H_∞ Filtering for a Mobile Robot Tracking a Free Rolling Ball

Xiang Li and Andreas Zell

Wilhelm-Schickard-Institute, Department of Computer Architecture,
University of Tübingen, Sand 1, 72076 Tübingen, Germany
{xiang.li,andreas.zell}@uni-tuebingen.de

Abstract. This paper focuses on the problem of tracking and predicting the location and velocity of a rolling ball in the RoboCup environment, when the ball is pushed consecutively by a middle-size omnidirectional robot to follow a given path around obstacles. A robust algorithm based on the H_∞ filter is presented to accurately estimate the ball's real-time location and velocity. The performance of this tracking strategy was also evaluated by real-world experiments and comparisons with the Kalman filter.

1 Introduction

In many mobile robots applications, the robots are required not only to adapt themselves to the external situation, but also have the ability to interact with the environment. Estimating and predicting the motions of moving objects are the foundation for the interaction tasks. For example, when robots play the football, it is very important to detect and predict the ball's position and velocity, so that the robot can catch the ball, push it through obstacles, and shoot it in the goal. In this paper we focus on tracking and predicting the location and velocity of a rolling ball in the RoboCup domain with a middle-size omnidirectional robot, under the condition that the ball is consecutively pushed by the robot.

Kalman filters ([1], [2], [3], [9]) have been used in many ball tracking problems. They provide efficient and convenient minimum-mean-square-error solutions for the state estimation problem, considering that both the process and the measurement noises of the target system are assumed as Gaussian with known statistical properties ([8]). Besides that, multiple model filters based on Kalman filters reveal much better performance than the single model filter in some applications. As one of the multiple model filters, the interacting multiple model (IMM) algorithm, used for the object tracking in the RoboCup ([5]), utilizes a Kalman filter for each mode of the target movement. However, in practical situations, the uncertainties of the target system and the measurements normally do not satisfy the Gaussian assumption, and the noise statistics is usually not available.

To avoid thinking about these uncertainties, a method to build a predictive model of the ball's movement is used in the estimation of the ball's position and velocity ([7]). It models a free rolling ball's movement as the linear movement and estimates the model parameters using ridge regression. By comparing the observed and predicted ball's positions, the method can also recognize the change points of the ball's movement. Due to the requirement of a buffer to store the observations of the ball's movement and the estimation of model parameters, the memory occupancy and computational complexity of this method are highly increased.

In this paper, we present a robust algorithm based on the H_∞ filter for an omnidirectional robot to track a rolling ball in the RoboCup domain. The H_∞ filter does not require priori knowledge of the noise statistics, only assuming that the noise signals have finite energy. Unlike the Kalman filter providing the minimum variance of the estimation error, the H_∞ filter provides the minimal effect of the worst noise on the estimation error. Experiments with a real omnidirectional robot show that this approach is efficient and yields highly robust estimations of the ball's location and velocity.

2 Problem Formulation

The ball tracking problems in the RoboCup domain are challenged by the interactions between the robots and the ball. These frequent interactions usually result in a highly non-linear movement of the ball, and it is very difficult to precisely estimate the uncertainty distribution of the interactions. Moreover, the measurement accuracy of the ball's position is also limited by the sensors and the corresponding signal processing algorithms. This paper focuses on tracking a rolling ball when it is consecutively pushed by an omnidirectional robot to follow a given path. Considering the uncertainty of the interactions between the robot and the ball, we utilize a new approach based on the H_∞ filter to estimate the ball's location and velocity.

The discrete representation of the ball's dynamics is described by the following equations:

$$\dot{p}_{k+1} = \dot{p}_k + \ddot{p}_k T \tag{1}$$

$$p_{k+1} = p_k + \dot{p}_k T + \frac{1}{2}\ddot{p}_k T^2 \ , \tag{2}$$

where p is the position of the ball, $\dot{p}$ and $\ddot{p}$ are respectively the velocity and acceleration of the ball. T is the sampling interval and k is the index of the sampling interval. We define a state vector consisting of the position and velocity as $x_k = [p_k, \dot{p}_k]^T$. Knowing the measurement value is the ball's position, we build the system model of the ball as follows:

$$x_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} u_k \tag{3}$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k \ , \tag{4}$$

where u denotes the system input and is equal to the acceleration $\ddot{p}$ which is completely determined by the friction of the ground and the pushing operation from the robot. But in the practical situation, the previous equation (3) can not give the precise state values because of the noise due to the rugged carpet ground and other unfortunate realities, and the precise output values can not be obtained from the equation (4), since measurement noise decreases the reliability of the measurement data. So a more precise mode is given as

$$x_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} u_k + w_k \tag{5}$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \ , \tag{6}$$

where w is called process noise and v is called measurement noise.

As we do not know exactly the friction of the ground, the moment when the robot collides the ball, and the corresponding effect of the collision on the ball's movement, the system input u is not available. But we can consider u as additional process noise and unify u with the process noise w. Then a more realistic system model is

$$x_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} w_k \tag{7}$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \ . \tag{8}$$

3 Robust H_∞ Filtering

As mentioned earlier, the Kalman filter requires the priori knowledge of statistical properties of the system and measurement noises, which are really hard to obtain practically. The ball filter with predictive model, described in [7], could bring a higher computational cost and memory occupancy, so this filter is not very efficient for the fast tracking problem. As a robust filter strategy, the minimax H_∞ strategy has the same efficient computation as that of the Kalman filter, and does not depend on the known noise statistics, but on the assumption of a finite disturbance energy. Consider the following linear system:

$$x_{k+1} = A_k x_k + B_k w_k \tag{9}$$

$$y_k = C_k x_k + v_k \ , \tag{10}$$

where $x_k \in \Re^n, w_k \in \Re^m, y_k \in \Re^p, v_k \in \Re^p$. A_k, B_k and C_k are matrices with appropriate dimension, (A_k, B_k) is controllable and (C_k, A_k) is detectable. Unlike the Kalman filter, which is interested in the estimation of the system state x_k, the H_∞ filter concerns the linear combination of x_k

$$z_k = L_k x_k \ . \tag{11}$$

The output matrix L_k is selected by the user according to the different applications. In our problem, we care about the ball's location and velocity, which just constitute the system state, so here L_k is specified as an identity matrix. The

H_∞ filter computes the estimated state $\hat{z}_k$ based on the measurement Y_k , where $Y_k = \{y_k, 0 \le k \le N-1\}$, and evaluates the estimation error by a performance measure, which can be regarded as an energy gain:

$$J = \frac{\sum_{k=0}^{N-1} \|z_k - \hat{z}_k\|_{Q_k}^2}{\|x_0 - \hat{x}_0\|_{p_0^{-1}}^2 + \sum_{k=0}^{N-1} \left(\|w_k\|_{W_k^{-1}}^2 + \|v_k\|_{V_k^{-1}}^2 \right)} \tag{12}$$

where N is the size of the measurement history, Q_k, p_0, W_k, V_k are the weighting matrices for the estimation error, the initial conditions, the process noise and the measurement noise. Moreover, $Q_k \ge 0, p_0^{-1} > 0, W_k > 0, V_k > 0$ and $((x_0 - \hat{x}_0), w_k, v_k) \ne 0$. The notation $\|x_k\|_{Q_k}^2$ is defined as $\|x_k\|_{Q_k}^2 = x_k^T Q_k x_k$. The denominator of J can be considered as the energy of the unknown disturbances, and the numerator is the energy of the estimation error. The H_∞ filter aims to provide an uniformly small estimation error $e_k = z_k - \hat{z}_k$ for any $w_k, v_k \in l_2$ and $x_0 \in R^n$, such that the energy gain J is bounded by a prescribed value:

$$\sup J < 1/\gamma \tag{13}$$

where sup denotes the supremum and $1/\gamma$ is the noise attenuation level. This condition keeps the robustness of the H_∞ filter, because the estimation energy gain is limited by $1/\gamma$ no matter what the bounded energy disturbances are.

To solve this optimal estimation $\hat{z}$ due to the bounded energy gain J, the H_∞ filter can be interpreted as a *minimax* problem ([10])

$$\min_{\hat{z}_k} \max_{(w_k, v_k, x_0)} J = -\frac{1}{2\gamma} \|x_0 - \hat{x}_0\|_{p_0^{-1}}^2 +$$

$$\frac{1}{2} \sum_{k=0}^{N-1} \left[\|z_k - \hat{z}_k\|_{Q_k}^2 - \frac{1}{\gamma} \left(\|w_k\|_{W_k^{-1}}^2 + \|v_k\|_{V_k^{-1}}^2 \right) \right] \tag{14}$$

where the estimation value $\hat{z}_k$ plays against the bounded energy disturbances w_k and v_k. Many strategies have been proposed for solving this *minimax* problem ([4]). We adopt a linear quadratic game approach ([10]), which does not require checking the positive definiteness and inertia of the Riccati difference equations for every step, but is implemented through recursive updating the filter gain H_k, the solution P_k of the Riccati difference equation, and the state estimation $\hat{x}_k$. The updating equations are given as follows:

$$\bar{Q}_k = L_k^T Q_k L_k \tag{15}$$

$$S_k = \left(I - \gamma \bar{Q}_k P_k + C_k^T V_k^{-1} C_k P_k \right)^{-1} \tag{16}$$

$$P_{k+1} = A_k P_k S_k A_k^T + B_k W_k B_k^T \tag{17}$$

$$H_k = A_k P_k S_k C_k^T V_k^{-1} \tag{18}$$

$$\hat{x}_{k+1} = A_k \hat{x}_k + H_k \left(y_k - C_k \hat{x}_k \right) \quad , \tag{19}$$

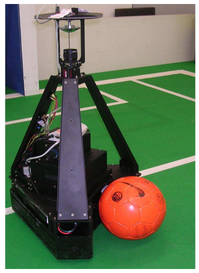

Fig. 1: The omnidirectional robot equipped with a digital color camera and a hyperbolic mirror on the top

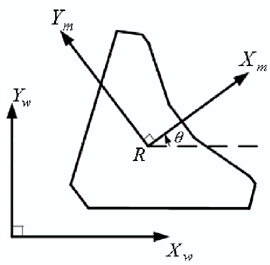

Fig. 2: Coordinate systems: $[X_w, Y_w]$ is the world coordinate system, $[X_m, Y_m]$ is the robot coordinate system

where $P_0 = p_0$ and $P_k > 0$. I is the identity matrix.

Apparently, these recursive equations have a similar form as the classic Kalman filter. Although we need not to know the statistics of noises w_k and v_k in the H_∞ filter, we should tune the weight matrices Q_k, p_0, W_k, V_k carefully, because these values determine the estimation error in the performance criterion (14). The weight matrices W_k, V_k can be chosen according to the experience about the noise. For example, if we know that the noises w is smaller than v, W_k should be smaller than V_k. p_0 is based on the initial estimation error. If we are highly confident about our initial estimation $\hat{Z}_0$, p_0 should be small. Similarly, if we care more about the precise estimations of some elements in the state, or some elements having bigger magnitude in their physical definition, the corresponding elements in the matrix Q_k can be set larger than others. As the performance criterion, γ can not be very large, because otherwise some eigenvalues of the matrix P may have magnitudes more than one. These eigenvalues prevent a proper derivation of the H_∞ filter equations, so that the H_∞ filter problem has no solution.

4 Experiments

The ball's observation values come from our omnidirectional view system and object detection process. Our omniderectional view system consists of a AVT Marlin F-046C color camera with a resolution of 780×580, which outputs signals up 50 times per second. In order to achieve a complete surrounding map of the robot, the camera is assembled pointing up towards a hyperbolic mirror which is mounted on the top of our omnidirectional robot, as shown in Fig.1. After obtaining the image from the camera, the other two processes, color calibration and distance calibration, map the colors to different classes based on the colors of objects and landmarks in the RoboCup domain, and the pixels in the image to the real world coordinates, respectively. At last, a fast object detection algorithm is used to get the ball's real world position, as described in [6].

While the camera image from the robot always displays the ball's relative position to the robot's position and orientation, the ball's relative position and velocity with respect to the robot coordinate system can be estimated directly by using the ball's observation values. When the ball's absolute position and velocity is required, the ball's observation values can be transformed into the world coordinate system, which is fixed in the robot playing field, by utilizing the robot's observation values. To prove the feasibility and the robustness of the H_∞ filter in estimating the ball's position and velocity with noisy observation values, we use the robot's odometer-based observation values in the experiments. The world coordinate system and the robot coordinate system are described in Fig.2.

All experiments were made in our robot laboratory having a half-field of the RoboCup-Middle size league. The H_∞ filter described in section 3 has been applied to tracking a rolling ball in the RoboCup domain, when the ball is pushed by a mobile robot to follow a linear path and a sinusoidal path with the constant desired translation velocity 0.3m/s and 0.5m/s respectively. The ball did not slide away from the robot during the whole pushing process because of the consecutive collisions with the robot. At every sampling time, the H_∞ filter estimated the ball's absolute position and velocity with respect to the world coordinate system, and the ball's relative position and velocity with respect to the robot coordinate frame. The noise attenuation level and weight matrices for estimating the x and y components were chosen as follows:

$$\gamma^x = 2.0,\ P_0^x = \begin{bmatrix} 30 & 0.004 \\ 30 & 2 \end{bmatrix},\ Q_k^x = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix},\ W_k^x = 1,\ V_k^x = 10\ ;$$

$$\gamma^y = 1.5,\ P_0^y = \begin{bmatrix} 10 & 0.05 \\ 30 & 0.8 \end{bmatrix},\ Q_k^y = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix},\ W_k^y = 10,\ V_k^y = 1\ .$$

To evaluate the performance of the H_∞ filter, a Kalman filter with assumed noise variance was also used to estimated the ball's position and velocity with the same observation values. The initial estimate error covariance matrices P_o and the probability distributions of process noise and measurement noise are chosen as follows:

$$p_0^x = \begin{bmatrix} 0.01 & 0.0001 \\ 0.003 & 0.005 \end{bmatrix},\ p(w^x) \sim N(0, 0.01),\ p(v^x) \sim N(0, 0.0001)\ ;$$

$$p_0^y = \begin{bmatrix} 0.01 & 0.0001 \\ 0.01 & 0.005 \end{bmatrix},\ p(w^y) \sim N(0, 1),\ p(v^y) \sim N(0, 0.0001)\ .$$

From the results shown in figures 3-8, we can see the H_∞ filter eliminated the high frequency components of the measurement and estimated the ball's position values sufficiently. Figures 5-8 show that the estimated positions from the H_∞ filter are slightly better than those from the Kalman filter. Figures 9-10 illustrate that the ball's velocity is effectively estimated and the H_∞ filter is better than the Kalman filter, while the estimated x-velocities from the H_∞ filter approach to the ideal robot's x-velocity 0.3m/s with less time and are more smooth than those from the Kalman filter.

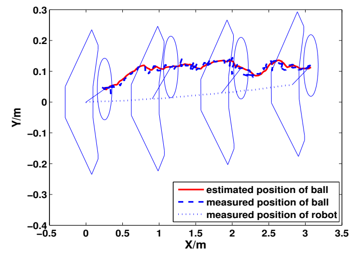

Fig. 3: Absolute positions of robot and ball along the linear path

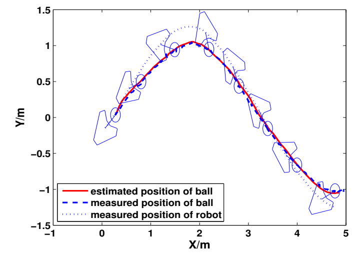

Fig. 4: Absolute positions of robot and ball along the sinusoidal path

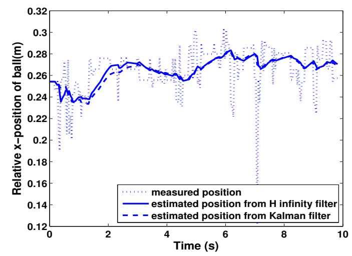

Fig. 5: Relative x-positions of ball along the linear path

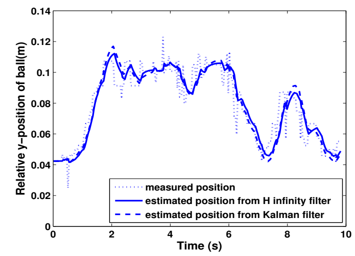

Fig. 6: Relative y-positions of ball along the linear path

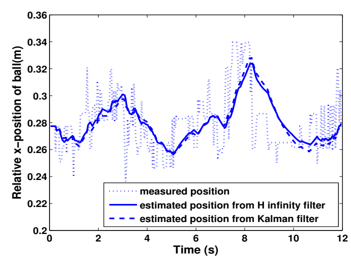

Fig. 7: Relative x-positions of ball along the sinusoidal path

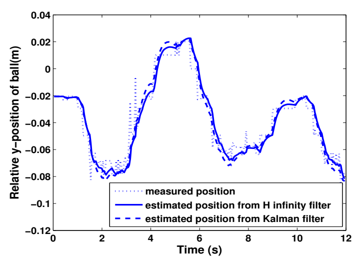

Fig. 8: Relative y-positions of ball along the sinusoidal path

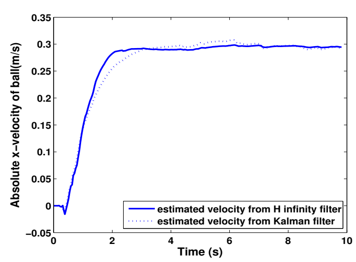

Fig. 9: Absolute x-velocities of ball along the linear path

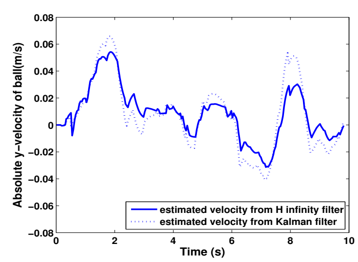

Fig. 10: Absolute y-velocities of ball along the linear path

5 Conclusion

In this paper we introduce a robust H_∞ filter, which does not require a priori knowledge about the statistical properties of the system and measurement noise, but only depends on the assumption of finite noise power. The recursive equations of the H_∞ filter are very similar to those of the Kalman filter, so the H_∞ has relatively low computation cost in the implementation and adapts to the real time estimation problem. With the real-world experiments, where the ball was following the given paths pushed consecutively by an omnidirectional robot, the performance of the H_∞ filter was evaluated by comparing the estimation values with those from the Kalman filter. The results of the estimated ball's position and velocity show that the H_∞ filter eliminates the high frequency noise components of the measurements and estimates the ball's position and velocity robustly in the pushing process. Moreover, the H_∞ filter in this application is shown to be superior to a Kalman filter, which requires manual tuning of the noise parameters.

References

1. Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley, 2001.
2. L. Freeston. Applications of the kalman filter algorithm to robot localisation and world modelling. Technical report, University of Newcastle, NSW, Australia, 2002.
3. A. Gelb. *Applied Optimal Estimation*. Cambridge, 1974.
4. B. Hassibi, A. Sayed, and T. Kailath. *Indefinite Quadratic Estimation and Control - A Unified Approach to H2 and H Infinity Theories*. Society for Industial and Applied Mathematics, Philadelphia, 1999.
5. P. Heinemann, M. Plagge, A. Treptow, and A. Zell. Tracking dynamic objects in a robocup environment in a robocup environment - the attempto tübingen robot soccer team. In A. D.Polani, B.Browning and K.Yoshida, editors, *RoboCup - 2003: Robot Soccer World Cup VII*, Lecture Notes in Computer Science. Springer-Verlag, 2004.
6. P. Heinemann, T. Rückstieß, and A. Zell. Fast and accurate environment modelling using omnidirectional vision. In H. U.Ilg, H.Bülthoff, editor, *Dynamic Perception*. Infix, 2004.
7. M. Lauer, S. Lange, and M. Riedmiller. Modeling moving objects in a dynamically changing robot application. In U. Furbach, editor, *Advances in Artificial Intelligence*, pages 219–303. Springer, 2005.
8. S. Petrov. Computer vision, sensorfusion und verhaltenssteuerung für fussballroboter. Master's thesis, Freie Universität Berlin, Institute für Informatik, 2004.
9. J. Ruizdel-Solar and P. Vallejos. Motion detection and tracking for an aibo robot using camera motion compensation and kalman filtering. In *Lecture Notes in Computer Science 3276 (RoboCup Symposium 2004)*. Springer, 2004.
10. X. Shen and L. Deng. Game theory approach to discrete H_∞ filter design. *IEEE Transcation on Signal Processing*, 45(4), 1997.