \hat{\dot{x}}_n + \frac{h}{T}\left(y_n - \hat{x}_n\right) \tag{4}$$

$$\bar{x}_n = \hat{x}_n + g\left(y_n - \hat{x}_n\right) \tag{5}$$

where,

$\bar{\dot{x}}_n$: filtered (smoothed) estimate of velocity at discrete time n

$\hat{\dot{x}}_n$: predicted estimate of velocity at time n based on the sensor values at time *n-1*and all preceding times

$\bar{x}_n$: filtered (smoothed) estimate of position at time n

$\hat{x}_n$: predicted estimate of position at time n based on the sensor values at time *n-1*and all preceding times

T : sample time (scan-to-scan period) for the system

y_n : sensor reading at time n

h and g : filter parameters

These equations provide an updated estimate of the present object velocity and position based on the present measurement of object position, y_n, as well as prior measurements.

The predicted velocity and position at time n are calculated based on a constant velocity model:

$$\hat{\dot{x}}_n = \bar{\dot{x}}_{n-1} \tag{6}$$

$$\hat{x}_n = \bar{x}_{n-1} + T\bar{\dot{x}}_{n-1} = \bar{x}_{n-1} + T\hat{\dot{x}}_n \tag{7}$$

Equations (6) and (7) allow transition from the velocity and position at time *n-1* to the velocity and position at time *n*. These *transition equations* together with (4) and (5) allow tracking an object and are combined to give just two tracking filter equations:

$$\hat{\dot{x}}_n = \hat{\dot{x}}_{n-1} + \frac{h}{T}\left(y_{n-1} - \hat{x}_{n-1}\right) \tag{8}$$

$$\hat{x}_n = \hat{x}_{n-1} + T\hat{\dot{x}}_n + g\left(y_{n-1} - \hat{x}_{n-1}\right) \tag{9}$$

Equations (8) and (9) are used to obtain running estimates of target velocity and position.

The constant velocity model is suitable for the ball as it does not accelerate or decelerate significantly unless it hits an obstacle. Tracking the robots using a constant velocity model results in a steady state tracking error whenever the robots accelerate (or change direction).

To overcome this problem, a constant acceleration model is used to filter the robot's positions. The filter is described by (4), (5) and the following equation:

$$\bar{\ddot{x}}_n = \hat{\ddot{x}}_n + \frac{2k}{T^2}\left(y_n - \hat{x}_n\right) \tag{10}$$

where,

k : filter parameter

$\bar{\ddot{x}}_n$: filtered estimate of acceleration at discrete time n

$\hat{\ddot{x}}_n$: predicted estimate of acceleration at time n based on sensor values at time *n-1*and all preceding times.

Based on a constant acceleration model, the *transition equations* for prediction of position, velocity and acceleration are:

$$\hat{\ddot{x}}_n = \bar{\ddot{x}}_{n-1} \tag{11}$$

$$\hat{\dot{x}}_n = \bar{\dot{x}}_{n-1} + T\bar{\ddot{x}}_{n-1} \tag{12}$$

$$\hat{x}_n = \bar{x}_{n-1} + T\bar{\dot{x}}_{n-1} + \frac{T^2}{2}\bar{\ddot{x}}_{n-1} \tag{13}$$

The constant-acceleration object is tracked with zero lag error in the steady state.

The Kalman filtering addresses two issues. It lessens the vision quantization error due to separate processing of odd and even scan fields. At the same time it eliminates (or significantly reduces) the negative effect of the variation in the calculation of ball position due to light intensity fluctuations.

To further improve the efficiency of the control over the robot and thus to make it turn faster, the target position P (Fig. 2) can be adjusted to be closer to the ball. This can be achieved by adding a special scale factor k. Thus the equations (3) are modified to:

$$V_L = V_c - K_a(\phi_2 + k(\phi_2 - \phi_1) - \theta_r) \quad (14)$$

$$V_R = V_c + K_a(\phi_2 + k(\phi_2 - \phi_1) - \theta_r) \quad (15)$$

The result of the simulation of the modified algorithm is shown in Fig. 4. It can be seen that the robot travels along a shorter path and reaches the shooting position faster. The value of the scale factor k can be optimized for given angles and distances from the interception point.

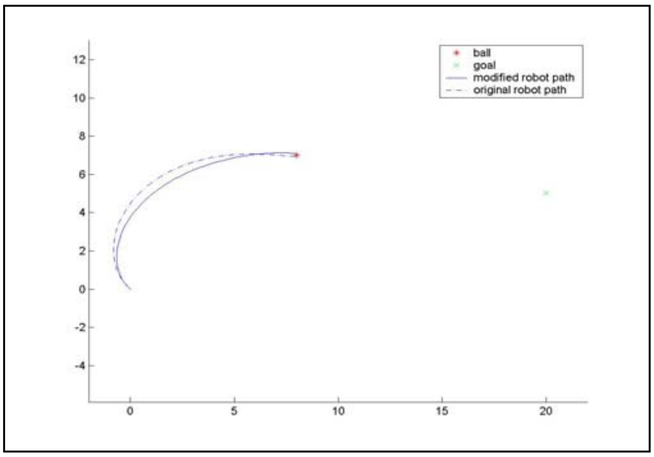

Fig. 4. Simulated path using original and modified control algorithms

IV. PREDICTING THE OBJECT'S POSITION SEVERAL SCANS AHEAD

In order to approach a moving target (e.g., the ball in a robot soccer game) the target's future position has to be predicted several scans ahead. This is where Kalman filtering has provided an effective tool enabling prediction of the ball position a few scans ahead and thus to achieve much greater target interception accuracy.

The ball position is predicted using a constant velocity model (16), while the robots' positions are predicted using a constant acceleration model (17):

$$\hat{x}_{n+t} = \hat{x}_n + t * T\bar{\dot{x}}_{n-1} \quad (16)$$

$$\hat{x}_{n+t} = \hat{x}_n + t * T\bar{\dot{x}}_{n-1} + (t * T)^2 \bar{\ddot{x}}_n / 2 \quad (17)$$

The constant velocity model of the ball is realistic for a fast moving ball that does not decelerate significantly, while the constant acceleration model of the robot is suitable for an accelerating robot or a robot moving on a circular path.

The position and time that the ball and robot will intersect is found by solving the simultaneous equations given by (16) and (17). In general if the two paths intersect there are two real valued solutions. Positive ***t*** are required for the collision points that will occur in the future while negative ***t*** values indicate collisions points that could have occurred in the past. If there are two positive solutions for ***t*** then the smaller value represents the first collision. When there are no real valued solutions to the simultaneous equations, the robot and ball paths are predicted to miss each other. In this scenario the linear prediction model of the robot motion can be used to get an approximation of the collision point. The predicted collision points are used to calculate robot's desired angle (1) and angle error (2) which are required for the calculation of the control action (3, 14 and 15).

V. EXPERIMENTAL RESULTS

By applying filtering, the variation in the position of a stationary ball was reduced from $\pm$0.5 cm to $\pm$0.3 cm. This resulted in substantial improvement of the accuracy of shooting. For a *BallTargetAngle* < 90^o, the ball was shot with 100% success rate.

The function was put to test during the Singapore Robotic Games (SRG) 2002 and FIRA Robot World Soccer Competition in South Korea in 2002. For a *BallTargetAngle* > 90^o, a State Transition Based Control algorithm [5] was implemented to position the robot at a good "*behind position*" first. After that, only when the *BallTargetAngle* became < 90^o and *RobotBallVector.x* was > 15cm, the shoot function was activated. The overall improvement in the team's performance was remarkable and several games were won.

The system based on the above control and prediction techniques was tested for an object (the ball) moving at up to 1.5m/s. The system successfully predicted the trajectory of a ball through the robot's workspace. Depending on the distance of the robot from the ball, the magnitude of the *RobotBallVector,* a ball prediction of 4 to 10 frames ahead was tested for shooting a moving ball with reasonable success. The only case when the prediction failed was when the *BallTargetAngle* > 90^o since the State Transition Based Control Algorithm does not guarantee that the robot will be placed in a good position behind the target in time to intercept the target. Other failure modes included when the target (ball) was near an obstacle (another robot) or the edge of the soccer field. In these scenarios a higher level behavior selection module is required to select alternative strategies.

VI. CONCLUSIONS

This paper has studied the application of Kalman filtering to motion tracking and trajectory prediction using a real-time video image processing system. The filter was evaluated on a target interception application, shooting the ball in a robot soccer system. The experiments show that the measurement error and the variation are significantly reduced by the filtering system. The target interception rate was also improved with the predictive filtering applied to both the target and the interceptor.

Future work includes improving the prediction of target position taking into considerations such as collisions with obstacles such as the walls of the soccer field. Additionally the control of the interceptor system can be improved, so that an optimal trajectory is followed. This requires finding optimal values of K_a and k for various angle error θ_e values (3, 14 and 15), which can then be gain scheduled. Another enhancement of the velocity calculation algorithm is to incorporate a derivative of the angle error to further dampen sudden swings in the robots angular position.

ACKNOWLEDGEMENT

The collaborative work was supported by a grant from ASIA 2000 HEEP funds and carried out jointly at Massey University and the Advanced Robotic and Intelligent Control Centre, Singapore Polytechnic.

REFERENCES

[1] C. H. Messom, "Robot Soccer – Sensing, Planning, Strategy and Control, a distributed real time intelligent system approach", AROB, Oita, Japan, 1998, pp. 422 – 426.

[2] Piao Songhao, Hong Bingrong, "Robot path planning using genetic algorithms", Journal of Harbin Institute of Technology, Vol. 8, No. 3, Sept 2001, pp215 – 217

[3] Meng Qingchun, Yin Bo, Xiaong Jianshe, "Intelligent learning technique based on fuzzy logic for multi-robot path planning", Journal of Harbin Institute of Technology, Vol. 8, No. 3, Sept 2001, pp. 222 – 227

[4] Jacky Baltes and Nicholas Hildreth, "Adaptive Path Planner for Highly Dynamic Environments", P. Stone, T. Balch, G. Kraetzschmar (Eds.): RoboCup 2000: Robot Soccer. World Cup IV Spinger Verlag 2001, pp 76 - 85.

[5] G. Sen Gupta, C. H. Messom, Sng HL, "State Transition Based Supervisory Control for a Robot Soccer System", 1st IEEE International Workshop on Electronic Design test and Applications DELTA'2002, January 29-31, 2002, Christchurch, New Zealand, p. 338-342

[6] M. Jamzad, B.S. Sadjad, V.S. Mirrokni, M. Kazemi, H. Chitsaz, A. Heydarnoori, M.T. Hajiaghai, and E. Chiniforoosh, "A Fast Vision System for Middle Size Robots in RoboCup", A. Birk, S. Coradeschi, S. Tadokoro (Eds.): RoboCup 2001: Robot Soccer World Cup V, Spinger Verlag 2002, pp 71 - 80.

[7] Damir Hujic et el, "The Robotic Interception of Moving Objects in Industrial Settings: Strategy Development and Experiment", IEEE/ASME Transactions on Mechatronics, Vol 3, No 3, September 1998, pp. 225 – 239

[8] Barry Cipra, "Engineers Look to Kalman Filtering for Guidance", SIAM News, Vol 26, No. 5, August 1993.

[9] R. E. Kalman, "A new approach to linear filtering and prediction problems", Trans. ASME J. Basic Eng., Vol 82D, 1960, pp. 35 – 45

[10] Pieter Jonker, Jurjen Caarls, and Wouter Bokhove, "Fast and Accurate Robot Vision for Vision Based Motion", P. Stone, T. Balch, G. Kraetzschmar (Eds.): RoboCup 2000: Robot Soccer. World Cup IV Spinger Verlag 2001, pp 149 - 158.

[11] Mark Simon, Sven Behnke, and Raúl Rojas, "Robust Real Time Color Tracking", P. Stone, T. Balch, G. Kraetzschmar (Eds.): RoboCup 2000: Robot Soccer. World Cup IV Spinger Verlag 2001, pp 239 - 248.

[12] Jens-Steffen Gutmann, Thilo Weigel, and Bernhard Nebel, "Fast, Accurate, and Robust Self-Localization in the RoboCup Environment", Manuela Veloso, Enrico Pagello, and Hiroaki Kitano (Eds.): RoboCup-99. Robot Soccer World Cup III, Springer Verlag, 2000 pp 304 - 317.

[13] Integral Technologies Inc, framegrabber specifications [Online]. Available: http://www.integraltech.com/Products/FrameGrabbers.html